Proceedings of

COGAIN 2007

‘Gaze-based Creativity, Interacting with Games and On-line Communities’

COGAIN NoE is funded by the EU IST 6th framework program.

COGAIN 2007 Papers Index

Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures

Jacob O. Wobbrock,[1] James Rubinstein,[2] Michael Sawyer,[3] and Andrew T. Duchowski[4]

[1]The Information School
University of Washington
Seattle, WA 98195 USA
wobbrock@u.washington.edu

[2]Department of Psychology
[3]Department of Industrial Engineering
[4]School of Computing
Clemson University
Clemson, SC USA 29634
{jrubins, msawyer, andrewd}@clemson.edu

Keywords: Text input, eye-typing, unistrokes, EdgeWrite, EyeWrite.

Introduction

People with severe motor disabilities often cannot use a conventional keyboard and mouse. One option for these users is to enter text with their eyes using an eye-tracker and on-screen keyboard (Istance et al. 1996). Such keyboards usually require users to stare at keys long enough to trigger them in a process called 'eye-typing' (Majaranta and Räihä 2002). However, eye-typing with on-screen keyboards has many drawbacks, including the reduction of available screen real-estate, the accidental triggering of keys, the need for high eye-tracker accuracy due to small key sizes, and tedium. In contrast, we describe a new system for 'eye-writing' that uses gestures similar to hand-printed letters. Our system, called *EyeWrite*, uses the EdgeWrite unistroke alphabet previously developed for enabling text entry on PDAs, joysticks, trackballs, and other devices (Wobbrock et al. 2003, Wobbrock and Myers 2006). EdgeWrite's adaptation to EyeWrite has many potential advantages, such as reducing the need for eye-tracker accuracy, reducing the screen footprint devoted to text input, and reducing tedium. However, the best interaction design was non-obvious. As a result, EyeWrite required extensive iteration and usability testing. In this paper we describe EyeWrite and its development, and offer initial evidence in favour of this new technique.

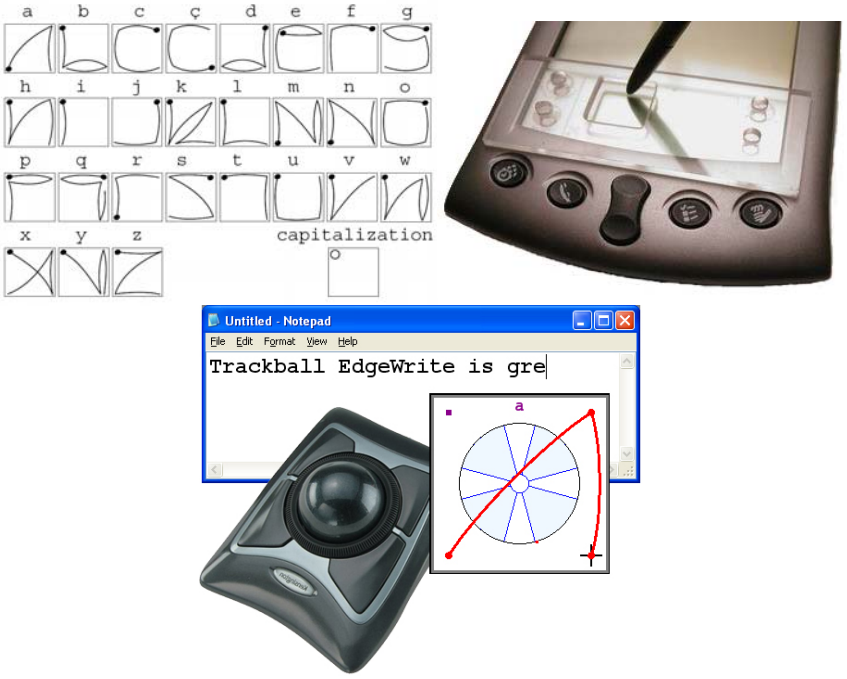

Figure 1. Stylized EdgeWrite letters, Stylus EdgeWrite on a PDA, and Trackball EdgeWrite. Used with permission.

Background and Related Work

EyeWrite is based on the gestural unistroke alphabet used in EdgeWrite (Wobbrock et al. 2003), a text entry method capable of being used on numerous devices (Figure 1). Our decision to use EdgeWrite's alphabet for EyeWrite was not arbitrary; in fact, the alphabet has important properties that make it useful for gaze input. Since the eyes move in saccadic bursts rather than smooth paths, it would be impossible to 'write fluidly' as one does with a pen. Fortunately, EdgeWrite recognizes characters based only on the order in which the four corners of its square input area are hit. This allows EyeWrite to employ four discrete corner targets that help users form their gestures. Another benefit of EdgeWrite's corner-based recognition scheme is that it provides tolerance to tremor in the stroke path, since all that matters is the order in which the corners are hit. This means that eye-tracker jitter is not overly detrimental to EyeWrite's recognition. A third benefit of using EdgeWrite's alphabet is that it has been shown to be very easy to learn, which is important when considering tradeoffs with on-screen keyboards, which are easily comprehended.

Most prior eye-based text entry methods use on-screen keyboards developed for eye-typing (Istance et al. 1996, Lankford 2000, Majaranta and Räihä 2002). To our knowledge, EyeWrite is the first letter-like gestural text entry system for the eyes. Only a few prior systems use eye-based gestures, but these gestures are defined by underlying screen regions, making these systems fancier variants of eye-typing. One system is *Dasher*, which uses expanding letter regions that move toward the user's gaze point (Ward and MacKay 2002). Although Dasher is fast, it can be visually overwhelming for novice users since letter regions 'swarm' toward the user. Other gestural interfaces are the systems developed by Isokoski as part of his exploration of off-screen targets (Isokoski 2000). These designs place letter regions beyond the edges of the screen to solve the Midas Touch problem. A noted issue, however, is that users have difficulty locating the off-screen targets.

The EyeWrite Design

Adapting EdgeWrite for use with the eyes may initially seem straightforward, but the design challenge was considerable. Two key questions were how to translate eye movements into EyeWrite gestures, and how to segment between letters when a letter was finished. This latter issue is the so-called 'segmentation problem.'

Our first design simply mimicked Stylus EdgeWrite (Wobbrock et al. 2003). A literal trace was drawn as the user moved his eyes within EyeWrite's on-screen input area. However, drawing a trace based on the literal eye position created strokes that were jagged and distracting, even with filtering. Since there was no 'stylus lift' signal as in Stylus EdgeWrite, segmentation was first achieved by looking for a cessation of movement. This worked poorly because eye-tracker jitter meant the eye-trace never stopped moving. A second scheme segmented letters after sufficient time had elapsed since the last corner was entered. This time was calculated using the average inter-corner time for the current stroke. However, this resulted in unwanted segmentations when users paused to think. It also meant that the gaze point would remain in the same corner after segmentation, which meant that this corner had to be re-entered before a new letter could be started there.

Our second design utilized a vector-based approach akin to Trackball EdgeWrite (Wobbrock and Myers 2006). In this approach, the absolute position of the eyes was no longer relevant. Instead, the *direction* in which the eyes moved indicated the corner to which the stroke should proceed. When such a vector was indicated, a stylized stroke was drawn from the previous corner to the indicated corner. Although this solved the jitter and distraction of the first design, it momentarily decoupled the stroke corner from the user's gaze point. This resulted in the creation of unwanted vectors and, in turn, unwanted corners.

Another problem was that the user could not look away from EyeWrite to verify their text without indicating a movement vector.

Our third design accommodated lessons from the first two. We returned to a tight coupling between the user's gaze and EyeWrite's input, but instead of drawing a literal eye-trace as in the first design, we drew stylized arcs between corners as in the second design. Instead of vectors, corners were simply hit-tested for the presence of the eyes—when the gaze point entered a new corner, an arc was drawn there. Thus, the gaze point and stroke corner were never decoupled. We also gave users direct control over the segmentation process by segmenting only when the eyes returned to the center of the input area. Users could therefore prevent segmentation and 'pause to think' by simply leaving their gaze in the current corner. Return-to-center segmentation also meant that every new letter would be started from the center. As in the first design, segmentation time was based on the average inter-corner time, but now with a minimum threshold about twice the time of a saccade. This prevented unwanted segmentations when moving among corners. Users could also clear their current stroke by simply glancing away from the EyeWrite square. Finally, to reduce the need to look away between letters to verify the last entry, an incremental recognition result was displayed in the current corner of the EyeWrite square. It was also displayed in the center of the square after segmentation, so users knew exactly what character had been produced. These improvements culminated in the current version of EyeWrite (Figure 2). EyeWrite is implemented in Visual C# using .NET 2.0. We run it on a Tobii 1750 eye-tracking system.

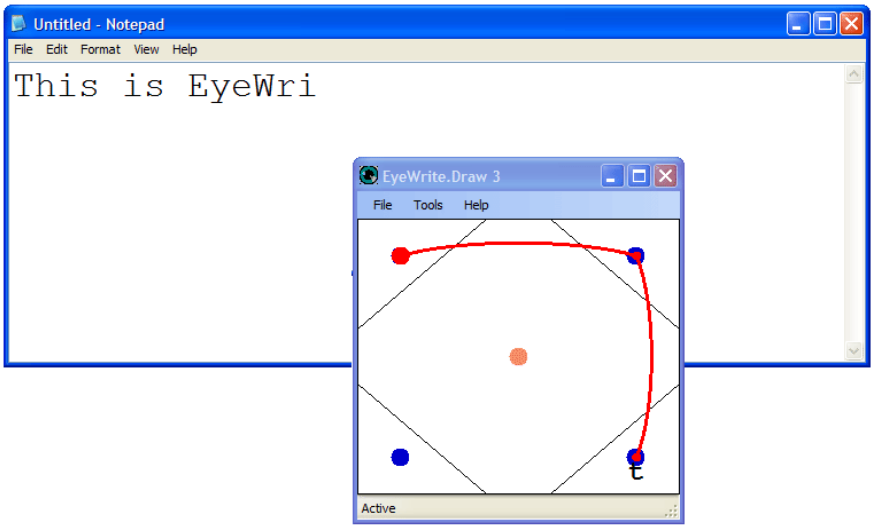

Figure 2. EyeWrite being used with Microsoft Notepad. Up to this point, a 't' has been made, which appears in the bottom-right corner. When the user is ready to segment, he will look at the salmon-coloured dot in the centre.

Conclusion and Future Work

To our knowledge, this paper is the first to describe a letter-like gestural writing system for the eyes. EyeWrite has potential advantages including reduced screen footprint, few large proximate targets, tolerance to eye-tracker jitter, ability to add commands without increased screen consumption, reduced distance between input and output areas, and greater elegance through minimalist design. Entering gestures may also be less tedious and more fun than repeatedly dwelling over keys (Wobbrock and Myers 2006).

Going forward, EyeWrite will be compared to eye-typing in a longitudinal study. Such a study will measure entry, error, and learning rates. Conceptually, the speed comparison amounts to whether fixating on 2-4 large, proximate targets (EyeWrite) is faster than dwelling on one smaller, more distant target (eye-typing). Our initial study results to-date indicate that experienced EyeWrite users can write at about 7.99 WPM with 1.25% uncorrected errors. A full study will allow us to elaborate on these preliminary findings.

References

Isokoski, P. (2000) Text input methods for eye trackers using off-screen targets. In *Proc. ETRA '00*. New York: ACM Press, pp. 15-21.

Istance, H.O., Spinner, C. and Howarth, P.A. (1996) Providing motor impaired users with access to standard Graphical User Interface (GUI) software via eye-based interaction. In *Proc. ECDVRAT '96*. Reading, UK: University of Reading, pp. 109-116.

Lankford, C. (2000) Effective eye-gaze input into Windows. In *Proc. ETRA '00*. New York: ACM Press, pp. 23-27.

Majaranta, P. and Räihä, K.-J. (2002) Twenty years of eye typing: Systems and design issues. In *Proc. ETRA '02.* New York: ACM Press, pp. 15-22.

Ward, D.J. and MacKay, D.J.C. (2002) Fast hands-free writing by gaze direction. *Nature 418*, p. 838.

Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003) EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proc. UIST '03*. New York: ACM Press, pp. 61-70.

Wobbrock, J.O. and Myers, B.A. (2006) Trackball text entry for people with motor impairments. In *Proc. CHI '06*. New York: ACM Press, pp. 479-488.