An Eye on Input:
Research Challenges in Using the Eye for Computer Input Control

I. Scott MacKenzie
Dept. of Computer Science and Engineering
York University
Toronto, Ontario, Canada M3J 1P3
mack@cse.yorku.ca

Abstract

The human eye, with the assistance of an eye tracking apparatus, may serve as an input controller to a computer system. Much like point-select operations with a mouse, the eye can "look-select", and thereby activate items such as buttons, icons, links, or text. Applications for accessible computing are particularly enticing, since the manual ability of disabled users is often lacking or limited. Whether for the able-bodied or the disabled, computer control systems using the eye as an input "device" present numerous research challenges. These involve accommodating the innate characteristics of the eye, such as movement by saccades, jitter and drift in eye position, the absence of a simple and intuitive selection method, and the inability to determine a precise point of fixation through eye position alone.

Looking

For computer input control, current eye tracking systems perform poorly in comparison to their manually operated counterparts, such as mice or joysticks. To improve the design of eye tracking systems, we might begin by asking: What is a perfect eye tracker?

This question is only worth considering from an engineering perspective, since it is not possible to know what meaning the user ascribes to a visual image. Yet, even from an engineering perspective, the challenge is formidable. First, consider the capabilities of a human interacting with a computer, apart from eye tracking. It is relatively straightforward for a computer user to look at, or fixate on, a single pixel on a computer display, provided the pixel is distinguishable from its neighbors by virtue of colour or contrast. Furthermore, if in control of an on-screen pointer via a mouse, the user can, with a little effort, position the pointer's hot spot (e.g., the tip of an arrow) on the desired pixel. So a "perfect" eye tracker would, like a mouse, return the *x-y* coordinate of the gazed-upon pixel. Is such single-pixel "looking" possible with an eye tracker, a perfect eye tracker? The answer is no.

Importantly, the software and algorithms in an eye tracking system have little to work with other than a digitized camera image of the user's face and eyes. An example is shown in Figure 1. The user's left eye is expanded showing two glints – corneal reflections from the eye tracker's infrared light sources – and a crosshair. Through some sophisticated software, the center of the crosshair maps to the user's point of fixation on the computer display. Or does it? Eye tracking systems are generally configured to hide the corresponding crosshair on the computer display even though the software indeed computes an *x-y* coordinate of the point of fixation. The reason for this is simple. The on-screen crosshair is rarely located at the point of fixation, or more correctly, at the point of fixation in the user's mind. Upon seeing an on-screen crosshair that is "near" the actual point of fixation, there is a strong tendency for the user to look at the crosshair. The result is obvious enough. The ensuing cat-and-mouse exercise is quickly a source of frustration. So, the on-screen crosshair is hidden.

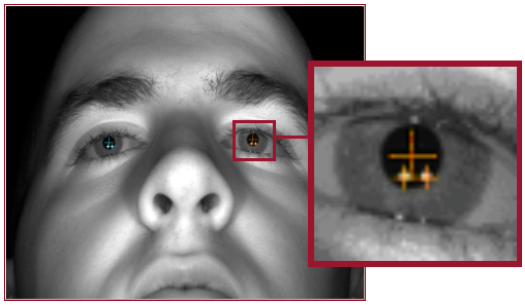

Figure 1

What is the problem in the scenario just described? Is this simply a matter of developing a better algorithm? The points of deviation are well known. The digitized camera image is of limited resolution. The user's head moves about slightly. Each user's eyes are different in size and shape. There are slight on-going movements of the eye, known as jitter and drift. These issues can be accommodated through improved engineering, better algorithms, and careful calibration. So, we might expect these challenges to be eventually overcome, such that the position the on-screen crosshair is precisely at the user's point of fixation. But, this is not the case. To illustrate what may be the most significant obstacle in perfecting eye tracking technology, consider the sketch in Figure 2.

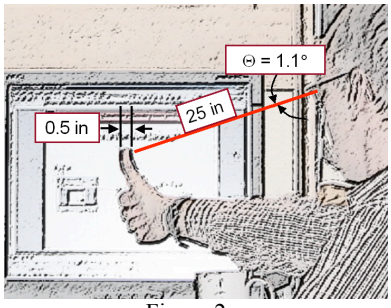

Figure 2

The user's arm is out-stretched to a distance of 25 inches. His thumb is half an inch wide. The angle of visual arc is about 1.1

Copyright © 2010 by the Association for Computing Machinery, Inc.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.
ETRA 2010, Austin, TX, March 22 – 24, 2010.
© 2010 ACM 978-1-60558-994-7/10/0003 $10.00

degrees. This exercise is commonly restated in the literature to demonstrate the approximate diameter of the eye's foveal image. The fovea is located near the center of retina and is responsible for sharp central vision. Although the fovea is only about 1% of the retina in size, the neural processing associated with the foveal image engages about 50% of the visual cortex in the brain. From an eye tracking perspective, the foveal image represents a "region of uncertainty". Figure 3 shows the approximate size of the foveal image superimposed on a typical full-screen rendering of a web page. Several selectable buttons fall within the foveal image, making selection difficult.

Figure 3

Even if it were possible to determine, through careful engineering and judicious calculation, the exact region on the computer display that is projected onto the fovea, it is not possible to determine a particular point, or pixel, within the foveal image to which the user is attending. This limitation cannot be overcome by improvements in eye tracking technology.

Selecting

While many applications of eye tracking systems are directed at visual search (where is the user looking?), other applications take on the additional and role of selecting – using the eye as a computer input control. Holding the point of fixation still for a period of time ("dwelling") is the most common method of selecting, but other methods are also possible and practical, depending on the context and application. These include using a separate manually operated switch or key, or using blinks, winks, voluntary muscle contractions, and so on.

Regardless of the method of selection, the system must accommodate the foveal image's region of uncertainty. The most obvious solution is simply to design interfaces where the largest selectable region is larger than the foveal image, as depicted in Figure 4. Clearly, such a big-button approach seriously limits the interface options. In view of this, numerous "smart" target acquisition techniques have been proposed. By and large, they involve determining which target is the intended target, given inaccuracies and fluctuations in what the eye tracking system returns as the user's point of fixation. The techniques allow for selection of targets smaller than the foveal image. Techniques proposed in the literature are numerous and accompanied with a myriad of terms, such as cursor warping, grab and hold, target expansion, fisheye lenses, snap on, snap clutch, nearest neighbor, and so on.

Figure 4

Accessible Computing

Eye tracking is of particular interest in accessible computing, where the user has a disability that limits or excludes manual input. If holding the head steady is difficult, eye tracking will be difficult as well. Yet in such cases, limited eye tracking may be possible, such as detecting simple eye gestures (e.g., look left, look right, etc.). Single-switch scanning keyboards have a long history in accessible computing. A scanning ambiguous keyboard (SAK) is a recent variation that promises higher text entry rates than traditional row-column scanning arrangements. Input only requires a single key or switch. The key press signal can be provided using the blink detection capability of an eye tracker. See the Ashtiani and MacKenzie paper elsewhere in the *ETRA 2010* proceedings.

Evaluating

Evaluating the eye working in concert with an eye tracking system requires a methodology that uniquely addresses the characteristics of both the eye and the eye tracking apparatus. As well as reviewing research challenges in looking and selecting, this presentation will summarize and advance the use of standardized empirical research methods for evaluating and comparing the many new approaches that researchers develop to improve the ability of eye tracking systems to allow the user through, his or her eyes, to control a computer system.