Editor: Ronald D. Williams, University of Virginia, Thornton Hall/Electrical Engineering, Charlottesville, VA 22903;
phone, (804) 924-7960; fax, (804) 924-8818; Internet, rdw@virginia.edu

Eye-gaze computer interfaces

Computers that sense eye position on the display

Thomas E. Hutchinson, University of Virginia

The human eye's point of regard and pupil size have been of interest for centuries, with the earliest record attributed to a second-century-BC Chinese jade dealer who employed "ocular analysis" to determine customer interest in his product. Devices for more accurate analysis have been around for decades, especially since the advent of microcomputers. By the late 1980s, personal computers were being used in ocular analysis by James Levine at IBM, at the University of Virgina, and in other laboratories throughout the world.

Method. The most widely used method of determining ocular point of regard is based on the "bright eye" phenomenon. The method uses infrared illumination of the face, originating from a light-emitting diode placed at the center of a video camera lens, to produce a high signal-to-noise image of the eye's pupil. The resulting video signal is digitized and sent to the personal computer for analysis. The analysis determines the pupil center, along with the center of a glint reflection arising from the front surface reflection of the infrared source. The vector distance between the glint and pupil center is a measure of point of regard and, when calibrated, determines eye position on a computer display with high accuracy. A cursor, placed at the point where the computer thinks the eye is looking, provides interactive control to null slight calculation errors.

Clearly, this description grossly simplifies the actual process. The analysis requires several thousand lines of code to implement algorithms that (1) accurately calculate pupil and glint center, (2) calibrate the system to the geometry of a particular user's eye, and (3) null the effect of both lateral and display-distancing head motions. Linking point-of-regard data to existing programs, of course, requires even more code.

Applications. The usefulness of a system that determines point of focus on a computer display is limited only by imagination. This is particularly true if the system also allows control of the computer with existing software. The idea of an eye-operated mouse leaps to mind; in fact, one that totally replaces traditional mouse functions for applications from spreadsheets to games has been implemented in the Windows environment at the University of Virginia.

The university's system, called ERICA, for Eye-gaze Response Interface Computer Aid, was conceived as a communication and environmental control device to aid the most severely handicapped, those without the use of hands or voice. It has been used by victims of spinal accidents, cerebral palsy, and ALS (Lou Gehrig's disease). Most recently, Steven Hawking's system in Cambridge was custom coded with a word processor containing his most frequently used words cued to his single-letter entries. The system works entirely by eye-gaze input with the "keyboard" displayed on a portion of the screen. Other systems of this type are available commercially from L.C. Technologies of Fairfax, Virginia.

The implications of an interactive eye-gaze computer interface are far reaching. ERICA's Eyemouse implementation alone would greatly enhance anyone's word-processing capabilities. When keyboard strokes are available and easy, combined eye/key activation is a powerful interactive

(Continued on p. 67)

What you look at is what you get

Robert J.K. Jacob, Naval Research Laboratory

As computers become more powerful, the critical bottleneck in their use is often in the user interface, not in the computer processing. Research in human-computer interaction seeks to increase the communication bandwidth between the user and the machine.

At the Naval Research Laboratory, we are studying hitherto-unused methods by which users and computers can communicate information, focusing on obtaining input from the user's eye movements. That is, the computer identifies the point on its display screen at which the user is looking and uses that information as part of its dialogue with the user. For example, instead of requiring the user to indicate an icon by pointing at it with a mouse or by entering its name on a keyboard, the computer can determine which icon the user is looking at and give the desired information immediately.

Speed potential. Eye trackers have existed for years, but their use has largely been confined to laboratory experiments. The equipment is now becoming sufficiently robust and inexpensive to consider use in a real user-computer interface. But, to make this happen, we need appropriate interaction techniques that incorporate eye movements into the user-computer dialogue conveniently and naturally.

Because people can move their eyes

extremely rapidly and with little conscious effort, a user interface based on eye-movement inputs has the potential for faster, easier interaction than current interfaces. A simple thought experiment suggests the speed advantage: Before you operate any mechanical pointing device, you usually look at the destination. Thus, the eye movement is available as an indication of your goal before you can actuate any other input device. However, people are not accustomed to operating devices simply by moving their eyes.

Midas Touch and other problems. Our experience is that, at first, it is empowering to simply look at what you want and have it happen. Before long, though, it becomes like the Midas Touch. Everywhere you look, another command is activated; you cannot look anywhere without issuing a command.

The challenge in building a useful eye-movement interface is to avoid the Midas Touch problem. New interaction techniques must be carefully designed to ensure not only the speed but also the natural, unobtrusive use of eye input. Our approach is to think of eye position as one piece of information available from a variety of input devices, not as the intentional actuator of the principal input device.

Unfortunately, people's eye movements are not nearly as slow or deliberate as their operation of conventional computer input devices. Eyes dart from point to point in rapid and sudden saccades. Even when a user thinks he or she is viewing a single object, the eyes do not remain still for long. Thus, it would be inappropriate to plug in an eye tracker as a direct replacement for a mouse. Wherever possible, we try to obtain information from the natural movements of the user's eyes while viewing the display, rather than requiring the user to make specific, trained eye movements to actuate the system.

Processing stages. We partition the problem of using eye-movement data into two stages. First we process the raw data from the eye tracker to filter noise, recognize fixations, compensate for local calibration errors, and try to reconstruct the user's conscious intentions from the available information.

This processing stage uses a model of eye motions (fixations separated by saccades) to drive a fixation recognition algorithm that converts the continuous, somewhat noisy stream of raw eye-position reports into discrete tokens that represent user's intentional fixations. The tokens are passed to our user interface management system, along with tokens generated by other input devices being used simultaneously, such as the keyboard or mouse.

Interaction techniques. Next, we design generic interaction techniques based on these tokens as inputs. The first technique we developed is for object selection. The task is to select one object from among several displayed on the screen, for example, one of several file icons on a desktop. With a mouse, this is usually done by pointing at the object and then pressing a button. With the eye tracker, there is no natural counterpart to the button press. We reject using a blink for a signal because it detracts from the naturalness possible with an eye-movement-based dialogue by requiring the user to think about when he or she blinks.

We tested two alternatives. In one, the user looks at the desired object and then presses a keypad button to indicate his or her choice. The second alternative uses dwell time. If the user continues to look at the object for a sufficiently long time, it is selected without further operations.

At first this seemed like a good combination. In practice, however, the dwell-time approach proved much more convenient. While a long dwell time might prevent an inadvertent selection caused by the user's simply looking around the display, it mitigates the speed advantage of eye-movement input and reduces interface responsiveness. On the other hand, if an erroneous selection can be undone trivially, a very short dwell time can be used. For example, if a second selection instantaneously overrides the first, selection of a wrong object followed by selection of the right object causes no adverse effect.

This approach, using a 150- to 250-millisecond dwell time, gives excellent results. The lag between eye movement and system response (required to reach the dwell time) is hardly detectable to the user, yet long enough to accumulate sufficient data for fixation recognition and processing. The subjective feeling is one of a highly responsive system, almost as though the system is executing the user's intentions before he or she expresses them.

When object selection is more difficult to undo, we use button confirmation rather than a longer dwell time.

Other interaction techniques we have developed and are studying include continuous display of attributes of eye-selected objects, selection and dragging of objects through a combination of eye movement and button press and release, pull-down menu commands using dwell time to select and lookaway to cancel, and forward and backward eye-controlled text scrolling.

The next step in this research is controlled observation of the new techniques to put our results on a more objective footing. Our first experiment will compare object selection by dwell time with conventional selection by mouse pick. Pilot runs suggest a 30 percent decrease in time for the eye over the mouse, although the eye trials show more variability.

Noncommand style. Eye-movement-based interaction techniques exemplify a new, noncommand style of interaction. In a noncommand-based dialogue, the user does not issue specific commands; instead, the computer passively observes the user and provides appropriate responses. Because the inputs in this style of interface are often unintentional, they must be interpreted carefully to avoid annoying the user with unwanted responses to inadvertent actions. Our research with eye movement provides an example of how these problems are being attacked.

Further reading

Eye-movement interfaces

T.E. Hutchinson et al., "Human-Computer Interaction Using Eye-Gaze Input," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 19, No. 6, 1989, pp. 1,527-1,534.

R.J.K. Jacob, "The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At Is What You Get," *ACM Trans. Information Systems*, Vol. 9, No. 3, Apr. 1991, pp. 152-169.

I. Starker and R.A. Bolt, "A Gaze-Responsive Self-Disclosing Display," *Proc. ACM CHI 90 Human Factors in Computing Systems Conf.*, Addison-Wesley/ACM Press, 1990, pp. 3-9.

General

J. Nielsen, "Noncommand User Interfaces," *Comm. ACM*, Vol. 36, No. 4, Apr. 1993, pp. 83-99.

B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd ed., Addison-Wesley, Reading, Mass., 1992.

Proc. ACM CHI 83-93 Human Factors in Computing Systems Conferences, Addison-Wesley/ACM Press, 1983-1993.

Robert J.K. Jacob is a computer scientist in the Human-Computer Interaction Lab at the Naval Research Laboratory, Code 5530, Washington, DC 20375-5337; his e-mail address is jacob@itd.nrl.navy.mil.