9

Eye Tracking in Virtual Environments

Joseph Wilder,[1] George K. Hung,[2]
Marilyn Mantei Tremaine,[1] and Manpreet Kaur[1]
Rutgers University
[1] Center for Advanced Information Processing (CAIP)
[2] Department of Biomedical Engineering
96 Frelinghuysen Rd.
Piscataway, NJ 08854-8088
wilder@caip.rutgers.edu, shoane@rci.rutgers.edu,
tremaine@acm.org, manpreet.kaur@t1.com

1. INTRODUCTION

Development of instruments for tracking eye movements has been under way for over 25 years (Cornsweet & Crane, 1973; Young & Sheena, 1975). Early eye trackers were cumbersome devices in which the user wore uncomfortable gear, such as bulky helmets or other equipment affixed to the head. Except under the most controlled conditions, these devices were not capable of maintaining their calibration because it was difficult to maintain strict positioning of the devices on the head. More recently, eye trackers have been developed that attempt to overcome these shortcomings. Ultralight headband-mounted trackers have been developed that reduce slippage. Also, newer devices, based on a tracking system mounted on the display screen rather than the head, permit the user to roam relatively freely over a comfortable region of space.

The primary goal of early eye trackers was to support research in human visual data acquisition. But as instrumentation technology continued to evolve, it eventually led to applications in a variety of settings where understanding of human perception, attention, search, tracking, and decision making are of great importance. This includes industrial inspection (Megaw & Richardson, 1979), medical image analysis (Reuter & Shenck, 1985), visual response to advertising (Lohse & Johnson, 1996; Russo & LeClerc, 1994), and analysis of the performance of airplane pilots (Sanders, Simmons, & Hoffman, 1979). More recently, eye-tracking measures have been used to understand the causes of cybersickness in flight simulators and virtual environment (VE) devices (Kaiser, 1999) and to pinpoint theories of language processing (Eberhard, Spivey-Knowlton, Sedivy, & Tanenhaus, 1995). At the same time, there have been efforts to develop eye trackers as visual communication and control devices (Freedman, 1984; Jacob, 1991; Velichkovsky & Hansen, 1996), e.g., to aid disabled people (Yamada & Fukada, 1987). In particular, eye movement–based human–computer interaction (HCI) techniques have been studied to assess the ability of gaze, along with speech and tactile interactions, to provide more natural interactions with computers than the traditional mouse and keyboard.

This chapter discusses the use of gaze-based communication and control systems to support multimodal interactions, in real and virtual environments. Specifically, it first describes how these new trackers work (section 2), then, how they are used as input devices (section 3), how they are being integrated into VE systems (section 4), the human factors analyses that are underway to assess the effectiveness of eye tracking in comparison and in combination with other modalities (section 5), and concludes with an assessment of future directions in eye tracking for virtual environments (section 6).

2. HOW EYE TRACKERS FOR VE WORK

Eye trackers have been developed for measuring many properties of visual behavior, for example, saccade and smooth pursuit, as well as accommodation, vergence, and pupillary response. The key measurements for visual communication and control that have been used are saccades and pursuit, and the fixations between these movements.

Normal eye movements (e.g., in reading) consist of saccades, or jumps, from one fixation (stationary position) to another. Typically saccades range in amplitude from 1 to 20 degrees, corresponding to a duration of 30 to 70 msec, and peak velocities of 70 to 600 deg/sec, respectively (Bahill & Stark, 1979). When following slowly moving targets in the range of 1 to 30 deg/sec, the eye can track these movements with a smooth-pursuit behavior that appears to partially stabilize the image of the target on the retina (Young & Sheena, 1975).

Several basic techniques have been used for tracking eye position. They can be divided into contact and noncontact methods. The contact method uses magnetic induction of two sets of orthogonal induction coils, driven in quadrature, on a scleral ring worn on the eye of the participant. Rotation of the eye results in changes in the amounts of phase-locked horizontal and vertical induced currents picked up by the scleral ring, thereby providing a signal proportional to eye position. The search coil technique provides accurate measurement of two-dimensional (2-D) eye position, but requires local anesthesia that limits the experimental session to about 20 minutes (Remmel, 1984). Among the noncontract methods, the limbus eye tracker is the simplest and least expensive. Two infrared (IR) photo-emitters pulsed at 1 kHz are aimed at the iris (dark)–scleral (white) boundary on either side of the eye. More, or less, light will be reflected depending on the position of the eye relative to the emitter. A pair of infrared detectors picks up the reflected light from the emitters. The differential signal from the emitters is demodulated and filtered to provide a signal proportional to horizontal eye position. This technique provides a relatively easy-to-use recording method, but it is limited to horizontal eye movements. Another technique is the Purkinje eye tracker, which measures the relative displacement of the images formed by the reflection of a light source at the anterior corneal surface and the posterior lens surface, which are known as the first and fourth Purkinje images, respectively. Rotation of the eye results in a greater displacement of the first relative to the fourth Purkinje image, thereby providing a signal proportional to eye position. However, this device requires precise alignment and is not suitable for experiments that permit relatively free movement by the participant (Young & Sheena, 1975).

The video-based eye tracker is the most suitable for 2-D recording of eye movements of a participant who is relatively free to move about in a region of space. It is currently used for communication and control in virtual environments. The tracker captures a video image of the eye illuminated by a distant, low-power, infrared light source and creates an image that is seen as a highlight spot on the surface of the cornea. The image is analyzed by a computer, which calculates the centroid of the corneal reflection as well as the centroid of the pupil. The corneal reflection from the front spherical surface of the eye is insensitive to eye rotations, but it is sensitive to translational movements of the eye or head. On the other hand, the pupil center is sensitive to both rotation and translation. Thus, the difference between the pupil center and the corneal reflex provides a signal proportional to eye rotation, and thereby the direction of gaze, which is relatively free of translational artifacts.

In head-mounted eye trackers, the IR source, the camera that views the eye, and a second camera that views the observed scene are all mounted on the head. Additionally, as the user roams about in a prescribed space, a magnetic sensor is affixed to the head to record the position of the user in the space. The system computes the point of gaze in the image obtained by the scene camera. On the other hand, in more recently developed computer interface-type trackers, the IR source and one camera are gimbal-mounted on a stationary platform containing the computer display. The user is not required to wear any devices. After a simple calibration procedure, the tracker can be used to record eye gaze position in the monitor display. Moreover, the user would be able to use the eye as a pointer in the same fashion one would use a mouse. Both systems allow the user to operate in a hands-free mode.

Two methods of illuminating the eye are in common use, dark-pupil and bright pupil. In the dark-pupil technique, the IR source is off-axis, causing light to be trapped in the interior of the eye. The pupil appears very dark in the eye image. On the other hand, if the IR illumination is provided coaxial to the eye, the eye appears to glow brightly. The dark pupil provides somewhat less contrast than the bright pupil for detecting iris boundaries. However, the bright pupil may interfere with the corneal reflex for relatively larger eye movements. Moreover, the bright pupil technique requires precise alignment of the incident and reflected light (which is consistent with our experience at the Vision Laboratory of the Rutgers University Center for Advanced Information Processing [CAIP]). Therefore, for measurements that allow for relatively free movement of the participant, the dark pupil technique is preferred.

Video-based eye trackers capture images of the eye using a standard RS170, 30 frames/sec video camera. Each frame is composed of two 512×240 interlaced fields that result in a 512×480 pixel image. However, since the tracker analyzes each field independently, images can be captured at a field rate of 60 times/sec. This relatively rapid recording of data allows for separation of fixations from saccades. The eye movement results can be plotted as a scan path, that is, locations of the sequence of fixations in the scene and paths of the saccades connecting them. Scan path plots are useful as indicators of the human observer's regions of interest during a visual task. The fixation locations derived from the data are also useful for eye–mouse-type pointing and menu selection operations described in section 4.

To make these systems work reliably, a number of technical issues need to be addressed. For example, gimbal-mounted eye trackers attached to computer displays must:

1. Lock on to and track the eye as the user's head moves about in the space in front of the display (usually a rather confined volume of space).
2. Maintain the image of the eye in sharp focus for reliable computation of the centroids using some form of autofocus.
3. Measure and keep track of the distance of the eye from the screen in order to compute the visual angle subtended by the screen (required to maintain pointing accuracy).
4. Be easily calibrated.

FIG. 9.1a. Head-mounted eye tracker.

Head-mounted trackers must:

1. Strictly maintain their position on the head or be able to recalibrate themselves with small shifts in position.
2. Provide for untethered operation (in most applications), either by RF links for the tracker and the magnetic head position sensor or by integration into wearable computers.

The value of this technology as an input mechanism will be closely linked with the degree to which problems associated with these issues are solved. Examples of head-mounted and computer-display-mounted eye trackers are shown in Fig. 9.1 (courtesy of ISCAN, Inc.).

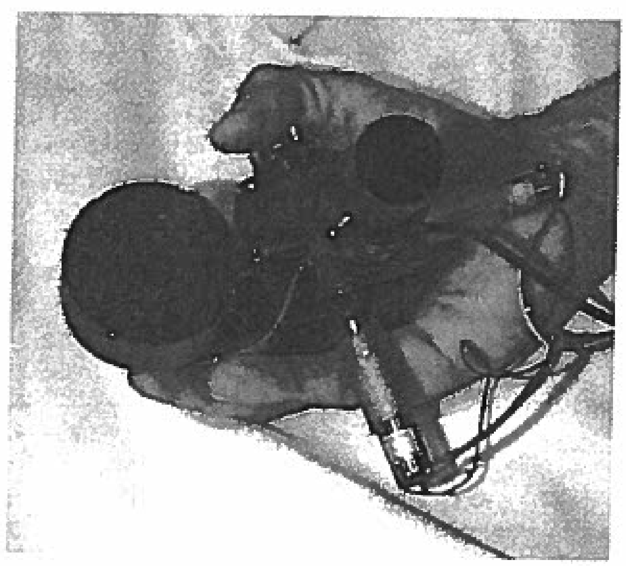

FIG. 9.1b. Gimbal-mounted eye tracker to be affixed to a computer display.

3. THE USE OF EYE TRACKERS AS A COMPUTER INPUT DEVICE

It is a common and often valid assumption that what a user is looking at on a computer screen is also the object the user wishes to select. Thus, people have attempted to use the eye tracker as an input device primarily for menu selection. Because the eye tends to rapidly saccade from place to place, object selection is defined by a dwell time of the eye resting on an object, usually 250 milliseconds. However, holding the eye in one position for 250 milliseconds is difficult for users. Dropping the dwell time to something more comfortable, e.g., 100 milliseconds leads to what is known as the "Midas" touch, that is, an overselection of screen objects. Notable successes have occurred with this type of input device for severely handicapped individuals, but everyday computer users find this form of input difficult and tiring (Jacob, 1991).

A key problem with using the eye as an input device for a computer system is that it already is being used as a visual input device for the human. As such, there are likely to be conflicts between the desired computer input behavior and the automatic human-eye behavior. Rather than requiring conscious control of the eye for selection, researchers have more recently looked at modeling the human-eye behavior in order to build intelligent heuristics for determining the computer input the user wants to make. A simple heuristic is that used by Zhai, Morimoto, and Ihde (1999). In their studies they assume that where the eye is looking is where the mouse cursor is best placed. No selection is made. Selection is still done by depressing the mouse button, but the user no longer has to move the mouse cursor a long distance to reach the desired selection target.

In another study by Tanriverdi and Jacob (2000), a threshold count of "landings" is used to deduce the desired selection object. It is assumed that if the eye saccades to a specific area of the screen a specified number of times, then the selection object in that area is the user's input. This heuristic is more complicated than simply counting the saccades over a selection object because a saccade may not even land on the desired object, only close enough to view it. Both of these mechanisms make inferences that may be undesired by the user so that, like the Midas touch of short eye dwell times, users may be faced with the extra effort of "undoing" the computer's decision making.

An alternate modality study, with speech recognition systems correcting speech recognition errors with additional speech, proved to be incredibly frustrating and time consuming for users (Karat, Halverson, Horn, & Karat, 1999). This is why both input designs (Tanniverdi & Jacob, 2000; Zhai, Morimoto, & Ihde, 1999) required a manual confirmation of a selection either with a mouse or Polhemus "click." A user could also gracefully exit from a computer inferred selection simply by looking elsewhere.

In addition to being a useful input mechanism when other methods are impractical or impossible (e.g., speech in a noisy environment), there is strong potential for combining input information from multiple modalities to further enhance the recognition of the information provided by any individual input device. For example, if gaze were combined with speech, it is possible that when a user says, "Put that there," the eye saccades to the location implied by "there" when the word is said. The timing interrelationship of the speech and the eye movement would then tell exactly where the "there" implied by the speech is located despite having a very noisy set of eye movements. The location of the eyes can also help untangle speech recognition. If a speaker said, "the red triangle," this could be understood as different from "the red quadrangle" simply by knowing that the speaker's eyes were saccading about a region of the screen containing the triangle not the quadrangle. Similar advantages might also be gleaned from looking at where a user is pointing or gesturing.

At this time, no natural, simple solution exists for using gaze alone for input selection. However, the above designs suggest that the use of gaze as an input mechanism is becoming more viable. Our own studies, and the two studies referenced above that make inferences from a user's natural eye behavior indicate that the design of a viable interaction mechanism is more complicated than simply using gross eye motor movements and dwell times. Such a mechanism may have to be trained to individual eye movement behaviors of the user as speech recognizers are now trained or be based on other basic eye movement behavior.

Head-mounted displays (HMDs) currently have limitations for gaze input because maintaining calibration is difficult due to slippage of the head-mounting device. Calibration is also a problem with a desk-mounted eye tracker, but the difficulties are not as severe. A gradual loss of calibration occurs over time, especially if the user is gazing at objects at the outer limits of the virtual scene. What is needed are mechanisms built into the VE that recalibrate the user periodically. Another calibration problem occurs if the user is moving to and fro during interaction. Face-tracking algorithms are being used (Yang, Stiefelhagen, Meier, & Waibel, 1998) to capture this motion and reset screen distance for the eye tracker. Overall, either head mounted or desk-mounted eye tracking devices are still difficult to use and take considerable fiddling, adjustment, and programming to interface with virtual environments. So, although promising as input devices, their English sports car tuning needs keep them from becoming over-the-counter input devices.

4. INTEGRATION OF EYE TRACKERS INTO MULTIMODAL VE SYSTEMS

Current human–machine communication systems rely, predominantly, on keyboard and mouse inputs that inadequately represent human capabilities for communication. More natural communication interfaces based on speech, sight, and touch can free computer users from the constraints of keyboard and mouse. Although these alternative modality interfaces are not currently sufficiently advanced to be used individually for robust human–machine interaction, they are sufficiently advanced to provide simultaneous multisensory exchange of information. Towards that end, the Rutgers University CAIP Center, under the National Science Foundation STIMULATE program (Flanagan et al., 1999), is conducting research to establish, quantify, and evaluate techniques for designing synergistic combinations of human–machine communication modalities. The modalities under study are sight, sound, and touch interacting in collaborative multiuser environments (see Fig. 9.2). The CAIP Center's goal is to design a multimodal HCI system with the following characteristics and components:

- Force-feedback tactile input and gesture recognition
- Automatic speech recognition, sound capture by microphone arrays and text to speech conversion
- Gaze and face tracking along with face verification for secure access
- language understanding and fusion of multimodal information (part of multimodal input management)
- Image understanding with region of interest detection and visual feature extraction
- Applications for collaborative work and design problems

As presently implemented, the current multimodal system incorporates the following components and characteristics:

- Force-feedback tactile input and gesture recognition
- Gaze tracking by a desk-mounted eye tracker, sound capture by a microphone array, automatic speech recognition, and a speech synthesizer
- Language understanding and fusion of multimodal information using a multimodal input manager

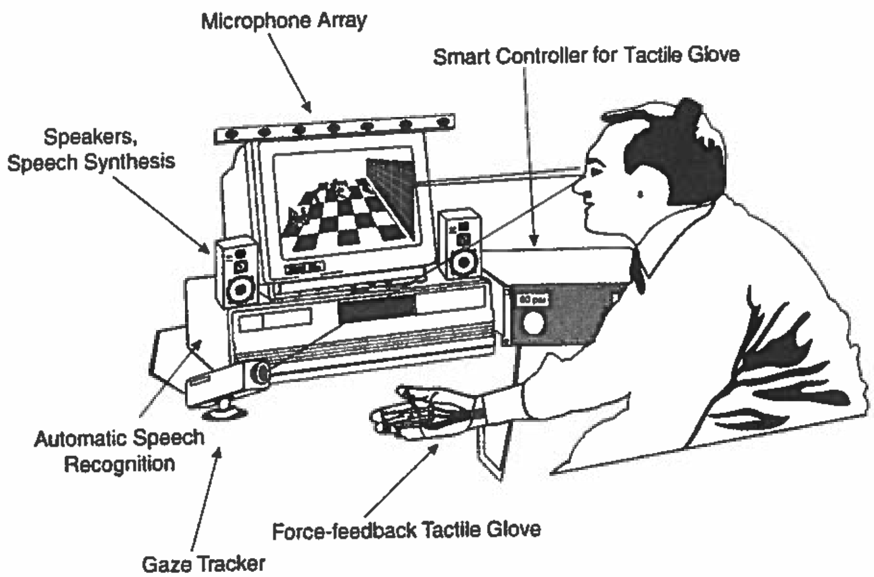

FIG. 9.2. CAIP Center multimodal system.

- Collaborative desktop used in an application requiring manipulation of 2-D and three-dimensional (3-D) graphical objects and icons on topographical maps (crisis management/disaster relief)
- Application for collaborative diagnosis of medical images

The integration of different input devices into one multimodal system also involves the integration of different platforms and languages. The contribution of gaze input to the multimodal system is to provide gaze directed hands-free visual communication. In the current implementation, gaze serves as a mouse-type pointer used in conjunction with the speech recognizer for issuing and carrying out commands like "Select it" and "Drop it." The gaze tracker used is a desk-mounted, nonintrusive ISCAN RK-726. It consists of a gimbal-mounted camera, an IR light source, and an ultrasonic depth sensor. The camera and IR source are used to compute the centroids of the pupil and the corneal reflection. The ultrasound sensor measures the distance of the eye from the screen to focus the camera lens automatically and to compute the visual angle subtended by the display screen.

The electronics for processing image data, carrying out tracking operations, and storing and transmitting data are mounted on a PCI bus circuit board installed in a Pentium personal computer (PC). This PC provides not only the coordinates of where the user is looking, but also the pupil diameter. It is possible to store gaze patterns and analyze them after the session. The gaze tracker PC is connected to a SUN Ultra workstation via a serial port connection. The main computer for the multimodal system is a Pentium Pro PC, which receives all multimodal inputs from the gaze tracker, tactile glove, and speech recognizer.

The main language used for the implementation of the multimodal system is Java because of its platform independence. An operating system (OS) level driver, Gaze Server, has been developed in the *C* programming language because of limited support for low-level programming in Java. The gaze server, which runs on the SUN Ultra workstation, reads the serial port from the eye tracker computer and writes gaze coordinates out to a socket. Since sockets are well supported in Java, a Java driver—Gaze Client—forms a TCP/IP (transmission control protocol/Internet protocol) connection with the gaze server to read the gaze coordinates. The coordinates of the eye position are then sent to the multimodal system after some preprocessing, which includes smoothing and translation of the coordinates to the appropriate screen location. The Gaze Handler in the modality manager and fusion agent uses these coordinates. The gaze handler, to give the user feedback on the accuracy of eye tracking, generates a gaze cursor on the screen. Similar OS level drivers, Java drivers, and modality handlers exist for all the independent modalities in the system.

One of the challenges faced in the use of the eye tracker as a mouselike pointing device is that the human eye does not move in the same calm and controlled fashion as the hand-controlled mouse. The eyes jump around from spot to spot. Special filters and averaging techniques have been developed to make the movements appear smooth and natural (Andre, 1998). These smoothing functions have the potential of interacting with the human's eye control, causing users to adjust their movements based on what they see is happening with the cursor on the screen.

Tracking the eye, determining gaze location on the monitor, and drawing the cursor on the monitor is carried out in real time. Although sockets are used for the connection, no noticeable delay is seen in the gaze cursor implementation for the multimodal application, though sometimes, due to high network traffic, system response time increases causing loss of eye position data and jumpiness of the gaze cursor. For this reason, a slightly different system integration approach has been used for human factors studies of the eye tracker as an input device. The human-factors testing program runs on a SUN workstation and reads directly from a serial port connection from the eye-tracker computer, thus avoiding network delays.

Calibration accuracy is another critical aspect of gaze tracking. Being 1 or 2 degrees off track is very distracting to the user and can result in mistaken commands. With desk-mounted eye trackers, the user is allowed to move the head but the degree of freedom is limited to about 1 cubic foot. The requirements for freedom of head movement and accuracy of tracking are obviously interdependent. Gaze tracking accuracy has been observed to be very sensitive to head movement, and it is very easy to lose calibration. One simple solution to this problem is to recalibrate. During experiments, it has been seen that calibration only takes about 10 seconds for an experienced user. This is a reasonably short period, so it is not a burden on the user or the application. However, for recalibration to be used effectively as a solution to the accuracy drift, the application has to provide specific commands to automate the process of calibration. In addition, data capturing is subject to human-perception inaccuracies. Although electronic performance characteristics of the eye tracker show that it is accurate to a fraction of a pixel, in practice, when humans are used to calibrate the system the accuracy is off by nearly as much as 5 pixels or 0.45 degrees.

Another problem faced in the implementation of the gaze tracker as a pointing device is that the presence of the gaze cursor causes what can be termed *Cursor drift*. This happens when the user is distracted by the presence of the gaze cursor on the screen, and starts tracking this cursor, thereby drifting further away from the target. If the eye tracker is always accurate, this problem is solved because the need for a gaze cursor disappears, improving the naturalness of the system.

For current eye-tracking systems, stable performance requires that all eye-tracker parameters (e.g., maximum and minimum pupil size, maximum and minimum corneal reflection size) have to be accurately adjusted for each user.

5. AN EVALUATION OF THE EFFECTIVENESS OF EYE TRACKING AS AN INPUT MECHANISM

Implied in the efforts to use gaze as an input medium is a belief that using the eye's positioning gives a user faster performance than other input devices. Certainly, eye saccades occur much quicker than hand motions or speech utterances used by more traditional input devices, but producing an effectively engineered gaze input device that actually takes advantage of the eye's speed may be difficult in practice. Early studies on the use of an eye tracker (Ware & Mikaelian, 1987) indicate that performance time is nearly one-half that of a mouse but with curious speed/accuracy trade-offs based on differences in the design of the target acceptance methods (pressing a button, dwelling on the target, etc.). Performance with any input device is dependent on differences in the design of the device not just the choice of input modality. For example, target acquisition times vary in a pointing task using a mouse depending on the shape and number of buttons on the mouse.

One can expect to uncover similar and often large differences when using eye-tracking hardware that varies in minor ways in the design of the user interface. In addition, the merging of gaze with a multimodal system with many types of input devices can also be expected to affect the overall performance of gaze input. We have therefore undertaken a series of experiments to capture the performance times of gaze in our multimodal setup for a simple target selection task. The next few paragraphs provide an overview of studies that were run and the results. The reader can find a more detailed presentation of these studies in Lin, Kaur, Tremaine, Hung, and Wilder (1999).

The experiment required users to position their eyes in a start circle until a target circle appeared at random positions on the screen. Users were then to saccade as rapidly as possible to the target circle and hold their eyes within the target until the software (200 ms) had recognized target acquisition. Users were first given a set of practice trials followed by 64 experiment trials. The direction of the target, target size, and distance to travel to the target were varied. A regression analysis was performed on the data to fit it to a log linear model based on target distance and target size. Such a model was developed for each of the four target directions used. Typical motor studies show that a user's performance is directly proportional to the distance traveled and inversely proportional to the target size, i.e.:

$MT = a + b * \log_2 (2D/W)$ where

MT = movement time

a and b are constants dependent on the properties of the input device

D = distance traveled

W = width of the target

This model is known as Fitts' law (Fitts, 1966) and is the standard mechanism used to examine and compare performance characteristics of a multitude of computer input devices (Mackenzie, 1992). Because this model is based on an iterative succession of motor control signals that the brain sends out to guide motor function, it is not surprising that eye movement has been shown to follow the same model as other motor types of target acquisition tasks (Bahill & Stark, 1979). Fitting a target acquisition task to Fitts' law tells the input device designer many details about the user's performance and how the device design is affecting the performance. For example, it may be that a log linear model does not fit the data well and that a linear model does. This implies that the motion that the user is making is not ballistic (i.e., does not have an acceleration and deceleration component) and that the user is required to exercise more guided motor control throughout the task. This type of performance is seen with pull-down menus, where users need to keep the cursor within the confines of the menu. It may also be that the constant a is very large. This implies that the device carries a cognitive load with it (e.g., the device may require the coordination of multiple muscle groups to start target acquisition, such as in using a glove to point to a target). Finally, the value b may be high, indicating that the performance of the input device deteriorates quickly as either the distance traveled increases or the target size decreases. Such performance occurs, for example, when a user is required to do multiple motor tasks simultaneously (e.g., holding down a mouse button in order to drag an object to a target).

The data resulting from our experiment were examined to evaluate the ISCAN eye-tracker input device used. When the experimental data were fit to a log linear model, an extremely poor correlation of 30% was achieved. This correlation was so low compared to the 90% Fitts found in his motor tasks that the data were examined at a low level, looking at each point captured by the eye-tracking device instead of the start circle leaving and target circle arriving time. The data showed that fixations as well as saccades were being included in the measures, with the user making small fixations just outside the start circle and/or just before they reached the target circle. A velocity threshold of 13 cm/sec was used to distinguish saccades from fixations and the fixations were discarded as error data. Overall, eight types of errors that occurred in the experiment were categorized. These included:

- Overshoots: Participant leaves the target circle after entering it.
- Lost gaze position: Eye tracker lost the gaze position due to a hardware failure or because the participant looked out of the boundary of the screen.
- Tick larger than 100 milliseconds due to workstation slow response.
- No saccade at all (i.e., velocity never exceeds the threshold).
- Velocity stays above saccade threshold in the target circle.
- First fixation point outside start circle.
- Second fixation point outside end circle.
- Out of envelope.

"Out of envelope" referred to a "wandering eye" that moved to the target circle in a curved path, moving far outside an envelope of acceptable efficient movements. Removing these variations in the data improved the Fitts' law correlation to 65%. There was, however, wide variation among participants. The Fitts' law model does not fit gaze as well as other pointing devices because the data show that participants are not making single clean gaze movements as they do with other devices. The resulting model predicts that roughly, for distances of 10 cm, gaze is 1.7 times faster than recent mouse designs. For distances of 20 cm, gaze selection performance is 2.1 times faster. The performance advantage can be found in the speed of the saccades, not in the cognitive time it takes to begin the task. The values of a for different mouse designs and gaze were comparable, as were the slopes of each performance model.

In summary, using gaze as an input mechanism is potentially quite promising. It is definitely faster than other input mechanisms, but its downside is that the eye, because of its dual usage, exhibits somewhat erratic performance. More detailed focus on eye behavior, such as using velocity thresholds to distinguish between saccades and fixations, is likely to improve the design of gaze input devices.

6. FUTURE DIRECTIONS

Approximately every 5 years a researcher investigates the use of eye tracking as an input device. Their conclusions are mixed about its viability because of the problems that are encountered in accurately interpreting data. Eye-tracking devices also improve significantly every 5 years so that both their cost and invasiveness are dropping. Our studies of gaze as an input mechanism, like prior studies, show large performance advantages for gaze. They also suggest that more considered design of input algorithms for processing eye-tracker data may circumvent problems facing the use of this mechanism as an input device. The combination of an eye tracker with other input modalities also indicates useful advantages that can be exploited in the interpretation of input from any one of the modalities.

7. CURRENT EYE-TRACKING RESEARCH IN VIRTUAL ENVIRONMENTS

Research in VEs is a growing and fluid field, so we hesitate to generate a list of research institutes or individuals that are conducting eye-tracking research in virtual environments. Also, given that this is a chapter in a book, no claims are made for such a list's validity over time. Below a set of key individuals and their research institutes currently involved in this work is provided. No claim is made either for the exhaustiveness or continued validity of this list. Individuals who are interested in obtaining further information on this area may want to look at the proceedings of the eye-tracking symposium that is jointly sponsored by ACM SIGCHI and ACM SIGGRAPH. The symposium takes place in late fall and the proceedings can be found in the Association for Computing Machinery's (ACM) digital library: http://www.acm.org/dl.

Researchers	*Institute*
Colin Ware	Department of Computer Science University of New Hampshire United States
Robert Jacob	Department of Computer Science Tufts University United States
William Krebs	Operations Research Department Naval Postgraduate School United States
Robert S. Kennedy	RSK Assessments, Inc. United States
Boris M. Velichovsky	Unit of Applied Cognitive Research Dresden University of Technology Germany
John Paulin Hanson	Risoe National Laboratory Denmark
Roel Vertegaal	Department of Computer Science Queens University Canada
Jie Yang, Alex Waibel	College of Computing Carnegie-Mellon University United States
Andrew T. Duchowski	Department of Computer Science Clemson University United States

8. REFERENCES

Andre, M. (1998). Multimodal human–computer interaction—The design and implementation of a prototype. Unpublished master's Thesis, Delft University of Technology, the Netherlands.

Bahill, T., & Stark, L. (1979). The trajectories of saccadic eye movements. *Scientific American, 240*(1), 108–117.

Cornsweet, T. N., & Crane, H. D. (1973). Accurate two-dimensional eye tracker using first and fourth Purkinje images. *Journal of the Optical Society of America, 63*, 921–928.

Eberhard, K. M., Spivey-Knowlton, M. J., Sedivy, J. C., & Tanenhaus, M. K. (1995). Eye movements as a window into real-time spoken language comprehension in natural contexts. *Journal of Psycholinguistic Research, 24*(6), 409–436.

Fitts, P. M. (1966). Cognitive aspects of information processing III: Set for speed versus accuracy. *Journal of Experimental Psychology, 71*, 849–857.

Flanagan, J., Kulikowski, C., Marsic, I., Burdea, G., Wilder, J., & Meer, P. (1999). Synergistic multimodal communication in collaborative multiuser environments. (Tech. Rep. NSF-STIMULATE). Rutgers University, CAIP Center.

Freedman, M. (1984). Eye-movement communication systems for computer control. In *Proceedings of the International Congress on Technology and Technology Exchange: Technology and the World Around Us* (pp. 34–35). Pittsburgh, PA: IEEE Press.

Jacob, R. J. K. (1991). The use of eye movements in human–computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems, 9*(3), 152–169.

Kaiser, J. P. (1999). Sensory adaptation effects following exposure to a virtual environment. Unpublished master's thesis). Monterey, CA: Naval Postgraduate School.

Karat, C.-M., Halverson, C., Horn, D., & Karat, J. (1999). Patterns of entry and correction in large vocabulary continuous speech recognition system. In *Proceedings of the CHI '99 Conference on Human Factors in Computing Systems* (pp. 568–575). New York: Addison-Wesley/ACM Press.

Lin, W., Kaur, M., Tremaine, M., Hung, G., & Wilder, J. (1999). Performance analysis of an eye-tracker. In *Proceedings of the SPIE Conference on Machine Vision Systems for Inspection and Metrology VIII* (pp. 106–113).

Lohse, G. L., & Johnson, E. J. A.(1996). Comparison of two process-tracing methods for choice tasks. *Organizational Behavior and Human Decision Processes, 68*(1), 28–43.

Mackenzie, I. S. (1992). Fitt's law as a research and design tool in human–computer interaction. *Human–Computer Interaction, 7*, 91–139.

Megaw, E. D., & Richardson, J. (1979). Eye movements and industrial inspection. *Applied Ergonomics, 10*, 145–154.

Remmel, R. S. (1984). An inexpensive eye-movement monitor using the scleral search coil technique. *IEEE Trans. Biomed. Engin., 31*, 388–390.

Reuter, B. W., & Schenck, U. (1985). Data acquisition in visual screening of microscopical samples. In *Proceedings of Image Science '85 Acta Polytechnica Scandinavica, Applied Physics Series* (pp. 206–209).

Russo, J. E., & LeClerc, F. (1994). An eye-fixation analysis of choice processes for consumer nondurables. *Journal of Consumer Research, 21*, 274–290.

Sanders, M. G., Simmons, R. R., & Hoffman, M. A. (1979). Visual workload of the copilot/navigator during terrain flight. *Human Factors, 21*(3), 369–383.

Tanriverdi, V., & Jacob, R. J. K. (in press). Interacting with eye movements in virtual environments. In *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, New York: Addison-Wesley/ACM Press.

Velichkovsky, B. M., & Hansen, J. P. (1996). New technological windows into mind: There is more in eyes and brains for human–computer interaction. In *Proceedings of CHI '96: Human Factors in Computing Systems* (pp. 496–503). New York: ACM Press.

Ware, C., & Mikaelian, H. T. (1987). An evaluation of an eye tracker as a device for computer input. In *Proceedings of the ACM CHI+GI '87 Conference on Human Factors in Computing Systems* (pp. 183–188). New York: ACM Press.

Yamada, M., & Fukada, T. (1987). Eye word processor (EWP) and peripheral controller for ALS patient. In *Proceedings of IEEE, part A: Physical science, measurement and instrumentation, management and education, Reviews* (pp. 328–330).

Yang, J., Stiefelhagen, R., Meier, U., & Waibel, A. (1998). Visual tracking for multimodal human–computer interaction. In *Proceedings of CHI '98* (pp. 140–147).

Young, L. R., & Sheena, D. (1975). Methods and designs—Survey of eye-movement recording methods. *Behavior Research Methods and Instrumentation, 7*(5), 397–429.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the CHI '99 Conference on Human Factors in Computing Systems* (pp. 246–253). New York: Addison-Wesley/ACM Press.

HANDBOOK OF VIRTUAL ENVIRONMENTS
Design, Implementation, and Applications

Edited by

Kay M. Stanney
University of Central Florida

LEA LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
2002 Mahwah, New Jersey London