

Evaluation of Eye Gaze Interaction

Linda E. Sibert

Human-Computer Interaction Lab
NCARAI
Naval Research Laboratory
Washington DC 20375 USA
202 767-0824
sibert@itd.nrl.navy.mil

Robert J.K. Jacob

Department of Electrical Engineering &
Computer Science
Tufts University
Medford MA 02155 USA
617 627-3217
jacob@eecs.tufts.edu

ABSTRACT

Eye gaze interaction can provide a convenient and natural addition to user-computer dialogues. We have previously reported on our interaction techniques using eye gaze [10]. While our techniques seemed useful in demonstration, we now investigate their strengths and weaknesses in a controlled setting. In this paper, we present two experiments that compare an interaction technique we developed for object selection based on a where a person is looking with the most commonly used selection method using a mouse. We find that our eye gaze interaction technique is faster than selection with a mouse. The results show that our algorithm, which makes use of knowledge about how the eyes behave, preserves the natural quickness of the eye. Eye gaze interaction is a reasonable addition to computer interaction and is convenient in situations where it is important to use the hands for other tasks. It is particularly beneficial for the larger screen workspaces and virtual environments of the future, and it will become increasingly practical as eye tracker technology matures.

Keywords

Eye movements, eye tracking, user interfaces, interaction techniques

INTRODUCTION

We describe two experiments that compare our eye gaze object selection technique with conventional selection using a mouse. We have previously found that people perform well with eye gaze interaction in demonstrations. The next step is to show that our technique can stand up to more rigorous use and that people are comfortable selecting objects using eye gaze over a more extended period of time. We compare the performance of eye gaze interaction with that of a widely used, general-purpose device: the mouse. Eye gaze interaction requires special hardware and software. The question is whether it is worth the extra effort. If it performs adequately, we can also gain some

hard-to-quantify side benefits of an additional, passive or lightweight input channel. For example, we have found that when eye gaze interaction is working well, the system can feel as though it is anticipating the user's commands, almost as if it were reading the user's mind. It requires no manual input, which frees the hands for other tasks. A reasonable definition of performing well is if eye gaze interaction does not slow down interaction and can "break even" with the mouse in a straightforward experimental comparison, despite the immaturity of today's eye tracker technology. If the eye gaze interaction technique is faster, we consider it a bonus, but not the primary motivation for using eye tracking in most settings.

Our experiments measured time to perform simple, representative direct manipulation computer tasks. The first required the subject to select a highlighted circle from a grid of circles. The second had the subject select the letter named over an audio speaker from a grid of letters. Our results show a distinct, measurable speed advantage for eye gaze interaction over the mouse in the same experimental setting, consistently in both experiments.

The details of the experiment give insight into how our eye gaze interaction technique works and why it is effective. It is not surprising that the technique is somewhat faster than the mouse. Our research tells us the eye can move faster than the hand. The test of our approach is how our entire interaction technique and algorithm preserves this speed advantage of the eye in an actual object selection task. We studied the physiology of the eye and used that information to extract useful information about the user's higher-level intentions from noisy, jittery eye movement data. Even though our algorithm is based on an understanding of how eyes move, it was unclear that our eye gaze interaction technique would preserve the quickness of the eye because the eye tracking hardware introduces additional latencies. Performance of any interaction technique is the product of both its software and hardware. The experiments show that we have been successful.

RELATED WORK

People continuously explore their environment by moving their eyes. They look around quickly and with little conscious effort. With tasks that are well-structured and

CHI '2000 The Hague, Amsterdam
1-58113-216-6/00/04

speeded, research has shown that people look at what they are working on [17]; the eyes do not wander randomly. Both normal and abnormal eye movements have been recorded and studied to understand processes like reading [16] and diagnosing medical conditions (for example, a link between vestibular dysfunction and schizophrenia shows up in smooth pursuit). People naturally gaze at the world in conjunction with other activities such as manipulating objects; eye movements require little conscious effort; and eye gaze contains information about the current task and the well-being of the individual. These facts suggest eye gaze is a good candidate computer input method.

A number of researchers have recognized the utility of using eye gaze for interacting with a graphical interface. Some have also made use of a person's natural ways of looking at the world as we do. In particular, Bolt suggests that the computer should capture and understand a person's natural modes of expression [5]. His *World of Windows* presents a wall of windows selectable by eye gaze [4, 6]. The object is to create a comfortable way for decision-makers to deal with large quantities of information. A screen containing many windows covers one wall of an office. The observer sits comfortably in a chair and examines the display. The system organizes the display by using eye gaze as an indication of the user's attention. Windows that receive little attention disappear; those that receive more grow in size and loudness. Gaze as an indication of attention is also used in the self-disclosing system that tells the story of *The Little Prince* [23]. A picture of a revolving world containing several features such as staircases is shown while the story is told. The order of the narration is determined by which features of the image capture the listener's attention as indicated by where he or she looks.

Eye gaze combined with other modes can help disambiguate user input and enrich output. Questions of how to combine eye data with other input and output are important issues and require appropriate software strategies [24]. Combining eye with speech using the OASIS system allows an operator's verbal commands to be directed to the appropriate receiver, simplifying complex system control [8]. Ware and Mikaelian [25] conducted two studies, one that investigated three types of selection methods, the other that looked at target size. Their results showed that eye selection can be fast provided the target size is not too small. Zhai, Morimoto, and Ihde [27] have recently developed an innovative approach that combines eye movements with manual pointing.

In general, systems that use eye gaze are attractive because people naturally look at the object of interest. They are used to performing other tasks while looking, so combining eye gaze interaction with other input techniques requires little additional effort.

DEMONSTRATION SYSTEM AND SOFTWARE ARCHITECTURE

Incorporating eye gaze into an interactive computer system requires technology to measure eye position, a finely tuned computer architecture that recognizes meaningful eye gazes in real time, and appropriate interaction techniques that are convenient to use. In previous research, we developed a basic testbed system configured with a commercial eye tracker to investigate interfaces operated by eye gaze [9, 10, 11, 12, 13, 14]. We designed a number of interaction techniques and tested them through informal trial and error evaluation. We learned that people prefer techniques that use natural not deliberate eye movements. Observers found our *demonstration eye gaze interface fast, easy, and intuitive*. In fact, when our system is working well, people even suggest that it is responding to their intentions rather than to their explicit commands. In the current work, we extended our testbed and evaluated our eye gaze selection technique through a formal experiment.

Previous work in our lab has demonstrated the usefulness of using natural eye movements for computer input. We have developed interaction techniques for object selection, data base retrieval, moving an object, eye-controlled scrolling, menu selection, and listener window selection. We use context to determine which gazes are meaningful within a task. We have built the demonstration system on top of our real-time architecture that processes eye events. The interface consists of a geographic display showing the location of several ships and a text area to the left (see Figure 1) for performing four basic tasks: selecting a ship, reading information about it, adding overlays, and repositioning objects.

The software structure underlying our demonstration system and adapted for the experiments is a real-time architecture that incorporates knowledge about how the eyes move. The algorithm processes a stream of eye position data (a datum every 1/60 of a sec.) and recognizes meaningful events. There are many categories of eye movements that can be tapped. We use events related to a saccadic eye movement and fixations, the general mechanism used to search and explore the visual scene. Other types of eye movements are more specialized and might prove useful for other applications, but we have not made use of them here. For example, smooth pursuit motion partially stabilizes a slow moving target or background on the fovea and optokinetic nystagmus (i.e., train nystagmus) has a characteristic sawtooth pattern of eye motion in response to a moving visual field containing repeated patterns [26]. These movements would not be expected to occur with a static display.

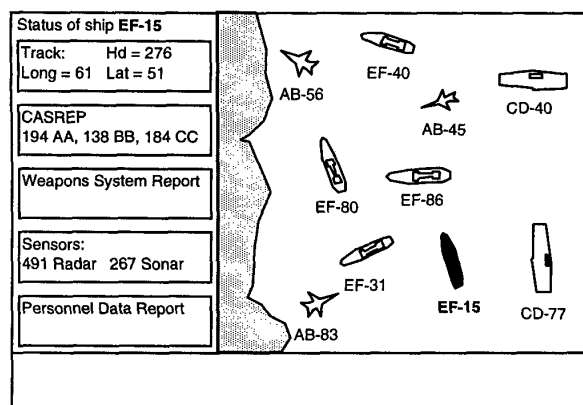


Figure 1. Display from eye tracker demonstration system. Whenever a user looks at a ship in the right window, the ship (highlighted) is selected and information about it is displayed in the left window.

The eyes are rarely still because, in order to see clearly, we must position the image of an object of interest on our fovea, the high-acuity region of the retina that covers approximately one degree of visual arc (an area slightly less than the width of the thumb held at the end of the extended arm). For normal viewing, eyes dart from one fixation to another in a saccade. Saccades are the rapid ballistic movements of the eye from one point of interest to another, whose trajectory cannot be altered once begun. During a saccadic eye movement, vision is suppressed. Saccades take between 30 and 120 msec. and cover a range between 1 to 40 degrees of visual angle (average 15 to 20 degrees). The latency period of the eye before it moves to the next object of interest is at least 100 to 200 msec., and after a saccade, the eyes will fixate (view) an object between 200 to 600 msec. Even when a person thinks they are looking steadily at an object, the eyes make small, jittery motions, generally less than one degree in size. One type is high frequency tremor. Another is drift or the slow random motion of the eye away from a fixation that is corrected with a microsaccade. Microsaccades may improve visibility since an image that is stationary on the retina soon fades [3]. Likewise, it is difficult to maintain eye position without a visual stimulus or to direct a fixation at a position in empty space.

At the lowest level, our algorithm tries to identify fixation events in the data stream and records the start and approximate location in the event queue. Our algorithm is based on that used for analyzing previously recorded files of raw eye movement data [7, 18] and on the known properties of fixations and saccades. A new requirement is that the algorithm must keep up with events in real time. The fixation recognition algorithm declares the start of a fixation after the eye position remains within approximately 0.5 degrees for 100 msec. (the spatial and temporal

thresholds are set to take into account jitter and stationarity of the eye). Further eye positions within approximately one degree are assumed to represent continuations of the same fixation. To terminate a fixation requires 50 msec. of data lying outside one degree of the current fixation. Blinks and artifacts of up to 200 msec. may occur during a fixation without terminating it. The application does not need to respond during a blink because the user cannot see visual changes because vision is suppressed.

Tokens for eye events - for start, continuation (every 50 msec. in case the dialogue is waiting to respond to a fixation of a certain duration), end of a fixation, raw eye position (not used currently), failure to locate eye position for 200 msec., resumption of tracking after failure, and entering monitored regions (a strategy typically used for mouse interaction) - are multiplexed into the same event queue stream as those generated by other input devices. These tokens carry information about the screen object being fixated. Eye position is associated with currently displayed objects and their screen extents using a nearest neighbor approach. The algorithm will select the object that is reasonably close to the fixation and reasonably far from all other objects. It does not choose when the position is halfway between two objects. This technique not only improves performance of the eye tracker (which has difficulty tracking at the edges of the screen, see discussion of the range of the eye tracker in the *Apparatus* section) but also mirrors the accuracy of the fovea. A fixation does not tell us precisely where the user is looking because the fovea (the sharp area of focus) covers approximately one degree of visual arc. The image of an object falling on any part of the fovea can be seen clearly. Choosing the nearest neighbor to a fixation recognizes that the resolution of eye gaze is approximately one degree.

The interaction is handled by a User Interface Management System that consists of an executive and a collection of simple individual dialogues with retained state, which behave like coroutines. Each object displayed on the screen is implemented as an interaction object and has a helper interaction object associated with it that translates fixations into the higher unit of gazes. This approach is more than an efficiency. It reflects that the eye does not remain still but changes the point of fixation around the area of interest. (Further details on the software are found in [13].)

STUDY OF EYE GAZE VERSUS MOUSE SELECTION

In developing our demonstration system, we have been struck by how fast and effortless selecting with the eye can be. We developed the interaction techniques and software system after much studying, tinkering, and informal testing. The next step was to study the eye gaze technique under more rigorous conditions. For eye gaze interaction to be useful, it must hold up under more demanding use than a demonstration and operate with reasonable responsiveness. We conducted two experiments that compared the time to select with our eye gaze selection technique and with a

mouse. Our research hypothesis states that selecting with eye gaze selection is faster than selecting with a mouse.

Our hypothesis hardly seems surprising. After all, we designed our algorithm from a understanding of how eyes move. Physiological evidence suggests that saccades should be faster than arm movements. Saccades are ballistic in nature and have nearly linear biomechanical characteristics [1, 2, 20]. The mass of the eye is primarily from fluids and the eyeball can be moved easily in any direction, in general. In contrast, arm and hand movements require moving the combined mass of joints, muscles, tendons, and bones. Movement is restricted by the structure of the arm. A limb is maneuvered by a series of controlled movements carried out under visually guided feedback [21]. Furthermore, we must move our eyes to the target before we move the mouse.

However, we were not comparing the behavior of the eye with that of the arm in these experiments. We were comparing two complete interaction techniques with their associated hardware, algorithms, and time delays. For our research hypothesis to be true, our algorithm, built from an understanding of eye movements, plus the eye tracker which adds its own delay, must not cancel out the inherent speed advantage of the eye.

METHOD

To test whether eye gaze selection is faster than selecting with a mouse, we performed two experiments that compared the two techniques. Each experiment tried to simulate a real user selecting a real object based on his or her interest, stimulated by the task being performed. In both experiments, the subject selected one circle from a grid of circles shown on the screen. The first was a quick selection task, which measured "raw" selection speed. The circle to be selected was highlighted. The second experiment added a cognitive load. Each circle contained a letter, and the spoken name of the letter to be selected was played over an audio speaker. The two experiments differed only in their task. The underlying software, equipment, dependent measures, protocol, and subjects were the same.

Interaction Techniques

Our eye gaze selection technique is based on dwell time. We compared that with the standard mouse button-click selection technique found in direct manipulation interfaces. We chose eye dwell time rather than a manual button press as the most effective selection method for the eye based on previous work [9]. A user gazes at an object for a sufficiently long time to indicate attention and the object responds, in this case by highlighting. A quick glance has no effect because it implies that the user is surveying the scene rather than attending to the object. Requiring a long gaze is awkward and unnatural so we set our dwell time to 150 msec., based on previous informal testing, to respond quickly with only a few false positive detections. The mouse was a standard Sun mouse without acceleration.

EXPERIMENT 1: CIRCLE TASK

The task for the first experiment was to select a circle from a three by four grid of circles as quickly as possible (the arrangement is shown in Figure 2). The diameter of each circle was 1.12 inches. Its center was 2.3 inches away from its neighboring circles in the horizontal and vertical directions and about 3 inches from the edge of the 11 by 14 inch CRT screen.

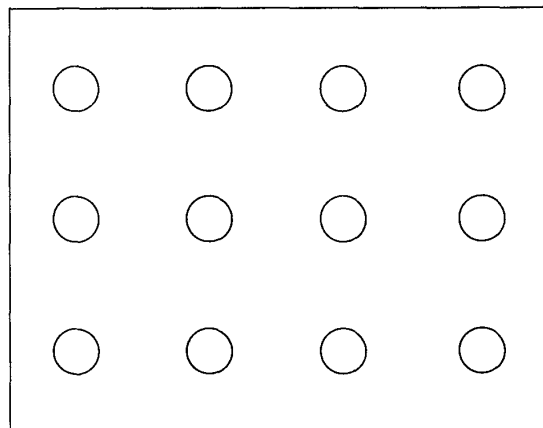


Figure 2. Screen from the circle experiment. The letter experiment has the same arrangement with the letters inscribed alphabetically in the circles, left to right, top to bottom.

Targets were presented in sequences of 11 trials. The first trial was used for homing to a known start position and was not scored. The target sets were randomly generated and scripted. One restriction was imposed that no target was repeated twice in a row. The same target scripts were presented to each subject. A target highlighted at the start of a trial; when it was selected, it de-highlighted and the next target in the sequence highlighted immediately. In this way, the end position of the eye or mouse for one trial became the start position for the next. No circle other than the target was selectable (although information about wrong tries was recorded in the data file). We presented the trials serially rather than as discrete trials to capture the essence of a real user selecting a real object based on his or her own interest. The goal was to test our interaction technique in as natural a setting as possible within a laboratory experiment.

Apparatus

The subject sat in a straight-backed stationary chair in front of a table (29.5 inches tall) that held a Sun 20-inch color monitor. The eye to screen distance was approximately three feet. The mouse rested on a 15-inch square table (28.5 inches tall) that the subject could position. The eye

tracker hardware and experimenter were located to the subject's left, which dictated that only individuals that use the mouse right-handed could be subjects (otherwise we would have had to rearrange the equipment and recalibrate). The operator stood in front of the eye tracker console to adjust the eye image when needed and control the order of the experiment. The subject wore a thin, lightweight velcro band around the forehead with a Polhemus 3SPACE Tracker sensor attached above the left eye, which allowed a little larger range of head motion with the eye tracker.

The eye tracker was an Applied Science Laboratories (Bedford, MA) Model 3250R corneal reflection eye tracker that shines an on-axis beam of infrared light to illuminate the pupil and produce a glint on the cornea. These two features - the pupil and corneal reflection - are used to determine the x and y coordinates of the user's visual line of gaze every 1/60 second. Temporal resolution is limited to the video frame rate so that some dynamics of a saccade are lost. The measurable field of view is 20 degrees of visual angle to either side of the optics, about 25 degrees above and about 10-degrees below. Tracking two features allows some head movement because it is possible to distinguish head movements (corneal reflection and center of pupil move together) from eye movements (the two features move in opposition to one another). We extended the allowable range that a subject could move from one square inch to 36 square inches by adding mirror tracking (a servo-controlled mirror allows +/-6 inches of lateral and vertical head motion). Mirror tracking allows automatic or joystick controlled head tracking. We enabled magnetic head tracking (using head movement data from the Polhemus mounted over the subject's left eye) for autofocus.

The position of gaze was transmitted to a stand-alone Sun SPARCserver 670 MP through a serial port. The Sun performed additional filtering, fixation, and gaze recognition, and some further calibration, as well as running the experiments. The mouse was a standard Sun optical mouse. Current eye tracking technology is relatively immature, and we did have some equipment problems, including the expected problem of the eye tracker not working with all subjects. Our eye tracker has difficulties with hard contact lenses, dry eyes, glasses that turn dark in bright light, and certain corneas that produce only a dim glint when a light is shown from below. Eye trackers are improving, and we expect newer models will someday solve many of these problems.

Our laboratory's standard procedure for collecting data is to write every timestamped event to disk as rapidly as possible for later analysis, rather than to perform any data reduction on the fly [15]. Trials on which the mouse was used for selection tracked the eye as well, for future analysis. We stored mouse motion, mouse button events, eye fixation (start, continuation, end), eye lost and found, eye gaze (start, continuation, end), start of experiment, eye and mouse wrong choices, eye and mouse correct choices, and

timeout (when the subject could not complete a trial and the experiment moved on). All time was in milliseconds, either from the eye tracker clock (at 1/60 sec. resolution) or the Sun system clock (at 10 msec. resolution). We isolated the Sun from our network to eliminate outside influences on the system timing.

Subjects

Twenty-six technical personnel from the Information Technology Division of the Naval Research Laboratory volunteered to participate in the experiment without compensation. We tested them to find 16 for whom the eye tracker worked well. All had normal or corrected vision and used the mouse right-handed in their daily work (required because the eye tracker and experimenter occupied the space to the left). All participants were male, but this was not by design. The four women volunteers fell into the group whom the eye tracker failed to track, though women have successfully used our system in the past. The major problems were hard contact lenses and weak corneal reflections that did not work well with our system.

Procedure

Each subject first completed an eye tracker calibration program. The subject looked, in turn, at a grid of nine points numbered in order, left to right, top to bottom. This calibration was checked against a program on the Sun and further adjustments to the calibration were made, if needed, by recording the subject's eye position as they looked at 12 offset points, one at each target location. These two steps were repeated until the subject was able to select all the letters on the test grid without difficulty. The subject then practiced the task, first with the mouse and then the eye gaze selection technique. The idea was to teach the underlying task with the more familiar device. The subject completed six sets of 11 trials (each including the initial homing trial) with each interaction device. Practice was followed by a 1.5 minute break in which the subject was encouraged to look around; the eye was always tracked and the subject needed to move away from the infrared light of the eye tracker (the light dries the eye, but less than going to the beach) as well as to rest from concentrating on the task. In summary, the targets were presented in blocks of 66 (six sequences of 11), mouse followed by eye. All subjects followed the same order of mouse block, eye block, 1.5 minute rest, mouse block, eye block. Because of difficulties with our set-up, we chose to run only one order. We felt this to be an acceptable, although not perfect solution, because the two techniques use different muscle groups, suggesting that the physical technique for manipulating the input should not transfer. Because of blocking in the design, we were able to test for learning and fatigue. Each experiment lasted approximately one hour.

Results

The results show that it was significantly faster to select a series of circle targets with eye gaze selection than with a mouse. The mean time for selection is shown in Figure 3.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.