EyeDraw: A System for Drawing Pictures with the Eyes

Anthony Hornof, Anna Cavender, and Rob Hoselton
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403 USA
+1 541 346 1372
{hornof, acavende, robhoz}@cs.uoregon.edu

ABSTRACT
This paper describes the development of EyeDraw, a software system that enables children with severe motor impairments to draw pictures by just moving their eyes. EyeDraw will help these children to have creative and developmental experiences currently missing from their lives. The project demonstrates how task analysis integrated at various levels of detail, including that of the unit task as well as that of visual-perceptual and oculomotor processing, can improve eye tracking for real-time input. The project introduces refined techniques for controlling a computer with the eyes. The paper discusses the motivation for the project, previous research on eye-control of computers, how EyeDraw works, and the results of user observation studies that are currently in progress.

Author Keywords
Children, eye tracking, input devices, interaction techniques, universal access.

ACM Classification Keywords
H.5.2. User Interfaces: Input devices and strategies, interaction styles.

INTRODUCTION
The field of human-computer interaction (HCI) uses eye tracking for two purposes: retrospective analysis and real-time input. For retrospective analysis, eye movement data are studied *post hoc* to evaluate usability issues and understand human performance. For real-time input, the computer in some way responds to eye movements made while using the computer [6]. This paper discusses a use of eye tracking for real-time input. Specifically, it discusses the design and implementation of EyeDraw, a system that will enable children with severe mobility impairments—children who can only move their eyes—to draw pictures with their eyes and thus benefit from the same creative and social activities as nondisabled children. The system is demonstrated to be usable by nondisabled children with no prior experience using an eye tracker. User observation studies with disabled children are currently in progress.

Several eye tracking systems have been designed to assist people with motor impairments that result from conditions such as cerebral palsy or Amyotrophic Lateral Sclerosis (ALS). Systems include Quick Glance (eyetechds.com), VisionKey (eyecan.ca), and the Eyegaze Communication System (eyegaze.com). Hundreds of people use these systems to communicate, such as by looking at letters to type with their eyes. The Eyegaze Communication System offers perhaps the most functionality, with software for uttering phrases via a speech synthesizer, making telephone calls, controlling lights and appliances, and turning pages in electronic books.

Better eye-controlled software is needed, especially for children with severe disabilities who miss out on creative and developmental activities such as drawing. EyeDraw addresses this need. Focusing the task analysis and design for EyeDraw on this specific user group and task also appears to be leading to useful new eye-control interaction techniques and improved use of eye trackers for real-time input. There is a history of technological innovation for universal access resulting in technology that is useful for the general public, such as with curb ramps, audio cassette recorders, and remote control devices for televisions [7].

PREVIOUS RESEARCH
HCI researchers have studied eye tracking for real-time input since the inception of the field. The first CHI conference had a paper on the topic [1]. Overall, eye tracking in HCI has had limited success because eye tracking is technically challenging and labor-intensive, and because eye movement data are difficult to interpret [6]. However, improvements in the accuracy and ease-of-use of eye trackers are contributing to new successes.

Previous research on how and why children draw pictures is useful in the design and development of EyeDraw. The research provides (a) evidence that a computer-based drawing program can provide important developmental experiences, (b) guidance for designing the most beneficial eye-drawing experience, and (c) a framework for evaluating the progression of drawings that children will make with their software. Children have been observed progressing through a series of five qualitative stages when beginning to draw with paper and pencil: random scribble, controlled scribble, basic forms, early pictorial, and later pictorial. Children follow the same stages of development when learning to draw on computers [4]. This taxonomy provides a framework for categorizing and evaluating the drawings of children using EyeDraw. EyeDraw emphasizes open-ended, spontaneous drawing because this has been shown to be better at inspiring creativity and self-expression than recipe art lessons and coloring-in drawings [3].

Previous interaction techniques for drawing with the eyes use *free-eye drawing.* In free-eye drawing, screen pixels are painted wherever the eye tracker records the gaze on the screen. Figure 1 shows free-eye drawings from Tchalenko [8]. Another free-eye drawing system is EaglePaint [5]. Both systems have assisted with drawings that would be categorized as in the "scribble" stages of drawing, but not in the basic forms or pictorial stages. Children have not evidently used the systems to draw recognizable objects and scenes such as houses, people, cars and trees.

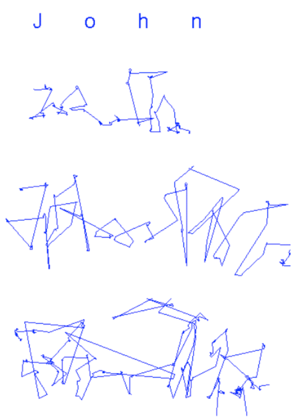

Figure 1. Three attempts to free-eye draw the name "John" from Tchalenko [8].

The difficulties in free-eye drawing can be explained in part based on the characteristics of human visual perception and oculomotor (eye movement) processing. First, free-eye drawing jams together two task activities that are usually independent when drawing a picture: viewing the drawing, and drawing the lines. Second, people do not have the same control over their eyes as over their hands and other limbs. People can move their eyes in quick bursts, but not slow adjusting movements. Alternative input techniques are needed for drawing with the eyes.

HOW EYEDRAW WORKS

Drawing with the eyes must be designed and accomplished at both the unit-task level of analysis [2] as well as at the visual-perceptual and oculomotor subtask level. To understand how EyeDraw works, a few terms pertaining to eye movements and eye tracking must be defined. The *gaze* is the vector that goes from the eye to the *gazepoint*, which is the point in a scene where a person is looking. The eyes (and thus the gaze) move around a static scene with a series of quick jumps called *saccades*, each of which lasts roughly 30 ms. Between saccades, the eyes (and the gazepoint) stay at the same location (with a slight tremor) for a *fixation* that lasts roughly 100 to 400 ms. A *dwell* is a long fixation. The eyes move, in short, to put items of interest into the high resolution vision which is at the center of their gaze.

The Eyegaze eye tracker uses the common pupil-center corneal-reflection technique, and reports the gazepoint on the computer screen 60 times per second, or once every 16.7 ms. EyeDraw averages the location of every six consecutive gazepoints reported by the eye tracker and displays them on the screen as the *eye cursor.* The eye cursor is a colored square (seven pixels wide) that dances around the screen wherever the user puts their eyes, with a small, roughly 133 ms delay.

Figure 2 shows the three modes that the cursor moves through along the way to the user issuing a drawing command. The first *Just Looking* mode uses a green cursor. As long as the user keeps moving their eyes around, the cursor will stay green. If the gaze dwells at a location for a minimum amount of time, the program enters a *Ready to Draw?* mode and the cursor changes to yellow. Based on user testing, the dwell time should initially be set to 500 ms, but should be decreased after practice.

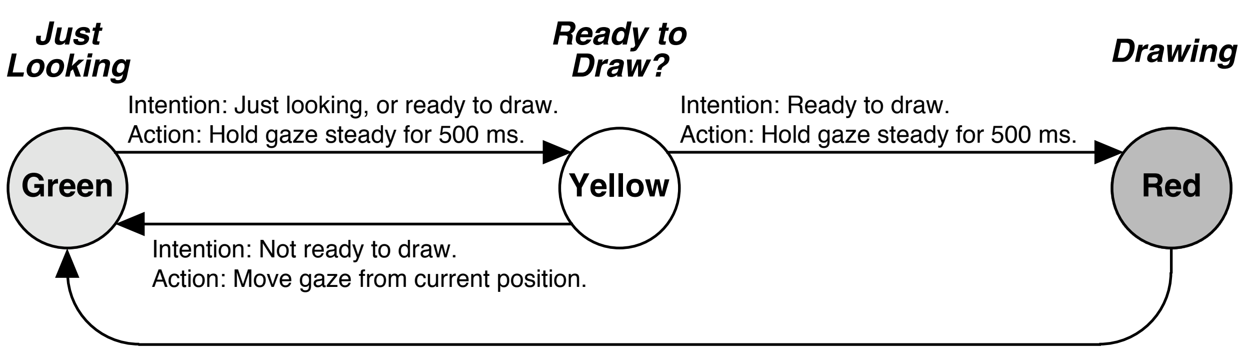

Figure 2. The transition of the user and the eye-cursor from the mode of *Just Looking* to the mode of *Drawing*. In the intermediary *Ready to Draw?* phase, the user can choose between going back to just looking, or moving forward with a drawing decision. Each transition is motivated by a user intention and action.

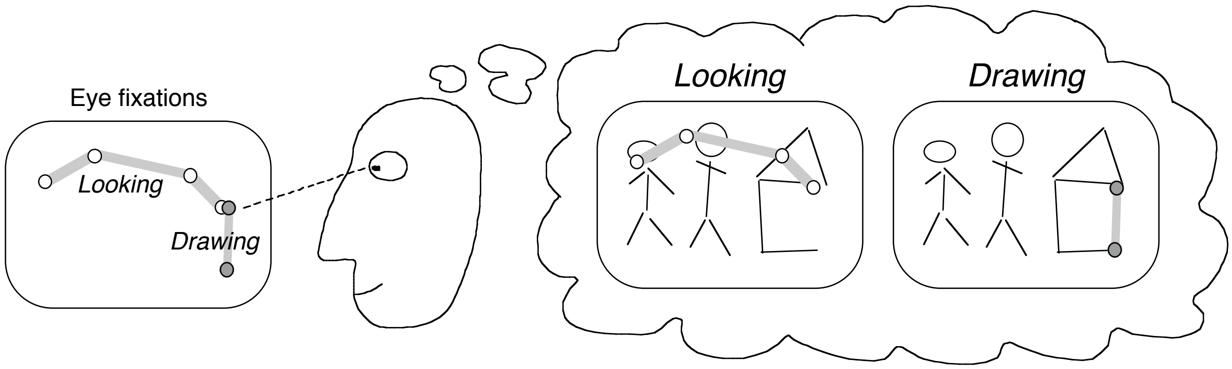

Figure 3. Without moving the gaze from the drawing, the user can shift between looking and drawing. The thick gray lines are eye movements. The circles are fixations. Gray circles are longer fixations that exceed a dwell threshold. Given this user input, EyeDraw would draw a line creating the right wall of the house.

Figure 3 demonstrates how an EyeDraw user can, while keeping their gaze on the picture, shift between using their eyes to (1) look at or study the drawing and (2) add to the drawing. This smooth subtask-switching is one of several differences between EyeDraw and previous free-eye drawing systems. The eye cursor usually stays green. When the cursor turns yellow, the user must decide whether to issue a drawing command, such as the start of a line. To stop the command from being issued, they need to move their eyes again to get back to the green *Just Looking* mode. To issue the command, they continue dwelling. After another 500 ms dwell, EyeDraw enters the *Drawing* mode and executes the command. The cursor briefly changes to red. Auditory feedback is also provided to confirm the drawing command was executed. The program then automatically returns to *Just Looking* mode.

The transition between the looking and drawing modes can be applied to a variety of drawing shapes, but only a line and circle are currently implemented. Rather than adding new shapes, we shifted our efforts into other basic functionality such as saving and retrieving drawings. This is in line with our efforts to get a usable, working prototype to users as early as possible.

Two sets of functionality for saving and retrieving are built into EyeDraw: (a) saving and retrieving the drawings in a scrapbook and (b) saving and replaying the eye-cursor commands that were used to create the drawings. The first feature enables users to save their drawings to a scrapbook to open and flip through later, all with their eyes. The second features is critical for the developers. Because our target users are geographically distributed all over the world, we will not be able to directly observe all of them using the software. The feature will allow us to replay a drawing session and record details such as how long it took to make a drawing, which tools were used, and how often a user selected the "undo" command.

Figure 4 shows EyeDraw with a drawing created by one of the authors after some practice.

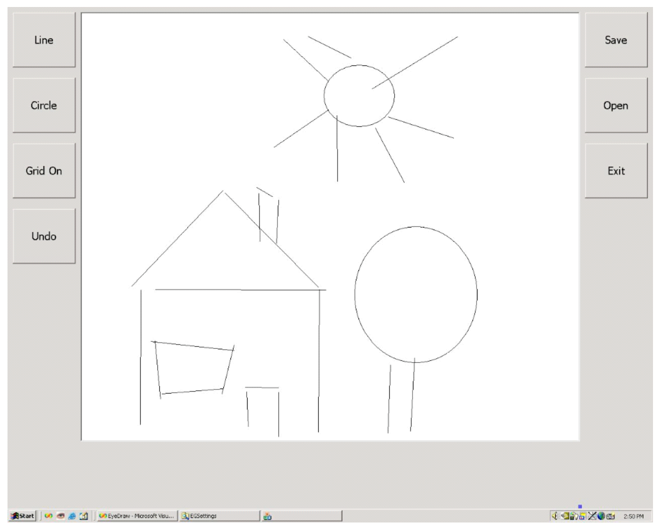

Figure 4. The EyeDraw program with a drawing made by one of the developers.

USER EVALUATION

We are conducting user observation studies of EyeDraw with both nondisabled and disabled users. Nondisabled children and adults who have never used an eye tracker are asked to draw a few specified shapes using their eyes, draw a few pictures, and then discuss their experience. Thus far, three of the four participants quickly and easily drew the requested shapes and recognizable scenes including houses, stick figures, a car, and a butterfly.

Figure 5 shows the butterfly drawn by a 14-year-old nondisabled participant the first time she ever used an eye tracker. Most drawings from the study with nondisabled users would be classified into the fourth stage of drawing, which is *early pictorial*. This is the stage at which the drawn shapes and configurations can be associated with people and things in the world.

One participant, a 7-year-old boy, did not get past the random scribble stage. We believe his performance was analogous to a child learning how to move their hand to control a crayon for the first time. It will be interesting to see how 7-year-old children who are practiced with the eye tracker perform with EyeDraw.

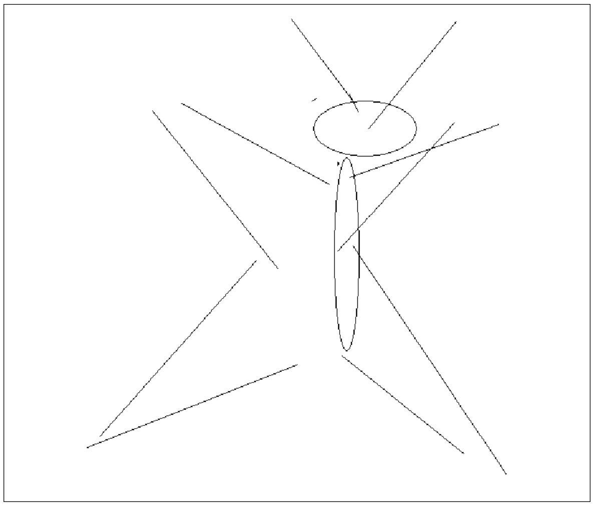

Figure 5. An eye-drawing made by a 14-year-old nondisabled participant the first time she ever used an eye tracker.

Lessons learned thus far from the user observation study include that 500 ms is a good starting point for a cursor activation dwell time, and that the optimal dwell time decreases with practice and familiarity with EyeDraw. The intermediate *Ready to Draw?* yellow cursor stage is helpful for beginners but not as necessary once users become familiar with EyeDraw. Based on watching patterns of use of the "Undo" command, some means of nudging the eye cursor is needed to reproduce the slow adjusting movements typically made by the hands.

As of this writing, we are starting to evaluate EyeDraw with children and adults with severe motor impairments. LC Technologies has put us in touch with a number of these users throughout the United States. We have initiated contact with the users and their caregivers and have sent them the software and associated paperwork for evaluation. The users are very excited at the prospect of being able to draw pictures with their eyes. We are very curious to see how it will go.

CONCLUSION

This paper discusses the design and development of EyeDraw, an interactive software system that will enable children with severe motor impairments to do something that is currently difficult or impossible for them to do, to draw pictures by just moving their eyes. Our initial design remains focused on this one user group and task. We will work directly with these users to improve the usefulness and functionality of the software as needed to better support this important childhood developmental activity.

Even as we maintain our focus, we anticipate that EyeDraw will spawn additional new applications and eye-control techniques. There is a rich history of technology and design modifications intended for disabled users also providing useful new technology to the general public [7]. The eye-control techniques developed in EyeDraw already surpass those used in previous attempts at free-eye drawing.

The project demonstrates how a detailed examination and understanding of human performance characteristics, and a task analysis that incorporates the user's task as well as the eye movement subtasks required for the task, can contribute to the design of new and useful real-time input applications for eye tracking in HCI.

ACKNOWLEDGMENTS

The authors thank Dixon Cleveland, Nancy Cleveland, and Peter Norloff, all from LC Technologies, for technical and domain expertise; Maureen Warman for requirements analysis; and Annie Zeidman-Karpinski for literature review.

The second and third authors are supported by a Research Experiences for Undergraduates grant from the National Science Foundation. The eye tracking lab used for these projects is supported by the Office of Naval Research through Grant N00014-02-10440. Both of these grants are awarded to the University of Oregon with Anthony Hornof as the principal investigator.

REFERENCES

1. Bolt, R. A. (1982). Eyes at the interface. *Proceedings of Human Factors in Computing Systems,* New York: ACM, 360-362.
2. Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
3. Duncum, P. (1995). Colouring-in and alternatives in early childhood. *Australian Journal of Early Childhood*, 20(3), 33-38.
4. Escobedo, T. H., & Bhargava, A. (1991). A study of children's computer-generated graphics. *Journal of Computing in Childhood Education*, 2(4), 3-25.
5. Gips, J., & Olivieri, P. (1996). EagleEyes: An Eye Control System for Persons with Disabilities. The Eleventh International Conference on Technology and Persons with Disabilities.
6. Jacob, R. J. K., & Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises (Section commentary). In J. Hyona, R. Radach, & H. Deubel (Eds.), *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movements*. Oxford: Elsevier Science.
7. Newell, A. F., & Gregor, P. (1997). Human computer interfaces for people with disabilities. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction.* (2nd ed.). Amsterdam: North-Holland, 813-824.
8. Tchalenko, J. (2001). Free-eye drawing. *Point: Art and Design Research Journal*, 11, 36-41.