See-Thru: Towards Minimally Obstructive Eye-Controlled Wheelchair Interfaces

Corten Singer
Björn Hartmann, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2018-61
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-61.html

May 11, 2018

See-Thru: Towards Minimally Obstructive Eye-Controlled Wheelchair Interfaces

by Corten Singer

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Björn Hartmann

Professor Björn Hartmann
Research Advisor

5/10/2018

(Date)

* * * * * * *

Eric Paulos

Professor Eric Paulos
Second Reader

10 MAY 2018

(Date)

See-Thru: Towards Minimally Obstructive Eye-Controlled Wheelchair Interfaces

Corten Singer

Department of Electrical Engineering & Computer Science, University of California, Berkeley

Abstract—**Eye-tracking interfaces are an active area of interest in human-computer interaction (HCI) research because of their potential to increase the communication bandwidth between humans and computers when the use of hands is not possible. For some people living with severe motor disabilities, the use of eyes is the only available input modality to control and interact with the various devices that enable their independence. One particularly enabling gaze-based application that has driven a large body of research is that of wheelchair navigation. The objective of this work is to develop and evaluate an eye-controlled wheelchair interface that improves upon the state of the art by considering, first and foremost, the interaction pattern between the user and the system. We explore removing the use of a computer screen from the navigation system, which normally serves to provide feedback to the user. The underlying goal behind this design decision is to avoid obstructing the user's field of view, which is inherently limited given the nature of their disability. We present a novel eye-tracking interface device that provides feedback to the user without a screen while simultaneously allowing the user to see through it in order to provide a clear view of where they are driving. This prototype has been evaluated against the screen-based state of the art in a preliminary clinical test with three users.**

Index Terms—**Eye Gaze, Eye Tracking, Gaze Control, Eyes-Only Interaction, User Interfaces, Power Wheelchair, Smart Wheelchair, User Experience, Assistive Technology, Gaze Gestures, Field of View, Obstructive**

1. Introduction

Eye-tracking is the process of estimating a user's gaze, or where a user is looking. Eyes and their movements have long been studied for their communicative importance in conveying a person's needs and emotions [1], as well as their strong indication of attention and intent [2]. In particular, researchers have explored gaze-tracking in the context of interaction. One vein of this research has resulted in the design of personal gaze-based devices that allow users with "locked in" neurodegenerative diseases or other severe motor disability to interact with the world with a transformative level of independence and autonomy.

The design requirements for these eye-gaze interfaces are substantially different from traditional human-computer interfaces because both command execution and feedback observation tasks are performed by human eyes simultaneously [3]. As most of these applications are designed to enable users to interact with computers, the user interfaces generally rely on robust screen-based feedback to help navigate between various execution and observation tasks. When used in the context of driving a wheelchair, the design requirements become even more complex. Specifically, this is due to the highly dynamic nature of the application in which the user must simultaneously watch where they are going and engage with their gaze-enabled interface. The additional task of evaluating the path ahead is arguably just as important as controlling the eye-gaze interface itself, yet most conventional eye-controlled wheelchair implementations regard it as an afterthought as they require users to deal with the obstructive computer monitor that is mounted directly in the center of their gaze. On top of all this, the active nature of mobile eye-tracking introduces further challenges that affect both the calibration and accuracy of the gaze detection sensors. This includes accidental head movements caused by bumpy terrain or varying lighting conditions that introduce noise to the eye-tracking sensors.

Despite these design challenges, portable gaze-controlled devices that are capable of maneuvering power wheelchairs hold the promise to greatly increase the independence and quality of life for those living with severe motor disability. As advancements to eye-tracking research and technology works towards creating realistic, consumer-available systems, it is essential that their designs are not only functional, but robust and realistically easy to use.

This paper proposes a novel gaze-based interaction technique for wheelchair control that seeks to address the primary challenge of a screen obstructing the field of view of the wheelchair user. Our technique

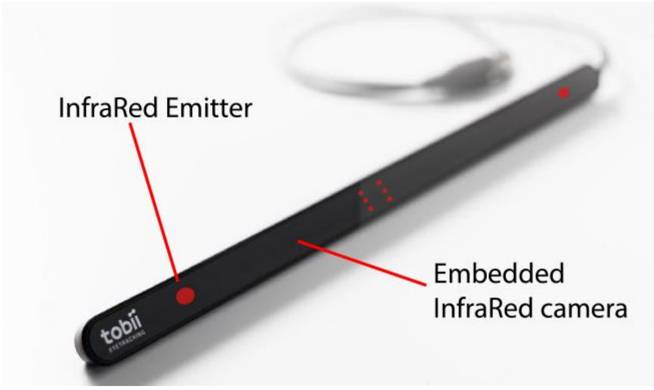

Figure 1. Tobii Eye Tracker 4C sensor before mounting to a computer.

replaces an LCD screen with a small set of spatially arranged LEDs that serve as both gaze targets and status indicators. The space in between LEDs is open, so that users can observe the environment during navigation. Users also have to complete a limited set of initial gaze gestures to invoke wheelchair control states to avoid unintended activations. We present two realizations of this technique: One based on an open frame equipped with a gaze tracker that is mounted to a wheelchair in the same location where a screen would be placed; and one with a head-worn gaze tracker that is augmented with LEDs in the user's visual periphery. We describe the implementation of these systems and report on initial feedback from three users who successfully completed wheelchair navigation tasks with our technique.

2. Related Work

Current eye-tracking devices draw from a varied repertoire of hardware sensing platforms and detection algorithms that extends well beyond the scope of this paper. In an effort to familiarize the reader with the specific technology behind the work proposed in this research, only the most relevant eye-tracking methods will be discussed. This section also details a few examples of current state of the art gaze-controlled wheelchairs that have served as an inspiration to the novel eye-tracking interface presented in this work.

2.1 Eye-Tracking Methods

Within the literature, most eye-tracking methods can be generally divided into two main groups: those measuring the angular eye position relative to the user's head (through a head-mounted sensor), and those measuring eye position relative to the world (through a sensor mounted in the environment) [4]. The first method is especially interesting because it enables gaze-detection in a mobile context by providing more freedom to adjust one's seating position without compromising the sensor accuracy. Unfortunately, such head-mounted systems are expensive due to their small market size, traditionally costing upwards of 12,000 USD, and are therefore not a realistic option for the average "locked-in" user. One exception, however, is that of Pupil Labs head-mounted eye-trackers, which may very well disrupt the market for wearable and affordable eye-trackers with a price point of around 1,500 USD. On the other hand, eye-trackers that measure absolute eye-position relative to the sensor's placement have become increasingly affordable, like the Tobii Eye Tracker 4C used in this research (see Figure 1), which costs only 150 USD. Regardless of the form factor, these sensing platforms operate with similar underlying principles in order to produce their measurements.

Video-oculography (VOG) works by tracking visible features of the eye – such as the pupil, iris, sclera – or reflections on the surface of the eye – corneal reflection [5]. Gaze direction is estimated by processing images recorded by a video camera (or multiple), either in a remote table-top system [4, 6], as shown in Figure 1, or on a head-mounted system [7, 8, 9, 10] as shown in Figure 2. These methods typically use IR light for illuminating the eye, and/or IR cameras for detection because the uncontrolled ambient lighting used as the light source in visible spectrum imaging [11] can add noise and unpredictability to the system.

Head-mounted VOG systems are often more user-friendly than table-top systems and they allow eye-tracking that is unaffected by head position. However, most systems that are interested in gaze point measurements on a user interface measure eye position relative to the surrounding environment [4]. Regardless of which method is employed, the fundamental steps to VOG signal acquisition are the same. For instance, either method performs eye localization, in which eye images are extracted from face images (or partial face images).

2.2 Eye-Tracking User Interfaces

Gaze can be much faster at indicating intent than conventional methods, such as a manual mouse. Gaze not only shows where current visual attention is directed, but it also precedes human action, meaning that we look at

Figure 2. A Pupil Labs head-mounted VOG device [7] with custom PCBs added onto the backs of the standard eye-tracking cameras.

things before acting on them [12]. As such, a key challenge involves designing effective interfaces that can successfully exploit this potential. This section discusses only a few of the most fundamental interface designs found in conventional eye-tracking systems. The following techniques comprise the basis upon which our proposed prototype has been implemented, although it should be noted that there are many others that exist beyond the scope of this work.

A. Gaze Regions

Active gaze regions are one of the simplest techniques to replicate mouse clicking in eye-tracking applications. Gaze regions are software-defined zones on a digital screen that are linked to specific pixel boundaries. When an eye-tracking system detects that a user's gaze has entered into the pixel boundaries of a particular gaze region, a unique action can be triggered automatically.

As mentioned previously, using gaze as an input method can introduce many diverse problems, since the eyes are used for both sensing and control. Most notably, systems need to be able to prevent the "Midas Touch" problem, in which all items viewed are selected [13], without the user intending to do so. There has been considerable research that has implemented and tested various methods to do just that over the years [14]. For systems that incorporate eyes-only interaction, the most common method used to prevent mistaken activations is to use a short delay period, known as a "dwell time", which differentiates observation from intended control commands. It works by displaying feedback to the user for a pre-defined period of inaction immediately after a user's gaze falls on a selectable gaze region. Given this feedback, users have the option to maintain their gaze until the period of inaction ends or to cancel the action by looking elsewhere. This consideration enables a much more usable interaction pattern when dealing with a screen interface with multiple active gaze regions because it allows the user to examine the possible selection options without immediately activating anything. Dwell time can vary depending on the skill level and responsiveness of a user, but generally longer dwell times can be uncomfortable and tiring. Majaranta el al devised a system to navigate this issue by allowing the possibility to adjust dwell time, which increased user satisfaction [15].

B. Scanning Interfaces

Some people may have difficulties in rapidly fixating their gaze due to the limiting nature of their particular disability. This may make it difficult to perform consecutive eye movements or movements in more than one direction, so an alternative method for selection is needed. The scanning method allows the eyes to be used

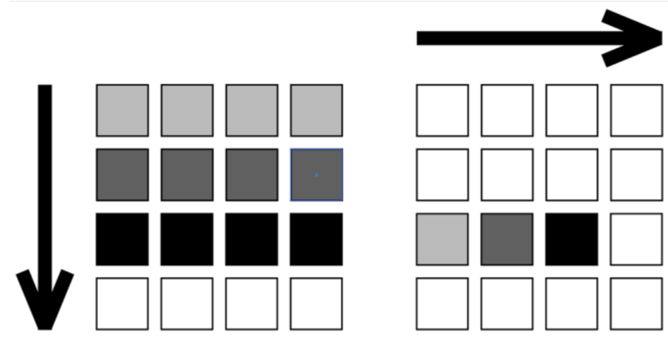

Figure 3. Scanning by groups of items. Redrawn from [18]

as a simple one-way selection switch, while the focus of the interface shifts between items automatically [16, 17]. The focus between objects changes according to a definable timer and the user simply needs to trigger this one-way selection switch (often a simple eye-movement) to select an object of interest when the focus lands on it.

Scanning can be slow if there are multiple options, so it can be helpful to scan groups of items before shifting focus onto single items. A common approach involves scanning items row-by-row until a row is selected, at which point focus is directed onto the individual items in the selected row, as demonstrated in Figure 3.

C. Gaze Gestures

Similar to the nature of using eye movements as triggers is the method of gaze gestures. In gesture based systems, the user initiates commands by making a sequence of "eye strokes" [19]. These gestures can be made with relative eye strokes at areas either on or off the screen, or by making strokes that do not focus on any particular area in the interface. The latter is particularly helpful for systems that no longer need to calibrate sensor measurements to specific pixel locations on a screen, and commands can be issued by making simple, easily-recognizable gestures. One notable example is that of Yamagishi et al who devised a system leveraging the use of eight directional gestures in order to select letters on an on-screen keyboard [21].

Much of the existing research concerns itself with complex gaze gestures, requiring 2 or more eye strokes.

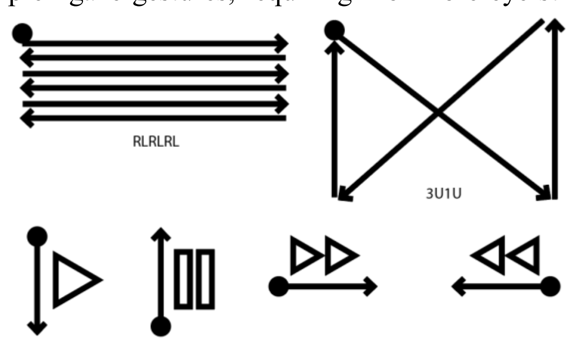

Figure 4. Complex gestures (top) can be fatiguing due to the large amount of eye movements. Redrawn from [20]

Although this has the advantage of increasing a user's gaze-enabled vocabulary without crowding the screen space, it may be difficult to remember a large number of gestures and physiologically it may be difficult to create and complete them [22]. As such, Møllenbach et al developed a system of single gaze gestures that are gestures requiring only one segment, or eye stroke, in one direction [23]. This provides the use of four key gestures for screen usage: Left to Right, Right to Left, Up to Down, Down to Up.

D. Importance of Feedback

A further area of concern lies in sufficient feedback in an eye-tracking interface. In an overview of eye-tracking in advanced interface design in 1995 [24], Jacob noted the important role of feedback in gaze interaction, specifically with the use of a cursor indicating the location of the user's gaze. User's generally know where they are looking, but not with a pixel-by-pixel accuracy. Also, slight calibration errors can give rise to discrepancies between where the user is looking and where the system thinks the user is looking, so visual feedback helps to bridge this gap. Even if an eye movement-based system does not incorporate the use of a cursor or some other method for feedback on the user's focus, feedback is still relevant for selection purposes in indicating whether the system has selected the intended object.

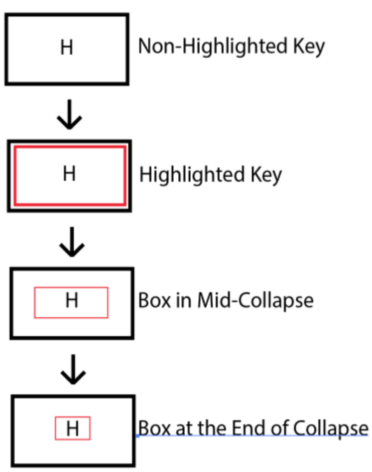

Figure 5. Animated gaze component highlighting. Redrawn from [18]

One example is that of animated feedback of the progression of dwell time, which helps users maintain their gaze on the desired object long enough to avoid premature exits [25]. This method works on top of highlighting gazed-upon objects on a screen, which helps the user verify that the system is actively aware of the location of their gaze [18].

2.3 EYE-CONTROLLED WHEELCHAIRS

Driving a power wheelchair with only eye movements necessitates a robust system capable of quickly detecting user intention while simultaneously providing control over many degrees of freedom. For people living with severe mobility challenges, such as degenerative neurological disorders like amyotrophic lateral sclerosis (ALS), these eye-controlled wheelchairs must incorporate an interface that is unobtrusive, easy to use, responsive, versatile and affordable (among other metrics that contribute to user satisfaction) [5].

Although many interaction techniques we have mentioned so far are sufficient for conventional eye-tracking environments, research shows that wheelchairs encounter situations that are much more dynamic [26]. Specifically, successful systems must design for users that are actively paying attention to both the environment through which they navigate and their eye-tracking interface. This means that these systems must consider the constant movement and changes in both body positioning and lighting conditions that are a natural part of navigating through real-world environments [27].

A. Gaze Regions

A majority of the research involving eye-controlled power wheelchairs concerns itself with the implementation of gaze-based interfaces that incorporate dwell time interactions with on-screen gaze regions that map to wheelchair motor controls. For instance, Barea et al developed one of the first gaze-driven power wheelchairs with a simple graphical user interface (GUI) composed of four gaze enabled buttons that mapped to directional control of the wheelchair's motors [28]. Each button, represented by an arrow pointing either up, down, left, or right, increases the speed of the chair in the direction corresponding to the selected button.

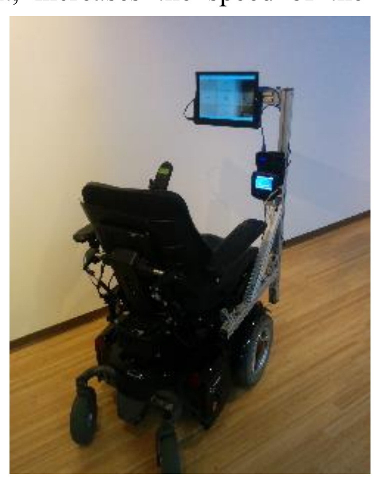

Figure 6. An example of a common eye-controlled wheelchair design made by Microsoft Research.

One obvious flaw with this system is that no matter where a user looks, their eye movements will initiate an action of some sort, which becomes very strenuous and fatiguing for the user. Furthermore, Barea's prototype only allowed for one-way communication through the interface, and no feedback was presented to the user, depriving them of any cues as to how the system interprets their eye-movements. Further still, this implementation requires a screen for user interaction, but this obstructs the user's field of view (FOV), which makes natural navigation even more challenging. As can be observed from the literature, this obstruction is a common design flaw shared amongst nearly all implementations built around a screen-based GUI.

In recognition of the fatigue brought on by activating a motor control command no matter where the user's gaze lands on the interface, Lin et al proposed a novel GUI that divided the screen into 9 regions, 4 of which are gaze contingent command regions while the remaining 5 regions are idle zones that allow the user to rest their gaze without any activation (see Figure 7) [29]. Just like the previous implementation, however, this system presents no visual feedback to the user whatsoever.

Once a user activates a command by gazing at one of the 4 active regions, the central idle space will expand and absorb the recently activated command region, effectively providing feedback as to which command is being executed by temporarily rendering it an idle zone. This GUI design allows the user to rest while still carrying out a navigation action, which provides much more natural interaction for the user. There is still, however, the inherent issue with the visual obstruction of the screen display blocking the user's FOV during navigation.

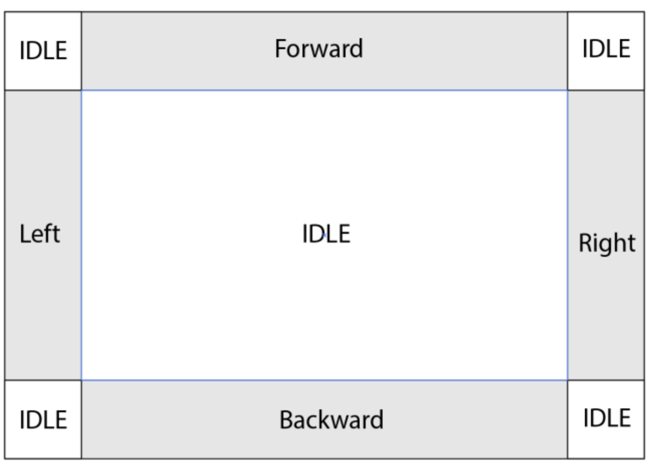

Figure 7. A gaze interface with idle gaze regions developed by Lin et al. Redrawn from [29].

Wästlund et al devised an interface that aimed to both provide adequate feedback to the user and reduce the inherent obstruction of the screen in use [30]. Similar to Barea's work, their solution consists of a gaze-contingent dynamic interface that can visually change its layout in response to gaze commands issued by the user. Unlike previous prototypes, however, this interface is overlaid on top of a live webcam stream of the view immediately in front of the screen so that the user can watch the screen to both interact with the GUI and to keep an eye on where they are going (see Figure 8). The GUI consists of four directional control buttons, with active highlighting of the currently selected button to give feedback to the user that indicates which command is currently activated.

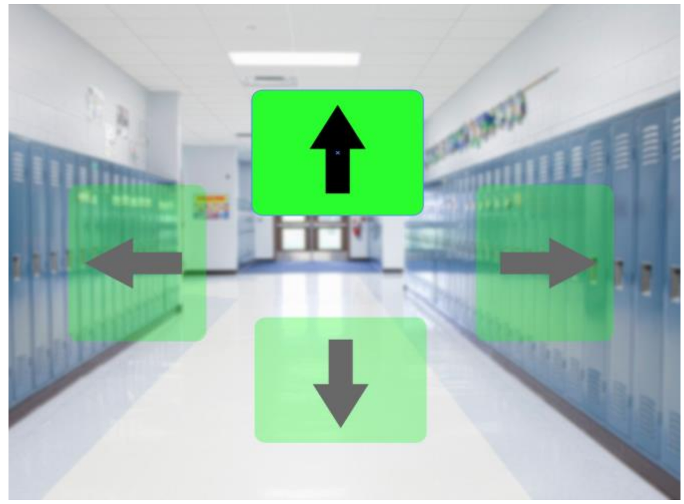

Figure 8. Live video stream of the environment in front of the power wheelchair with gaze components overlaid. Redesign from [30].

Virtually all other state of the art gaze-driven wheelchairs involve a similar GUI design as this one proposed by Wästlund et al, whereby a live video stream is used to work around the inherent obstruction posed by the use of a computer screen. Although this video stream is a clever way to allow the user to see behind the screen, it can feel disorienting and unnatural because it does not fit neatly into the user's understanding of the scene around them. On top of their own perception of the environment through their eyes, the user must mentally process the wide-angled view of the path ahead through the digital screen. This introduces a large margin of visual overlap that is confusing to reconcile.

B. Scanning Interfaces

This is perhaps the least explored interface design in the literature related to eye movement-based control of power wheelchairs, and for understandable reasons. In practice, the interaction pattern of conventional scan-based eye-controlled wheelchairs is unnatural and awkward because users are forced to approximate the distance of each desired movement within a small error margin rather than simply thinking about the general location they wish to travel towards. In other words, the user is given the extra task of breaking down each navigational intent into a sequence of distance values, which shifts the focus from thinking about *where to go* to the less intuitive question of *how far to go*.

One notable design involving the use of a scanning interface provides the option to drive the wheelchair forward a variable distance – e.g. 2ft, 5ft, 10ft, etc. – using rotary encoders to measure how far the chair has travelled. On top of the awkward interaction experience inherent to the scanning method, these implementations suffer from a considerable disconnect between the amount of distance that a user truly desires to travel and the amount of distance that is actually selected through the interface. The issue is that the average human does not have an accurate enough mental representation of discrete distances without any way to physically measure them. This becomes problematic when a user wishes to travel to a specific point ahead of them and they must guess exactly how far away that point is.

C. Gaze Gestures

As with most gesture based eye-tracking systems, delivering appropriate feedback to the user is a great challenge. Without the explicit need for on-screen targets to interact with, however, some researchers have removed the use of a screen altogether in their gesture based implementations of wheelchair control.

For instance, Purwanto et al created a gesture based system using only an off-the-shelf webcam mounted in front of the user on the wheelchair [31]. In this implementation, gestures that convey an explicit gaze direction (such as holding a gaze to the left) correlate to a driving action in the corresponding direction. Stop actions occur when the user does not focus their gaze in any particular direction. This implementation has the advantage of an extremely low cost and easy set up. However, the lack of feedback makes it difficult to use.

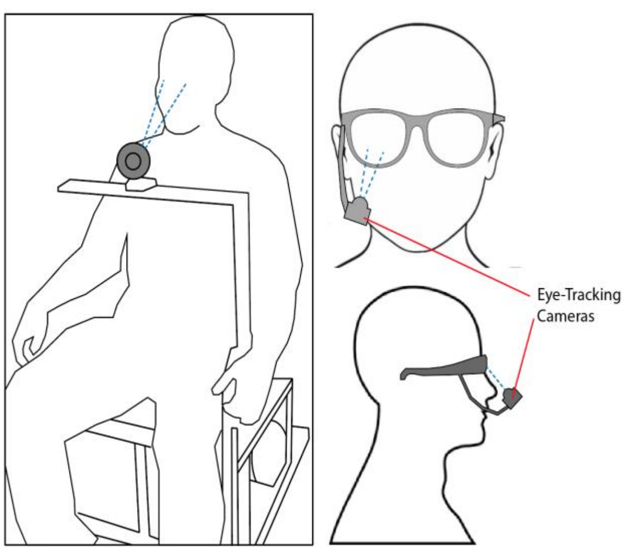

Figure 9. *Left*: Gesture based wheelchair control using only a webcam; *Right*: Head-pose free alternative for the previous gesture based implementation. Redrawn from [31] and [32], respectively.

Plus, the need to constantly hold a gaze in the desired direction of navigation prevents the user from experiencing the full benefits of an unobstructed, screen-free FOV. Other related work propose a similar gesture based implementation using head-mounted cameras for head-pose free interaction [32, 33].

With only these few exceptions, most previous eye-controlled wheelchair prototypes have essentially adapted conventional desktop eye-tracking applications to a portable computer mounted to a power wheelchair. This overlooks the fact that a digital screen (which is very useful in the desktop context) can actually encumber the user during navigation tasks. Nevertheless, if a screen-free implementation is not properly designed, the resulting interaction pattern can be even more difficult to operate. Although user testing has demonstrated the functionality of these various prototypes, our novel interface explores a "middle ground" that incorporates the advantages of both extremes, resulting in a more natural interaction.

3. TECHNICAL DESCRIPTION

A functional prototype of an eye-controlled wheelchair has been developed with a novel control interface that is designed to be as minimally obstructive as possible. The following aspects were realized in order to satisfy the needs of the target population:

- The proposed system can be used with a standard, commercially-available power wheelchair to avoid the costs of buying a new one
- The user can drive the power wheelchair by simply gazing at the control interface
- The user can see through the control interface to have a better view of the path ahead
- The user should have the option to rest and gaze freely without activating any drive commands once navigation is complete
- The chair can easily switch between the gaze-controlled driving mode and a standard driving mode

3.1 DESIGN DECISIONS

The proposed interface device is designed to mount onto 1/4"-20 UNC screws, which is the standard screw size for camera tripods. This decision was made so that our interface could readily mount onto most consumer-available articulating arms, which can be easily attached to the side of most standard power wheelchairs.

Additionally, the interface is designed in the shape of a frame with a large space in the middle so that users can see directly through it. Not only does the empty space afford a minimally obstructed FOV, but it also reduces overall material weight, which makes it more stable when mounted on the articulating arm.

Furthermore, our system is designed with multiple different control states so that users can "clutch" in and out of a driving state at will. We recognize that the intended users of our system spend most of their day seated in their wheelchair, and only a portion of that time is spent navigating. Thus, it is crucial that a successful system gives its users the ability to turn on and off the navigation interface whenever they so in order to prevent unintentional driving behaviors in inappropriate settings.

Lastly, our system was developed using standard RNET software, which many wheelchair manufacturers use to control their products. In using this standard software, our prototype is completely compatible with the control system of the Permobil C400 power wheelchair used in this research. This means that the wheelchair can easily switch between the gaze-controlled mode we have developed and the default joystick control mode.

3.2 System Components

The system developed for this research contains the following components: a Tobii Eye Tracker 4C, a Microsoft Surface Pro tablet, an Arduino Micro MCU, a Permobil C400 power wheelchair with an OMNI device, a Manfrotto 143 Magic Arm, an Adafruit Feather M0 MCU, a servo-mounted laser pointer, and a novel "wire-frame" interface device. The system is built around a finite state machine (FSM) that contains three main states: Neutral State, Direct Drive State, and Scan Drive State.

The Tobii Eye Tracker 4C is used to sense the user's eye movements, which are processed by a C# program running on the Microsoft Surface Pro. When the user's gaze fixates on various gaze-activated regions defined by the C# code, the Surface Pro sends serial commands to both the Arduino Micro and Adafruit Feather M0. The purpose of the Arduino Micro is to transform these serial commands into drive commands which are sent to the OMNI device on the power wheelchair. When given appropriate input signals, the OMNI device is capable of controlling the motors of the wheelchair. The use of an OMNI device not only exposes control of the power wheelchair's motors, but it allows the system to have multiple programmable profiles, which affords the ability to easily switch between gaze-based control and standard joystick control. The purpose of the Adafruit Feather M0 is to control the RGB LED feedback actuators embedded in "wire-frame" interface device so that the user can be informed as to how the system perceives their various eye movements. This microcontroller is also responsible for controlling the laser pointer during the Scan Drive State. Lastly, the Manfrotto 143 Magic Arm serves the purpose of mounting the interface device onto any standard wheelchair.

A. Wire-Frame Interface Device

The wire-frame interface comprises 8 directional RGB feedback LED actuators that are mapped onto 8 corresponding gaze-active regions in the C# application.

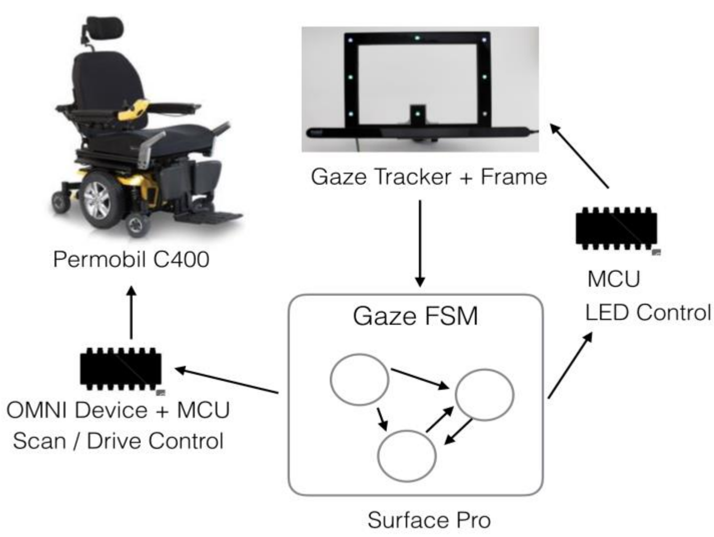

Figure 10. System diagram of the See-Thru prototype

Figure 11. The Permobil C400 power wheelchair used in this study.

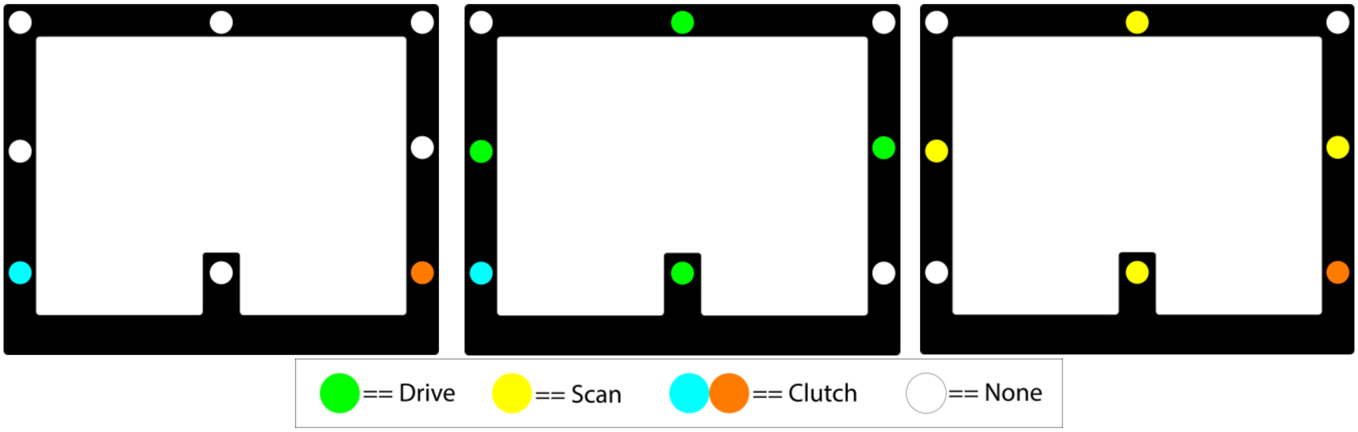

Figure 13. *Left to Right*: The three main control states of the proposed interface – Neutral State, Direct Drive State, Scan Drive State.

These LED actuators are embedded within an acrylic frame with an empty space in the center that can be seen through by the user. The acrylic frame is designed to be about the size of the screen that the Tobii Eye Tracker 4C is calibrated to. Each LED is programmed to change color depending on the state of the system. For instance, in the Direct Drive State, a green LED indicates that the wheelchair will move in the respective direction if the user fixates their gaze on it. In the Scan Drive State, a yellow LED indicates that the wheelchair system will begin scanning in the respective direction if the user fixates their gaze on it. In both of these states, a red LED indicates that the wheelchair will stop whatever action it is carrying out and return to the state's default configuration. In all states of the system, a cyan LED represents a single part of the clutching sequence into (or out of) the Direct Drive State, while an orange LED represents part of the clutching sequence related to the Scan Drive State.

Further, each LED is capable of *active highlighting*, whereby they will increase their current brightness if a user's gaze lands on them in order to communicate that the system is aware of the location of

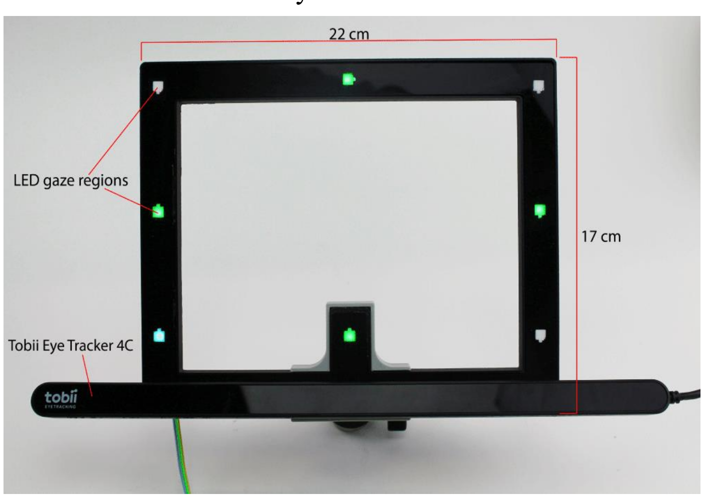

Figure 12. The wire-frame interface device in the Direct Drive State with a Tobii Eye Tracker 4C mounted on it.

their gaze. Lastly, each gaze-enabled LED region can only be activated after a defined *dwell period*, during which the user must constantly maintain their gaze on the desired region. If the user's gaze is maintained for the entire duration of the *dwell period*, then the system will trigger the appropriate response associated with the respective gaze-enabled region.

The interface contains a magnetic mounting strip so that the Tobii Eye Tracker 4C can be directly mounted onto it as opposed to the eye-tracking enabled tablet computer, which is the conventional setup when using Tobii desktop eye-trackers. As the wire-frame interface is not capable of any processing whatsoever, the placement of the LED actuators relative to the magnetic mounting strip must precisely match the location of the software defined gaze-active regions in the C# application running on the Surface Pro. In this way, the wire-frame interface works by tricking the Tobii sensor into thinking that it is actually mounted onto the Surface pro and that the screen-based application is running like it is traditionally supposed to. In actuality, the wire-frame serves as a "stand in" for the computer screen, and the LED feedback actuators serve as physical representations of the digitally defined gaze-enabled buttons (see Figure 12). Under the hood, the Tobii sensor and C# application are unable to tell whether the user is fixating on the gaze-enabled buttons on the screen or the corresponding LEDs in the wire-frame interface.

In order for all of this to function properly, the program on the Surface Pro must still run as usual even though the screen is stowed away to avoid obstructing the user's FOV. Furthermore, the user first has to calibrate the sensor with the screen because the calibration process is not publicly available and requires a complex procedure involving the digital LCD display.

B. Finite State Machine

Each state in the system's FSM serves a unique function, and the system exposes "clutching" controls to

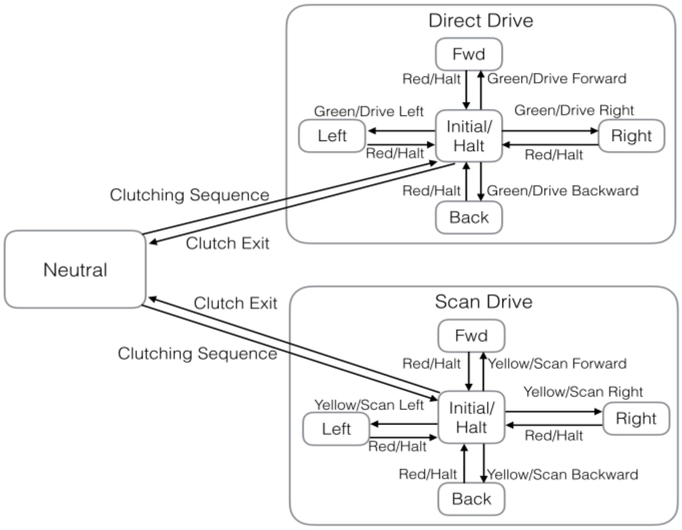

Figure 14. Diagram of the See-Thru finite state machine.

allow the user to switch between them at will (see Figure 13 and Figure 14).

The Neutral State is the simplest state in the FSM and it serves as a "resting" state where the user's eye movements are not capable of triggering any drive commands. This is a crucial safety measure for a realistic eye-controlled wheelchair system because it affords the user the option to attend to the world beyond the eye-tracking interface without having to deliberately avoid looking at it. This way, the user's gaze may unintentionally land on the interface and nothing will happen. Thus, in this state, the user is free to live their life beyond the realm of navigation without any unplanned side-effects. The Neutral State consists of only 2 gaze-enabled LED regions positioned at the bottom-left and bottom-right. These 2 LEDs serve as the initial "clutching" controls that are used to step into the Direct Drive State and the Scan Drive State, respectively. The remaining 6 gaze-enabled LED regions are turned off in order to convey their inactive status.

The Direct Drive State comprises 4 main directional drive regions and a single clutch LED. The drive regions adhere to the following convention: The Top-Middle gaze region maps to Forward Drive, Bottom-Middle maps to Backward Drive, Left-Middle maps to Left Drive and Right-Middle maps to Right Drive. The state clutch LED is activated by gazing at the Bottom-Left gaze region. The directional drive LEDs are initially green to indicate that gaze-based activation will result in actuation of the power wheelchair's motors in the respective direction. The clutch LED is cyan in order to be easily distinguished from the green drive regions. This clutch region allows the user to switch back into the Neutral State when the navigation task is complete. When a gaze-enabled directional drive command region is activated, the wheelchair will begin to move. Also, the interface will change its layout so that the 7 remaining LEDs become red, indicating that the wheelchair will be halted if they are activated next. When the user halts the wheelchair after issuing a drive command, the interface returns to the initial Direct Drive State, exposing the original 4 green drive regions and the single cyan clutch region.

The Scan Drive State is similar in design to the Direct Drive State, except that it follows a different color-coded scheme and the Up/ Down control regions are now mapped to the actuation of a scanning laser pointer instead of the wheelchair. Rather than green, the main directional command regions are yellow to help convey to the user that they are in a different state. Additionally, the clutch LED is orange instead of cyan, so that the user can easily determine which state they wish to step into. The red color-code, however, is preserved and its role is still to stop the scanning functionality of the wheelchair. When a user activates either the Up or Down gaze-enabled command region in the Scan Drive State, a servo-controlled laser pointer mounted to the side of the wheelchair will begin to scan forwards or backwards, respectively. The idea is to physically ground the conventional eye-tracking scanning interface onto the environment immediately ahead of the user so that they can choose where to go simply by pointing the laser there. This avoids requiring users to guess the exact distance that they wish to travel.

C. Clutching Mechanism

The system's clutching mechanism consists of a feedforward sequence that begins when the user fixates their gaze on the initial color-coded gaze-enabled LED (see Figure 15). After the initial clutch region is activated, all of the gaze-enabled LEDs in the interface turn off except for the next LED in the sequence. The sequence

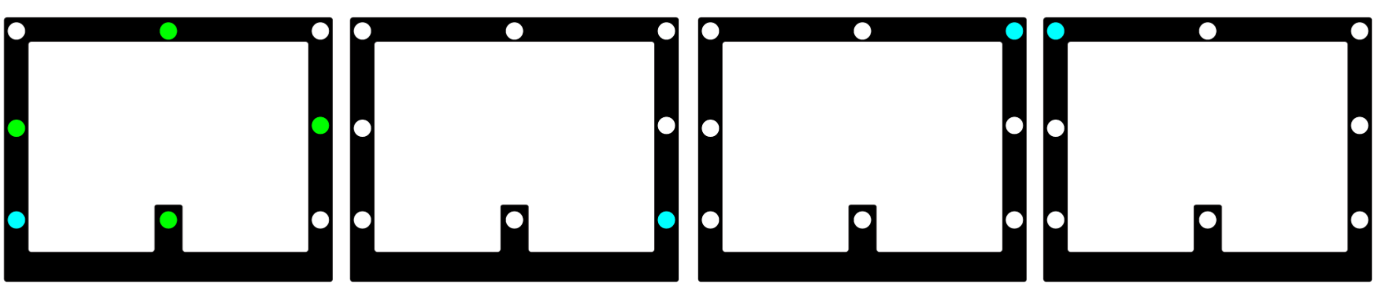

Figure 15. A full clutching sequence starting from the Neutral State. After activating the last gaze-enabled clutch region, the system will end up in the Direct Drive State

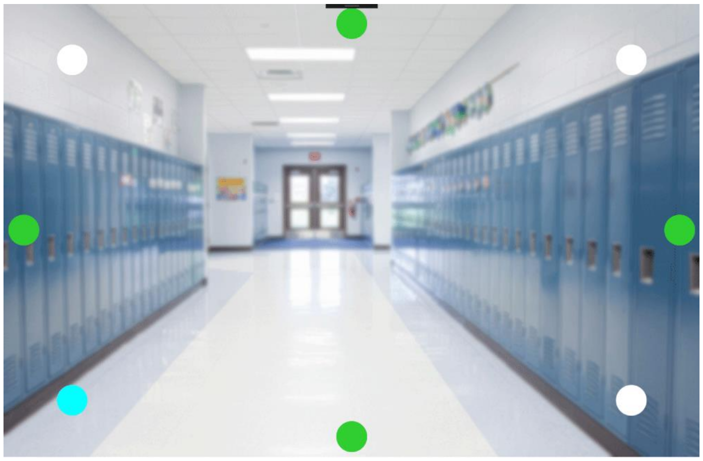

Figure 16. The screen-based interface used as a control in the experimental study incorporates a live video stream of the environment behind the screen. The gaze-enabled command regions are overlaid on top of this video stream.

consists of 4 consecutive LED activations placed strategically around the interface so that only clearly intentional eye-movements are capable of switching states. If the users gaze fixates on an inactive LED at any point during this clutching sequence, the system will return to the most recent state before the clutching began. This is particularly useful in the Neutral State, whereby accidental clutching activations caused by natural eye-movements do not invoke any unwanted drive commands.

4. Experiments

Through user testing and evaluation, our first goal was to verify that our proposed prototype is in fact usable by the target population. Second, we were interested if the wire-frame mechanism does indeed provide better visibility than an LCD screen-based control scheme. In sum, we set out to determine whether our eye-tracking interface was not only functional, but preferable to use when compared to the state-of-the-art screen-based alternative. We evaluated our wire-frame interface against a screen-based control that was built around a similar gaze-tracking interaction scheme.

The prototype was tested with 3 individuals, each with severe motor disabilities that necessitate total dependence on a personal power wheelchair in order to navigate. It should be noted that only one participant, who we will refer to as O, controlled their wheelchair via a hands-free interface, while the remaining participants navigated their wheelchairs using standard joystick modules.

The study consisted of two main experiments, the first using the interface proposed in this paper and the second using a screen-based control. The screen-based control is designed around the same gaze-control scheme as the wire-frame interface, but in place of the frame, we mount an LCD screen that displays a live video stream beneath the color-coded gaze-regions (see Figure 16). Each experiment consisted of a series of 5 eyes-only navigation tasks, each one increasing in difficulty (see Figure 17). Each task required that the user start in the Neutral State and then clutch into the appropriate drive state once the task began. Before beginning the experiments, each participant was trained to use the interface, which consisted of a walk-through of the system's FSM along with a practice drive lasting roughly 5 minutes.

Task 1 had the participants use the Direct Drive State to drive the power wheelchair forward in a straight line for 30ft, then backwards for 7ft, all without driving outside demarcated boundaries that were 6ft wide made of blue tape.

Task 2 had the participants use the Direct Drive State to navigate the wheelchair along a curved path approximately 50ft in length, followed by a 180 degree turn, all without driving out of bounds. The curved path was designed to represent simple obstacle avoidance.

Task 3 had the participants use the Direct Drive State to navigate in a "figure 8" pattern around 2 obstacles placed 7ft apart in order to test fine steering control. This task had no boundaries.

Task 4 had the participants use the Direct Drive State to drive the wheelchair forward in a straight line for

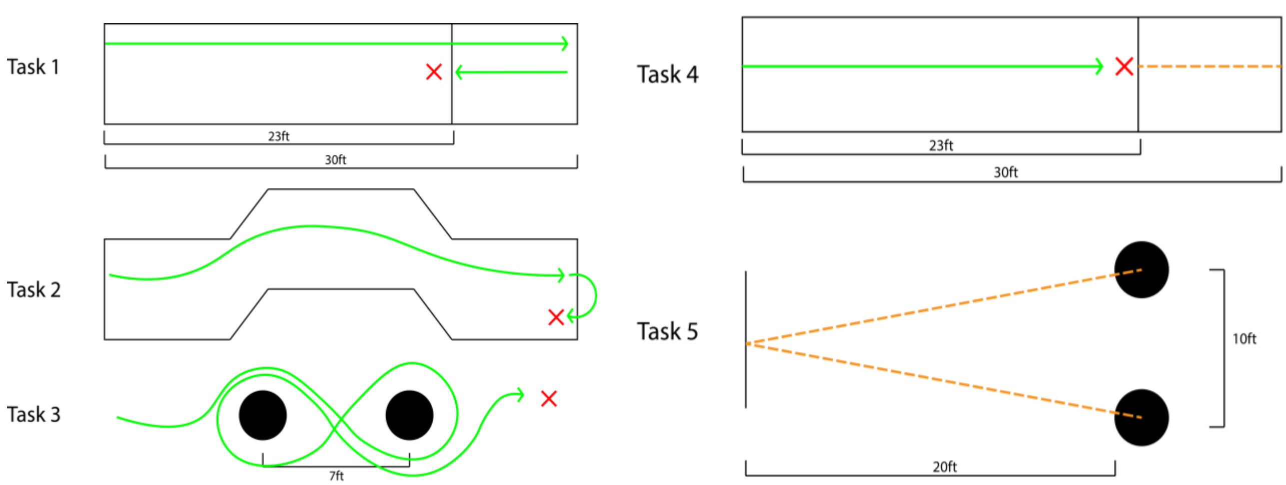

Figure 17. A visual diagram of the 5 navigation tasks carried out in each experiment.

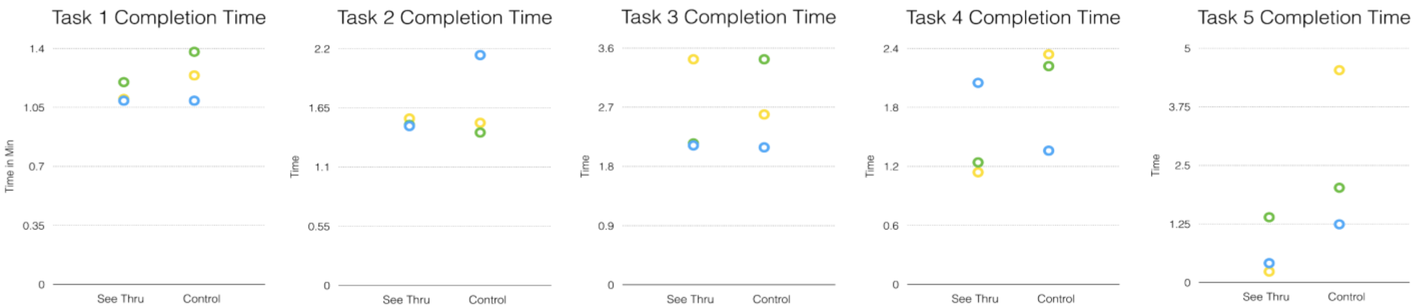

Figure 18. Visualization of the time taken by each participant to complete all five navigation tasks. Each graph compares the time required for task completion when using the wire frame interface compared to the screen-based control.

23ft. Participants had to then switch into the Scan Drive State in order to point the scanning laser as close as possible to a horizontal strip of tape placed 7ft ahead of them.

Task 5 had the participants use the Scan Drive State to place the scanning laser onto a physical target 20ft away, and then onto another target 10ft to the left of that.

Each task was timed from start to finish (see Figure 18), and a note was made of any instance in which the participant drove out of bounds. Also, we asked each participant to rate the perceived difficulty of each task on a Likert Scale of 1 through 10, 1 representing the easiest level of difficulty and 10 representing the most challenging (see Figure 19). Lastly, we made note of any comments made by the participants regarding their experience using the eye-controlled wheelchair interfaces.

5. Results

Each participant was able to successfully complete every navigation task using both interfaces. The success found in the driving tasks verify that our proposed wire frame interface can not only be used to safely control a power wheelchair, but can be quickly learned by user's who have no prior experience with eye-tracking systems. Furthermore, it is evident that the novel wire frame interface has distinct advantages over the conventional LCD screen-based interface. These advantages can be seen in the objective data points that were collected during each task, as well as in the subjective feedback given to the researchers by the participants.

As shown in the graphs of Figure 18, the participants generally completed the navigation tasks with a shorter completion time when using the wire frame interface compared to the screen-based control. Perhaps the most promising results were found in Task 5, in which the users were tasked with the challenge of landing the scanning laser on 2 distant targets placed directly ahead of them. The participants clearly had quicker completion times when using the wire frame interface in a context that specifically required an unobstructed FOV.

As can be seen in Figure 19, the wire frame interface was generally easier to use than the screen-based control across all tasks, with the exception of Task 3. Both interfaces, however, were considered relatively easy to use. In fact, the only apparent error observed throughout all the experiments was with one participant who struggled to complete Task 3 using the screen-based system. Specifically, this participant collided with the obstacles that were to be driven around on two separate occasions. This indicates that although such eye-tracking systems are reasonably easy to drive by first-time users, there is still a learning curve that must be overcome before safely navigating in real-world situations.

Figure 19. Visualization of the Likert Scale difficulty ratings associated with each task during both experiments. Participants were asked to assign a difficulty value from 1 to 10 to each navigation task, where one is the easiest and 10 is the most difficult.

On top of these data gathered during the experimental sessions, we received feedback from the participants that helped to shed light on their experience while interacting with our gaze-controlled wheelchair prototype.

5.1 Participant Feedback

The biggest pain point that was brought up by the participants was a problem with the eye-controlled wheelchair system that was independent of the eye-tracking interface itself. The issue, which was expressed by virtually all participants, was the fact that the wheelchair did not drive perfectly straight when the Forward Drive command was activated. This was due to hardware issues with the wheelchair's motors in which the left wheel's motor was weaker than the right, resulting in a slight left angle when the chair received a signal to drive straight. As a result, participants mentioned how they had to constantly switch between activating the Forward Drive command and the Right Drive command in order to offset the motor issue.

The rest of the feedback received from the participants concerned the user experience while interacting with both the LCD screen-based control and the wire frame interfaces.

After the participants had a chance to try both interfaces, they were able to compare and express the key differences that shaped the different interaction experiences. For example, two participants explicitly expressed that they preferred the novel wire frame interface because it was "easier to get a sense of where [they are]" and that they liked that they "can see through the frame this time". The third participant made a comment regarding the screen-based control to a similar effect. They mentioned how the live video stream actually made the task of navigation "harder because of the mismatched perspective" between the video and their own FOV. This participant went on to mention how they felt "super focused on the screen rather than their surroundings" simply because there was a video feed to attract their attention.

Although there were clear advantages to the wire frame interface, there was one clear flaw that was not present in the screen-based control. The wire frame eye-tracking accuracy was slightly worse due to the fact that the eye-tracking sensor needed to first be calibrated on the screen and then transferred to the wire frame interface. Although slight, this change in the accuracy of the sensor was recognized by every participant, resulting in a suboptimal interaction pattern. One participant mentioned how they were unsure if the sensor was malfunctioning when used on the wire frame interface, but they were able to figure out a way to work around this sensor noise by looking slightly to the side of the normal gaze region to trigger the desired drive commands.

Aside from this obvious issue with calibration, most other comments by the participants reflected issues present in both interface designs. Most importantly, the participants mentioned how the dwell time that was programmed into the activation of each gaze region introduced a latency that was at times "frustrating" and "unresponsive". This meant that users had to preemptively fix their gaze on the desired gaze region in order to trigger the corresponding command by the time they actually desired.

6. Discussion

Overall, our new wire frame interface was well-liked and easy to learn by all participants. Although a lot of the data gathered generally indicates that the participants found it easy to use the LCD screen-based control and even easier to use our prototype device, there were still a handful of challenges that were brought to the attention of the researchers.

Most notable was the aforementioned problem of sensor calibration for the wire frame interface. Due to the fact that we do not have access to the proprietary Tobii Eye Tracker calibration procedure, participants had to calibrate the sensor while mounted onto the screen-based control and then switch out the screen for the frame device. Since the wire frame interface had no individual calibration routine, there was a noticeable effect on the accuracy of the eye-tracker when compared to its use on the screen, which required users to slightly adjust their gaze in order to compensate. For one participant, the process of swapping out the screen for the wire frame interface introduced so much sensor noise that the calibration process had to be repeated. Having access to this calibration process would undoubtedly improve the user interaction.

Another important issue with the eye-controlled wheelchair interfaces was that of the latency introduced from having dwell times programmed into each gaze region. As mentioned above in the section on Related Work, interfaces with multiple gaze regions need to be programmed with dwell periods in order for proper operation, but it interesting to see how this can be problematic for a dynamic application such as wheelchair navigation. In particular, precise timing is crucial for adequate control of this type of application so it would be interesting to explore ways in which a novel interaction design could work around this pain point.

Although it was clear in the experimental sessions that the participants did not like that the wheelchair itself was not able to drive perfectly straight, their successful completion of each navigation task shows the robust nature of the interface design when confronted with realistic issues that could very well arise in real life situations. Nonetheless, this angled-drive issue was such a prominent problem due to the fact that the wheelchair's motor controller was driven by a digital switch interface, rather than an analog one. This means that the wheelchair system had binary control over directional movement, whereby the wheelchair could either move in a single direction at one speed or it could not move at all. It would be interesting to explore the analog switch interface of the motor controller, which would allow continuous signals to drive the motors. This means that users would have complete control over the wheelchair, with the ability to switch between varying levels of speed and direction. With this control mechanism, users could easily counteract the angled-drive problem by simply choosing a direction that offsets the angle in order to get straight movement of the wheelchair.

Another challenge of the proposed interface was with the limited control of the scanning laser in the Scan Drive State. The laser was only able to scan forward and backward as there was only one servo motor driving its movement. It would be worth exploring how a laser scanning system could benefit from extra degrees of freedom so that users do not have to first rotate the wheelchair in the direction they wish to scan.

It should be noted that this research is specific to the sole task of navigation, but it is extremely important to recognize that having access to a computer is essential for the severely disabled. In virtually every case, access to a computing platform allows those living with severe motor disability to communicate and interact with the world around them in ways that our minimally obstructive interface cannot. As such, we recognize that it would not be practical to simply get rid of the user's eye-tracking enabled computer and it would thus be an important task to design an integrated system that incorporates aspects of our minimally obstructive navigation interface with more conventional practices concerning the use of computers.

7. Conclusion

Visual feedback in all of the implementations of eye-controlled power wheelchair that we have observed in the literature is "all or nothing". In other words, there is either an entire screen used to deliver feedback or there is no visual feedback whatsoever. Most conventional implementations rely on a digital display screen in order to provide gaze-enabled control buttons and visual feedback to the user. This screen poses a problem, however, as it directly obstructs the user's FOV, which is particularly problematic if the user suffers from a restrictive mobility impairment that may limit their ability to move and see beyond this screen. On the other hand, we saw that gesture based systems can remove the use of a screen altogether, but this comes at the cost of removing visual feedback which is essential for safe and reliable navigation.

The research presented in this paper has set out to explore *intermediate* feedback methods in which an array of LEDs is use to deliver pertinent information relevant to the state of the system as it responds to given eye-based input. This serves to assist the user in their interaction with the wheelchair, while simultaneously avoiding the need for a bulky screen, which is basically an over-engineered feedback platform that obstructs the user's FOV.

Along with this "intermediate" feedback, this paper explores a novel approach to the scanning method in eye-controlled wheelchair implementations. The proposed interface explores a system design that grounds this scanning method in the physical environment that lies immediately in front of the user. This serves to avoid the disconnect that arises from requiring the user to guess the distance they wish to travel. Specifically, this system leverages the use of a laser pointer mounted on a controllable servo motor that is capable of directly pointing to a location in space that can be travelled to.

In the future, we wish to evaluate another minimally obstructive eye-tracking interface that will offer even more flexibility and ease of use. We have already developed the hardware behind a screen-free head-mounted prototype that is analogous to the novel prototype presented in this paper. Rather than using a computer-mounted eye-tracker that requires users to maintain the same seating position throughout its use,

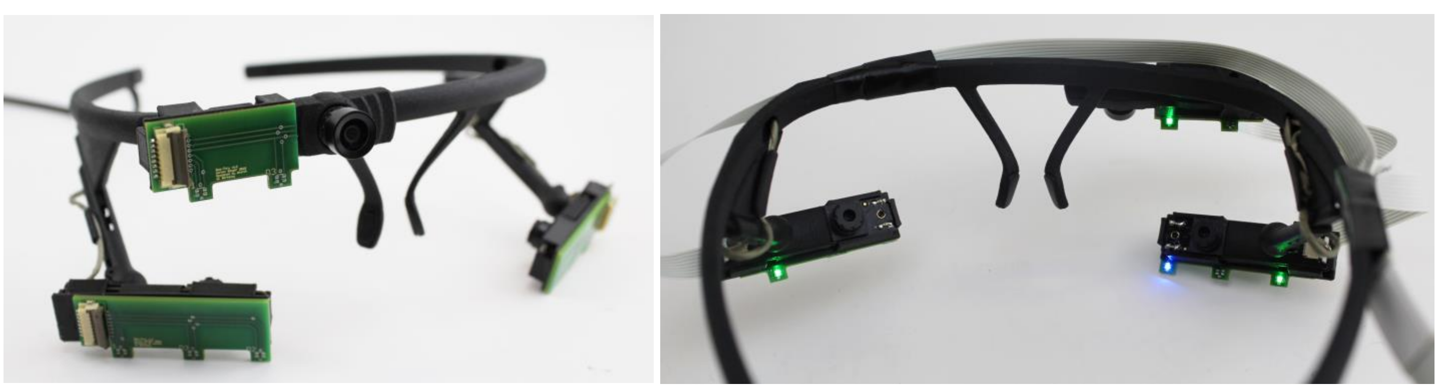

Figure 20. *Left*: Front view of the custom printed circuit boards mounted behind the pupil-detection cameras on the Pupil Labs eye-tracker. *Right*: POV perspective of the interface prototype with the feedback LEDs sticking out into the user's visual field.

such as the Tobii Eye Tracker 4C, our next prototype is based on the wearable Pupil Labs VOG eye tracker, which measures pupil position relative to head position. This means that even if the user changes position slightly as they navigate in their power wheelchair, the calibration and accuracy of the eye-tracking interface is unaffected. As can be seen in Figure 20, the feedback LEDs of this new prototype are even less obstructive than before and they are perceived by the user entirely in their peripheral vision. Thus, the user's FOV is almost completely free from any obstruction, allowing for natural and unencumbered navigation.

REFERENCES

[1] Kleinke, C.L. (1986) Gaze and Eye Contact: A Research Review. Psychol. Bull. 100, 78—100

[2] Zhai, S. What's in the Eyes for Attentive Input. Communications of the ACM, 46, 3, (2003), 34-39.

[3] Z. Ahmed and A. Shahzad. Mobile robot navigation using gaze contingent dynamic interface. In Blekinge Inst. of Technol., Masters Thesis in Comput. Sci., Ronneby, Sweden, 2010.

[4] H. singh, and J. singh, "Human Eye Tracking and Related Issues: A Review," International Journal of Scienific and Research Publicaions" vol. 2, no. 9, pp. 1-9, 2012.

[5] Kristian Lukander, " Mobile usability - Measuring gaze point on handheld devices" Master's thesis, 2003. www.ndltd.org

[6] N H Cuong, and H T Hoang, "Eye Gaze Detection with a Single WebCAM Based on Geometry Features Extraction", 2010 11 International Conference on Control, Automation, Robotics and vision, Singapore, 7-10 December, 2010, pp 2507-2512.

[7] Kassner, M., Patera, W., and Bulling, A. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Procs of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 1151– 1160. ACM.

[8] D. Li, D. Winfield, D.J. Parkhurst, "Starburst: A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature-Based and Model-Based Approaches", *Proc. Vision for Human-Computer Interaction Workshop IEEE Computer Vision and Pattern Recognition Conf.*, 2005.

[9] D. Li, J. Babcock, and D.J. Parkhurst. openEyes: a Low-Cost Head-Mounted Eye-Tracking Solution. In 2006 Symposium on Eye Tracking Research & Applications, page 100. ACM, 2006

[10] J. Breuninger, C. Lange and K. Bengler. Implementing gaze control for peripheral devices. *Proc. of the 1st International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, PETMEI. ACM, 2011.

[11] Hansen, D., and Pece, A. 2005. Eye tracking in the wild. *Computer Vision and Image Understanding 98*, 1, 155–181.

[12] P. Majaranta and A. Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in Physiological Computing*, pages 39–65. Springer, 2014. 1, 3

[13] Jacob, R. J. K.: The Use of Eye Movement in Human-Computer Interaction Techniques: What you look at is what you get. ACM Transactions of Information Systems, Volume 9, Issue 2, p. 152-169, April 1991

[14] G. Boening, K. Bartl, T. Dera, S. Bardins, E. Schneider, T. Brandt, "Mobile Eye Tracking as a Basis for Real-Time Control of a Gaze Driven Head-Mounted Video Camera", *Proc. 2006 Symp. Eye Tracking Research and Applications*, pp. 56, 2006..Velichkovsky B, Sprenger A, Unema P (1997) Towards gaze-mediated interaction: collecting solutions of the "Midas touch problem". In: Proceedings of the IFIP TC13 international conference on human-computer interaction, INTERACT'97. Chapman and Hall, London, pp 509–516

[15] Majaranta P, Ahola UK, Špakov O (2009a) Fast gaze typing with an adjustable dwell time. In: Proceedings of the 27th international conference on human factors in computing systems, CHI 2009. ACM, New York, pp 357–360.

[16] Frietman, E.E.E., Bruogeman, F., and Van Spronsen, C.J. 1984. Eyeball-Position-controlled Communication System EPCOS, Microprocessing and Microprogramming 14 (3), 173- 175

[17] Kate, J.H. Ten, Frietman, E.E.E., Willems, W., Ter Haar Romeny, B.M., and Tenkink, E. 1979. Eye-Switch Controlled Communication Aids, in Proceedings of the 12th International Conference on Medical & Biological Engineering, Jerusalem, Israel.

[18] P. Majaranta, K.-J. Rih, "Twenty Years of Eye Typing: Systems and Design Issues", *Proc. Symp. Eye Tracking Research and Applications*, pp. 15-22, 2002.

[19] Drewes H, Schmidt A (2007) Interacting with the computer using gaze gestures. In: Proceedings of INTERACT '07. Lecture notes in computer science, vol 4663. Springer, Heidelberg, pp 475-488.

[20] Mollenbach, E., Hansen, J., & Lillholm, M. (2013). Eye Movements in Gaze Interaction. Journal of Eye Movement Research, 6, 1–15

[21] Yamagishi K, Hori J, Miyakawa M: Development of EOG-based communication system controlled by eight- directional eye movements. Proceedings of the 28th IEEE EMBS Annual International Conference; Aug. 30 - Sept. 3, 2006 New York, USA; 2006, 2574-2577.

[22] Porta, M. and Turina, M. 2008. *Eye*-S: a full-screen input modality for pure eye-based communication. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applica- tions* (Savannah, Georgia, March 26 - 28, 2008. ETRA '08. ACM, New York, NY, 27-34.

[23] Møllenbach, E., Lillholm, M., Gail, A. and Hansen, J.P. 2010. Single gaze gestures. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (ETRA '10). ACM, New York, NY, USA, 177-180.

[24] Jacob, R. J. Eye Tracking in Advanced Interface Design. In *Virtual Environments and Advanced interface Design*, W. Barfield and T. A. Furness, Eds. Oxford University Press, New York, 1995 (http://www.cs.tufts.edu/~jacob/papers/barfield.pdf)

[25] Hutchinson, T. E., Wrote, K. P., Martin, W. N., Reichert, K. C., and Frey, L. A. 1989. Human-Computer Interaction Using Eye-Gaze Input, IEEE Transactions on Systems, Man, and Cybernetics 19 (6), 1527-1534.

[26] Yoshinori Kobayashi, Yuki Kinpara, Erii Takano, Yoshinori Kuno, Keiichi Yamazaki, and Akiko Yamazaki. 2011. Robotic wheelchair moving with caregiver collaboratively depending on circumstances. In Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA '11). ACM, New York, NY, USA, 2239-2244.

[27] Steinman, R. M. (2004), 'Gaze control under natural conditions'. In: L. M. Chalupa and J. S. Werner (eds.): The Visual Neurosciences. Cambridge, MA: The MIT Press, pp. 1339–1356.

[28] Barea, R.; Boquete, L.; Bergasa, L.M.; Lopez, E.; Mazo, M. Electrooculographic guidance of a wheelchair using eye movement codification. *Int. J. Rob. Res* 2003, *22*, 641–652

[29] C. S. Lin, C. W. Ho, W. C. Chen, C. C. Chiu, and M. S. Yeh. Powered wheelchair controlled by eye-tracking system. *Optica Applicata*, 36:401– 12, 2006.

[30] Wästlund W, Sponseller K, Pettersson O (2010) What you see is where you go: testing a gaze-driven power wheelchair for individuals with severe multiple disabilities. In: Proceedings of the 2010 symposium on eye-tracking research and applications, ETRA 2010. ACM, New York, pp 133–136

[31] D. Purwanto, R. Mardiyanto, K. Arai: " Electric wheelchair control with gaze direction and eye blinking", Proceedings of The Fourteenth International Symposium on Artificial Life and Robotics, GS21-5, B- Con Plaza, Beppu, [2008].

[32] Q. X. Nguyen and S. Jo, "Electric wheelchair control using head pose free eye-gaze tracker," Electron. Lett., vol. 48, no. 13, p. 750, 2012.

[33] Kohei Arai and Ronny Mardiyanto, Eyes Based Eletric Wheel Chair Control System, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 2, No. 12, 2011.

AUTHOR

Corten Singer – A Masters candidate at the University of California, Berkeley (email: cortensinger@berkeley.edu)

Corten Singer <cortensinger@berkeley.edu>
to Jay

10:56 PM (8 hours ago)

Hello Jay,

Long time no talk! I hope things are going well on your end up there in Redmond. I had such a blast working with you guys....truly miss it up there.

So my Master's Program is coming to an end and I am just about to submit my thesis in a few days. In my thesis, I included a photo from a blog post of yours (https://blog.jaybeavers.org/how-to-drive-a-wheelchair-with-your-eyes-generation-2/). Specifically, I included the first photo in the blog post that shows the entire wheelchair setup as an example of a previous implementation. My advisor is helping me edit the paper and mentioned that I might run into copyright issues with the UC Berkeley Graduate Division if I dont have explicit permission to use the photo.

I figured that it would be ok since your blog is posted publicly, but would you mind confirming with me that it would be ok to include your photo? If you are not ok with my use of the photo, that is totally ok and I will be very glad to have nipped this in the bud before it became an issue! An email response should suffice if you would like to confirm. I would like to be safe rather than sorry on this...Im sure you understand.

I hope you and the Enable Team are all doing well. Tell them I say hello from Berkeley!

Thanks for your help,
Corten

Jay Beavers
to me

7:11 AM (18 minutes ago)

Nice Corten!

Permission granted.

Let's catch up some day soon over a video call when you have the time. There's progress to discuss.

Get [Outlook for Android]()