EyeTAP: Introducing a Multimodal Gaze-based Technique using Voice Inputs with a Comparative Analysis of Selection Techniques

Mohsen Parisay

Department of Computer Science and Software Engineering,
1455 De Maisonneuve Blvd. W. Room EV 3.172, QC H3G 1M8,
Concordia University, Montreal, Canada
m_parisa@encs.concordia.ca

Charalambos Poullis

Immersive and Creative Technologies Lab,
Department of Computer Science and Software Engineering,
Concordia University, Montreal, Canada
charalambos@poullis.org

Marta Kersten-Oertel

Department of Computer Science and Software Engineering and PERFORM Center,
Concordia University, Montreal, Canada
marta@ap-lab.ca

Abstract

One of the main challenges of gaze-based interactions is the ability to distinguish normal eye function from a deliberate interaction with the computer system, commonly referred to as 'Midas touch'. In this paper we propose EyeTAP (Eye tracking point-and-select by Targeted Acoustic Pulse) a contact-free multimodal interaction method for point-and-select tasks. We evaluated the prototype in four user studies with 33 participants and found that EyeTAP is applicable in the presence of ambient noise, results in a faster movement time, and faster task completion time, and has a lower cognitive workload than voice recognition. In addition, although EyeTAP did not generally outperform the dwell-time method, it did have a lower error rate than the dwell-time in one of

[*] Corresponding author
Email address: m_parisa@encs.concordia.ca (Mohsen Parisay)

our experiments. Our study shows that EyeTAP would be useful for users for whom physical movements are restricted or not possible due to a disability or in scenarios where contact-free interactions are necessary. Furthermore, EyeTAP has no specific requirements in terms of user interface design and therefore it can be easily integrated into existing systems.

Keywords: Gaze-based interaction, eye tracking, Midas touch, voice recognition, dwell-time, contact-free interaction

1. Introduction

In gaze-based interaction eye tracking sensors measure a user's gaze position on a computer screen and differing methods (e.g. dwell time, multimodal interaction, etc.) are employed to allow the user to interact with the system. Gaze-based interaction offers a suitable alternative to conventional input devices (i.e. keyboard and mouse) in several different scenarios including for users for whom manual interaction might be difficult or impossible, or in situations where contact-free interaction is required. However, gaze-based interaction has well-known challenges among which is *Midas touch*, where a system cannot distinguish the basic function of the eye (i.e. looking and perceiving) from deliberate interaction with the system. In this paper, we propose EyeTAP (Eye tracking point-and-select by Targeted Acoustic Pulse), a multimodal gaze-based interaction approach that addresses the Midas touch problem by integrating the user's gaze to control the mouse with audio input captured using a microphone to trigger button-press events for real-time interaction.

Traditionally, pointing and clicking is done with a mouse; a user uses a mouse to move a cursor to a target (pointing phase), and clicks on the mouse to select or trigger a function (selection phase). We designed EyeTAP as a multimodal method point and click interaction method that uses eye gaze for pointing and auditory input for selection. Specifically, with EyeTAP the mouse pointer position is captured using an eye tracker and selection is done by generating an acoustic signal (e.g. a tongue click, microphone tap, verbal command), which in our studies was captured by a headset microphone. Our solution thus provides a contact-free interaction method for users (including those with special needs) and addresses the Midas touch problem. EyeTAP provides contact free interactions in case scenarios where the use of speech commands are not possible, e.g. due to reasons such as difficulty of word detection by user's language, accent, or pronunciations of words; or for users with severe disabilities not capable of speaking or interacting with keyboard and mouse. Figure 1 illustrates the overview of EyeTAP.

In comparison to gaze-based multimodal interactions which use gestures, foot pedals, or buttons, using speech/sound enables contact-free interactions and supports users to point and select a target based on two separate modalities by simply using a microphone. This allows for a smooth and simple-to-use interaction technique that does not require extensive equipment or training. In addition, using sound input does not require users to shift their gaze focus (e.g. to a button or other hardware device) to trigger a function.

EyeTAP's ability to use different modes of interaction for selection, such as a mouth click or a microphone tap, overcomes the limitations of natural language processing methods and is applicable when speech commands are not feasible (e.g. due to disabilities or due to the surrounding environment). We showed that EyeTAP can be an alternative to using speech with no need for voice recognition engines independent from users language or accent.

We performed four extensive user studies comparing EyeTAP to dwell-time, eye tracking with voice recognition, and mouse interaction for point-and-click tasks. The analysis of the results showed that although EyeTAP had comparable performance with other gaze-based interaction techniques, it did not outperform the dwell-time method on most criteria. At the same time, EyeTAP generally performed better than gaze-based interaction with voice recognition selection and thus might be suitable in cases where users cannot use voice commands, have restricted physical movement, or where manual interaction with an input device is not possible, e.g. medical practitioner having both hands busy or in a situation where physical contact with equipment should be avoided.

The contributions of this paper are twofold. First, we have designed and developed a simple-to-use, multimodal gaze-based interaction technique. The proposed approach allows for a completely hands-free interaction solution between the user and the computer system using only an eye-tracker and an audio input device. Second, we present four user studies comparing EyeTAP with two other widely-used gaze-based interaction techniques and the mouse.

2. Related Work

In this section, we provide an extensive literature review of gaze-based interaction techniques addressing the Midas touch problem. Although, some studies are not directly related to our proposed method, we were inspired by their intuitions and the approaches provided a broad view of both hands-on and hands-free multimodal gaze-based interaction techniques.

In eye-based interaction, the Midas touch problem occurs when a user accidentally activates a computer command using gaze when the intention was simply to look around and perceive the scene. According to Jacob [1], this problem occurs because eye movements are natural, i.e. the eyes are used to look around an object or to scan a scene, often without any intention to activate a command or function. This phenomenon is one of the major challenges in eye interaction techniques [2, 3], and diverse methods have been proposed to address the Midas touch problem. The solutions can be categorized into four groups according to the interaction technique they employ: (a) dwell-time processing, (b) smooth pursuits, (c) gaze gestures, and (d) multimodal interaction. Below, we describe each of these solutions and provide example use-cases, as well as describe their shortcomings or relationship to our work.

2.1. Dwell-time processing

Dwell-time is the amount of time that the eye gaze must remain on a specific target in order to trigger an event. Researchers have tried to detect specific thresholds to handle the Midas touch problem [4, 5]. For example, Pi *et al.*

proposed a probabilistic model for text entry using eye gaze [4]. They reduced the Midas touch problem by assigning each letter a probability value based on the previously chosen letter such that a letter with lower probability requires a longer activation time to be activated and vice-versa. Velichkovsky *et al.* applied focal fixations to resolve the Midas touch problem by assigning the mean duration time (empirically set to 325 ms) of a visual search task to trigger a function [5]. Dwell time has been shown to be even faster than the mouse in certain tasks, e.g. selecting a letter given an auditory cue [6]. The method of applying focal fixations may be very subjective since searching time varies across users when applying the dwell-time technique [7]. Moreover, increasing the threshold may increase the duration time of the entire interaction. Conversely, reducing the amount of dwell-time may lead to more errors for some users [8]. Pfeuffer *et al.* investigated visual attention shifts in 3D environments for menu selection tasks [9]. They compared three interaction techniques for menu selection: (1) dwell-time (activation threshold of 1 sec.), (2) gaze button (applying eye gaze to point, selecting by a button press), and (3) cursor (applying eye gaze to point to a context, precise movement and selecting by a manual controller). They found that the dwell-time technique was the fastest in case of performance. In addition, the cursor technique was found to be the most physically demanding technique. They also found that dwell-time was considered to be the easiest method according to users. However, the gaze button and the dwell-time caused the highest eye fatigue.

Although dwell-time has been found to be the fastest technique among eye tracking techniques, some studies [8, 10, 11] show that it is error prone particularly in situations when a lower dwell-time is used. However, longer dwell times may cause eye discomfort or fatigue [9]. For this reason, we decided to turn towards multimodal techniques to address the Midas touch problem.

2.2. Smooth pursuits

Smooth pursuits are a form of eye movement that occurs when a moving stimulus (e.g. an object or animation) is followed with gaze [12]. The method is typically implemented by using a visual point on the interface, then to activate the target the user must fixate on one of these points. This technique has been used to select targets [13], control home appliances [14], to activate functions such as mouse clicks [15] or to use the music player on a smartwatch (Orbits) [16]. Schenk *et al.* proposed a framework (GazeEverywhere) which enables users to replace mouse inputs [15]. This solution includes a computer to process gaze interactions (gaze PC), a computer to show the results (unmodified PC) which are connected via a micro-controller to trigger mouse click events, and a glass pane to project gaze targets on a second screen. Vidal *et al.* introduced an interaction technique (Pursuits) for large screens using moving objects to be activated by eye gaze [13]. They used a desktop eye tracker and a public display to select targets on the screen. Velloso *et al.* presented a framework (AmbiGaze) to control ambient devices such as TVs and stereos (each assigned with an infrared (IR) beacon) with eye gaze using a head-mounted eye tracker [14]. The system employs a server to process gaze inputs and control the devices. Esteves *et al.* presented a framework for a multi-touch Android smartwatch to input commands using a head-mounted eye tracker [16]. They developed three use-cases: a music player, a notifications panel with six colored points on the smartwatch screen representing six applications (e.g. social media apps), and a missed call menu with four commands, call back, reply text, save number and clear the notification.

Smooth pursuit gaze-based interaction has several drawbacks. First, it requires a moving stimulus [17] and therefore, it requires implementing an additional graphical user interface (GUI) to handle the events. Second, this kind of point-and-select may slow down the interaction due to the pursuit time which can add latency to target selection completion time. In addition, the presence of moving paths on a limited screen size may limit users to a restricted set of functions. Third, this type of interface may lead to visual distraction on the screen and may not be suitable for long working sessions or for users with disabilities; in fact, moving objects require free space on a screen which is therefore dependent on the screen size. Thus, although smooth pursuits is a promising method for public and large digital displays, it is not an ideal method for everyday interaction.

2.3. Gaze gestures

Gaze gestures are sequences of eye movements that follow a predefined pattern in a specific order [18]. Researchers have proposed techniques which can be applied to analyze eye movements to detect unique gestures (e.g. [19, 18, 20, 21]). Drewes *et al.* assigned up, down, left, right and diagonal directions to different characters on the keyboard thereby allowing a user to select a letter by moving the eye gaze in any direction [18]. In addition, they tried to distinguish between natural and intentional eye movements by using short fixation times during gesture detection and long fixation times to reset the gesture recognition. Istance *et al.* developed two-legged and three-legged gaze gestures (up, down and diagonal patterns) for command selection to play World of Warcraft for users with motor impairment disabilities [21]. In a similar work, Hyrskykari *et al.* studied both dwell-time and gaze gesture interactions in the context of video games and found that gaze gestures had better performance for command activation [20]. Moreover, gaze gestures produced fewer errors than the dwell-time and led to less visual distractions. Bâce *et al.* proposed an AR prototype, containing a head-mounted eye tracker and a smartwatch, to embed virtual messages to real-world objects to be shared with peer users [19]. The authors integrated eye gaze gestures as a pattern to encode and decode messages attached to a specific object previously tagged by another peer user, thus using gaze gestures as an authentication mechanism for secure communication. In general, gaze gestures have shown promising performance to address the Midas touch problem.

As gaze gesture techniques rely only on performing specific sequences of eye movements, they may lead to eye fatigue in a long working session as longer eye inputs are correlated with eye fatigue [9]. In addition, the detection algorithms may reduce the speed of interaction and the limited amount of possible eye gestures may reduce the number of functions available to users. Further, applying gaze gesture commands requires a guiding system since users need to map commands with their corresponding gestures [22]. Learning the correct gestures may also be challenging and requires training for novice users [22]. This kind of interaction solution, therefore, may not be appropriate for users who must use a system over a long period of time or for users with disabilities.

2.4. Multimodal Interaction

Multimodal techniques apply extra inputs from another modality (e.g. touch, audio, etc.) as the trigger of a function in addition to eye tracking. They can be divided into the following sub-categories: using mechanical switches, touch interaction, head movements, facial gestures, hand gestures, and gaze gestures.

2.4.1. Applying a specific (mechanical) switch

For certain specific domains, such as rehabilitation, and user groups (i.e. users with motor impairments or severe disabilities), researchers have used mechanical switches to activate an event or function. For instance, Rajanna *et al.* proposed a combined framework for users with disabilities which applies a foot pedal device to click on objects and to enter text [23]. Meena *et al.* applied a soft button on a wheelchair to control the movements of the wheelchair in different directions (horizontal, vertical and diagonal) [24]. Sidorakis *et al.* applied a switch for a gazed-controlled multimedia framework on virtual reality head-mounted displays (Oculus Rift) to resolve the Midas touch problem [25]. Biswas *et al.* proposed a joystick to control point-and-select tasks for combat aviation platforms to address the Midas touch problem [26].

2.4.2. Touch interaction

Some researchers have proposed the integration of using touch interaction, for a limited number of functions, to increase the accuracy of target selection. Pfeuffer *et al.* applied a cursor at the gaze point to be controlled by a finger holding a tablet where a finger tap on the screen leads to a click on the current location of the pointer (CursorShift method) [27]. In a similar study by Pfeuffer *et al.*, the authors investigated the integration of finger touch and pen inputs on a tablet for zooming or annotating tasks on images [28]. Although this technique was not introduced as a solution to the Midas touch problem, it can increase the accuracy of selection which leads to reducing Midas touch. Stellmach *et al.* proposed an interaction technique to select targets on a remote screen via eye gaze and a handheld touchscreen device [29].

2.4.3. Eye gaze and head movements

Stellmach *et al.* proposed multimodal techniques to interact with distant targets in which they studied combinations of gaze and head movements joint with a smartphone touch modality for precise selection and manipulations [30]. Kytö *et al.* proposed similar techniques for AR headsets. They investigated head movements and eye gaze movements with a variety of combinations including selection on device and hand gesture commands and found the highest error rates and lowest completion time for the eye only selection technique [31].

2.4.4. Facial gestures recognition

Rozado *et al.* studied the potential of using live video monitoring to detect facial gestures to enhance eye tracking interaction [32]. In their work (FaceSwitch), they associated facial gestures (opening mouth, raising eyebrows, smiling and twitching the nose up and down) to simulate left and right mouse clicks and customized some keyboard functions such as page down key press. They found that increasing the number of gestures leads to lower recognition accuracy when monitored simultaneously.

Facial gesture recognition has several drawbacks. First, real-time video monitoring to detect the correct face gesture is very challenging beyond controlled lab conditions to address the real-life scenarios [33]. In addition, any emotional change or unwanted facial behavior may lead to false activation of functions, since modeling the human behavior is challenging [33]. Another drawback is the latency between pointing using the eye tracker and selecting using the facial gesture algorithm; precise timing is required for smooth interactions. Moreover, modeling of facial expressions requires a wide range of visual signal processing

[33].

2.4.5. Gaze and speech interaction

Besides the above related works which were aimed at addressing the Midas touch problem, multimodal interaction have also considered gaze and voice commands. Mayer *et al.* proposed an interaction technique (WorldGaze) to track user's fields of view and gaze point to refine the voice command engines on smartphones for more precise results [34]. Beelders *et al.* studied word processing tasks using voice commands and eye gaze compared with mouse and keyboard interactions in their work [35]. However, although they showed the application of speech interaction is feasible for word applications, the gaze and speech interaction technique could not reach the effectiveness and performance of keyboard interaction. Acartürk *et al.* reviewed the challenges and possibilities of gaze and speech modalities for elderly users in their work [36]. Esteves *et al.* conducted comparative studies using head mounted displays (HMDs) to investigate the performance of hands-on and hands-free (including gaze and speech) interaction techniques and found that applying a clicker and dwell-time were the most favorable interaction techniques [37].

Miniotas *et al.* proposed a technique for selecting closely spaced targets based on speech commands [38]. They applied a grid of 5 × 5 squares as stimulus to test two interaction techniques: (a) gaze and speech, and (b) gaze only. They suggested a dwell-time of 1500 ms for targets of size of 30 × 30 pixels with distance of 10 pixels for the best performing setup for target selections based on their results. However, they reported a slow performance in case of selection speed when activation threshold for the dwell-time increased.

Beelders *et al.* conducted an experiment to study eye gaze and speech commands comparing to the mouse for target selection tasks [39]. They applied a stimulus as shape of a circle with 800 pixels diameter containing 16 squares on its edge to be selected in all directions. They found that the mouse had a significantly higher performance in case of throughput and completion time and also stated that using dwell-time technique should be more efficient than speech commands. Sengupta *et al.* integrated gaze and voice inputs for web browsing tasks such as search, navigation, and bookmark of pages [40]. They found the multimodal approach had a higher performance than each modality alone.

Zhao *et al.* proposed a multimodal technique of eye gaze by smooth pursuits, and speech commands and found promising results when compared to mouse clicks [41]. They found that the selection of a word for confirmation should match the task for a better performance. Further, participants who chose the activation word scored higher compared to those who used a pre-determined word. Similar to the EyeTAP method, the authors also suggested applications of other sound inputs such as pseudowords or exclamation for users with severe disabilities.

2.4.6. Gaze and hand gesture interaction

Gaze has also been combined with hand gesture inputs, for example, Chatterjee *et al.* proposed an interaction technique that uses gaze and hand gestures to select targets at the most desired location on screen [42]. They found that the combination of gaze and hand gesture outperformed each interaction modality alone. Pfeuffer *et al.* proposed a similar approach of applying eye gaze and a hand pinch to select and manipulate targets in a 3D space for virtual reality (VR) platforms [43]. Hand-gesture interactions are prone to muscular fatigue [44] and therefore may challenge users in certain circumstances.

2.4.7. Gaze and button press

Hild *et al.* investigated multimodal gaze-based interactions: gaze and button press by hand, gaze and button press by foot, and the mouse input [45]. They found overall faster performance for gaze-based techniques than the mouse for task completion time. Kumar *et al.* proposed a technique (EyePoint) comprised of eye gaze and button press on keyboard to improve the accuracy of gaze-based pointing in a Look-Press-Look-Release pattern of commands [46]. The EyePoint technique was designed in four steps to select a target accurately. The user looks at a desired target (Look), then presses and holds a hotkey on the keyboard which magnifies the specific spot on the screen (Press). A second look at the magnified scene is then done to refine the exact location of target to be selected (Look), then the key is released to select that target (Release). Gaze and button techniques have shown promising results in improving the selection accuracy.

2.4.8. Gaze gesture recognition

Istance *et al.* proposed a technique (Snap Clutch) to resolve the Midas touch problem [3]. They applied a disengagement technique to turn off gaze selections when not needed by defining four modes provided in up, left, right, and down directions on the screen. These modes are activated when looking at different directions (eye gesture) and visual feedback appear on the screen to confirm the intention.

2.5. Summary

We reviewed a wide range of techniques that can be applied with good accuracy and are suitable for specific domains with specific peripherals or extra user interface designs. The need for contact-free gaze-based interactions is necessary to deal with the emerging requirements regarding hygiene interactions from a safe distance. Building on the promising results found for multimodal techniques, and specifically exploring the use of non-speech sounds to allow for a more diverse population of users as suggested by Zhao *et al.* [41], we developed EyeTAP. EyeTAP can be applied to fill the gap for both able-bodied and disabled users with or without physical contact (to the microphone), with no need for specific user interface design or peripherals and using the simplicity of the Morse code [47] to encode/decode input signals.

3. EyeTAP Prototype

Using a multimodal solution that combines eye-gaze with acoustic inputs (audio or speech detection) can be regarded as an alternative to the reviewed literature on multimodal interaction methods and has the advantage of not requiring additional hardware (in comparison to other gaze-based techniques) other than an eye tracker or a specialized user interface design. Although there has been some work done on audio detection to simulate system events for computer interactions (e.g. [48, 49, 50]) on signal processing for complex interactions. Conversely, in our work we applied acoustic inputs only as a way of sending commands.

A simple mouse interaction consists of moving the pointer to a target (pointing phase), and clicking on it to trigger a function (selection phase). In the EyeTAP prototype the mouse pointer position is captured using an eye-tracker (in our case the Tobii 4C) and selection is done by generating an acoustic pulse by mouth (e.g. a mouth click) which is captured by a headset microphone (Logitech H370). The experiments using the EyeTAP prototype were run on a commodity computer system: 64-bit Windows 10 PC with Intel i7 2.67GHz CPU, 12 GB RAM, 1 TB hard disk and NVIDIA GeForce GTX 770 graphics card. Thus, EyeTAP is a cost-effective system that can be applied at almost any work space. Figure 1a gives an overview of the the EyeTAP system.

3.1. Eye Tracking: Pointing Phase

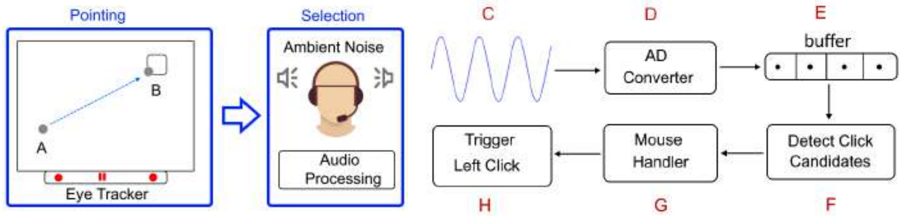
The Tobii SDK (TobiiEyeXSdk−Cpp−1.8.498) supports different events related to eye tracking activities such as providing the location of the current eye gaze, positions of both eyes, fixation points and user presence in front of the eye tracker. We employed the eye gaze library (API) to obtain users' gaze locations. These locations show the current gaze position on the screen as pixels. The SDK supports eye movements in a 3D coordinate system (horizontal, vertical, depth) but we applied a 2D coordinate system (x, y) such that the mouse cursor was synchronized with the gaze positions to control the mouse pointer on the screen. Eye tracking for the EyeTAP prototype was developed in C++ and integrated as a new plug-in into the Tobii SDK.

3.2. Auditory Processing: Selection Phase

To select a target the user makes a sound which is captured by a headset microphone. The intensity of the noise and distance of the microphone are adjusted by the user prior to using the system. A detected pulse in the real-time audio signal (amplitudes larger than a predefined threshold) is regarded as a click. The threshold's value can be adjusted based on the environment to reduce background ambient noise. When a significant increase in the signal (greater than the threshold) is detected a mouse click event is triggered as shown in Figure 1b. In general, recording is categorized into two phases: audible and silent periods. Any audible period with an intensity (amplitude) greater than the predefined threshold triggers an input signal to the system; on the other hand, values smaller than the threshold value are suppressed. Thus, any spoken sound e.g. speaking into the microphone or clicking the tongue, can trigger a click-event. Signal detection is continuous and works in real-time. The selection time-point is the moment the input pulse goes over the specified threshold at which point the click-event is triggered. This is purposedly designed to reduce possible synchronization issues resulting from eye gaze drifting away from the initial selection point. Thus our method initiates the selection phase as soon as it detects a trigger signal while the gaze pointer is still on the target.

Specifically, click detection is implemented as follow. First we capture the analog sound wave stream received from the microphone via the *AudioFormat* class provided in the Java Platform Standard Edition. 7 API [51] and digitize it using the sampling rate of 44100 Hz in a fixed buffer size of 256 bytes at a time. The buffer size is regarded as a *detection window* which is a queue for further processing. We set an empirical amplitude as threshold for pulse detection based on the available noise in the environment. Any receiving signal with an amplitude higher than the threshold is regarded as a 'click candidate' if it remains above the threshold for a minimum of 3 consecutive time-steps in which case it is considered a physical click and a mouse event is triggered. This step is necessary to enable a smooth flow of clicks in the case of noise or random vocal inputs by users and to reduce the effects of sudden noise inputs to the auditory detection API to avoid 'over clicking' events. The output of the auditory processing module is a series of 0s and 1s which are coupled with a mouse interaction event handler to trigger a left click based on 1 values. The entire workflow of the auditory module operates in real-time.

The intuition behind the auditory processing was inspired from the simplicity of the Morse code [47], which consists of a series of ON/OFF signals triggered by tone or light. In this case, information is interpreted using dots and dashes and therefore can be used to represent transmitted signals through a sequence of True/False variables. Figure 1b illustrates the step-wise operation of target selection phase by the EyeTAP technique.



(a) (b)

Figure 1: (a) The EyeTAP system: the eye tracker is used to move the pointer from A to B. The user makes an acoustic pulse and the signal processing module interprets the signal as an input and triggers a click event to select B. (b) The pipeline of the audio processing module. Analog audio waves are received from the microphone (C), and converted to a digital using an analog to digital converter (AD Converter) (D). The converted signal is stored in a fixed-sized buffer for further processing (E). A function detects the amplitudes higher than threshold as click candidates from the buffer (F). More than three click candidates in buffer are recognized as a click signal to be sent to a mouse event handler (G). Mouse handler triggers a left click (H).

3.3. *Hypotheses*

We hypothesize that a multimodal gaze-based interaction technique based on sound inputs can be applied to (a) enable a high accuracy contact-free interaction and (b) provide an alternative to mitigate the Midas touch problem. Furthermore, we hypothesize that our proposed technique will be easier to use compared to dwell-time and gaze with voice recognition and will be faster than the voice recognition technique.

4. Evaluation

To evaluate the effectiveness of the developed EyeTAP method, we ran four user studies with 33 participants (13 female, from 22 to 35 years old, $mean = 26.06$). Prior to running the experiments, subjects were informed about the purpose of the study, trained on each of the methods to be tested, and participated in a pre-test questionnaire probing them on their background in the fields of eye tracking, voice recognition technologies and their preferred kind of interaction in the case of contact-free alternatives. The Tobii calibration software was used to calibrate the system for each participant before starting the study. At the end of the user studies subjects filled out a post-test questionnaire, which consisted of the NASA TLX questionnaire [52] followed by specific questions about the subjects' perceptions of the different interaction methods. The order of interaction method was randomly selected for each participant.

We played an artificial ambient noise through stereo desktop speakers of 50 dB to simulate a typical work environment since EyeTAP and voice recognition rely on audio inputs. Participants were asked to produce a tongue click type sound ('tick') which lasted for 2 seconds on average.

To determine the effectiveness of the EyeTAP method, we analyzed the results of our experiments using an analysis of variance (ANOVA) followed by Bonferroni posthoc tests with the IBM SPSS software, and applied descriptive statistics based on dispersion with the JASP 0.11.1 software [53].

4.1. Interaction Techniques

We applied two eye tracking techniques to be compared with the performance of EyeTAP and included mouse as the baseline technique for point-and-select tasks. In other words, for all tests our independent variable is the interaction technique: (a) the mouse, (b) dwell-time, (c) eye tracking with voice-recognition, and (d) EyeTAP.

4.1.1. Mouse

For the mouse method (our baseline method for comparison), subjects simply used a mouse to move to targets and select them in numerical order.

4.1.2. Dwell-time

For the dwell-time method an internal timer was used to determine if a target was selected. Given the range of dwell-time is typically 300-1100 milliseconds for target selection [54], we defined the target activation threshold to 500 milliseconds, since this showed the best performance in [55, 54]. In other words, a target was selected when a subject focused on a target for 0.5 seconds, and if the subject moved their gaze away from the target prior to 0.5 seconds the target selection process would restart.

4.1.3. Eye Tracking with Voice recognition

For voice recognition, eye tracking was used for pointing and voice for selection. The method was developed using the built-in Windows 10 speech recognition capabilities available in the .NET framework. We implemented a C# application to respond to the activation keyword 'select' to trigger a mouse click. The same microphone was used as for the EyeTAP test.

4.2. User Study 1: Matrix-based Test

In the first user study, the EyeTAP interaction method was compared with: (a) the mouse, (b) dwell-time, and (c) eye tracking with voice-recognition. In this test, a matrix of buttons (targets), were randomly distributed across the screen. The task of the subjects was to point and click on buttons shown on the screen in increasing numerical order for various levels of difficulty from 1 (easy) to 5 (hard), described in detail below. The order of interaction methods seen by each subject was randomly selected for each participant however, the level of difficultly was presented in ascending order.

We were inspired by Miniotas *et al.*'s work that applied a stimulus composed of a grid of 5 × 5 squares [38]. The matrix grid was designed to cover a large area of the screen and to have equally-sized targets in close adjacent proximity. This enabled the analysis of errors that are most important for the Midas touch problem. Furthermore, since different areas of a screen have different accuracy in target selection for eye tracking applications [56], this test allowed us to study target selection accuracy on different areas of the screen.

4.2.1. Stimulus

The stimulus consisted of 77 buttons (11 columns × 7 rows) some labeled with numbers and others not, which covered the entire screen at a resolution of 1920 × 1080 pixels on a Dell P2411Hb monitor. Two marginal columns (far left, far right) and two rows (top, bottom) were removed from the active selection due to the high difficulty to be selected by users during the pilot-test. Buttons that were not labeled are considered as *barriers* or *distractions.* To provide feedback to the subject, labeled buttons change color after the user has successfully pointed and selected on the correct button. Wrongly selected barriers (buttons with no label) are highlighted in red. The level of difficulty of the stimulus was also increased across subject trials. This was done by increasing the number of targets that had to be selected by the subject. Five levels of difficulty were used for each interaction method: level 1 (4 targets), level 2 (6 targets), level 3 (8 targets), level 4 (10 targets) and level 5 (12 targets). Targets were randomly distributed over the entire screen for each level. Figure 2 shows the matrix-based test during difficulty level 5. The cursor that was used was a black circle because it was easier for users to keep it on the target's boundary rather than a pointer. The rationale of 'difficulty' for a higher number of targets lies in the experience that the selection of more targets caused eye fatigue for some users during the test, especially for the dwell-time method.

4.2.2. Measures

The following dependent variables were recorded: *completion time*, *path cost of selecting targets*, *error locations*, and *cognitive load* (based on the NASA TLX scores). An internal logging module recorded subjects' actions, selection times,

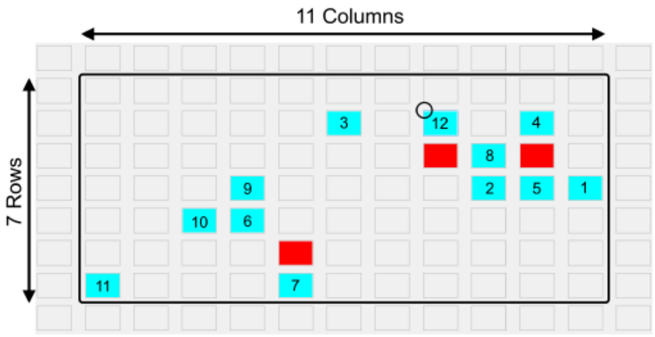


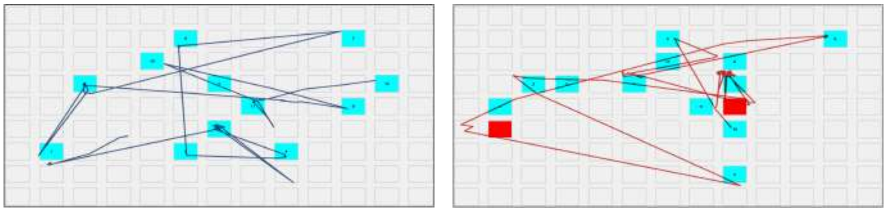
Figure 2: The matrix-based test for difficulty level 5. Target buttons are distributed randomly across the screen. The red buttons illustrate errors. The black circle on number 12 shows the current eye gaze location. Labels were enlarged for higher visibility.

as well as the number of correct and wrong selections.

For the path cost measure the shortest path between targets and the produced path by each interaction method was processed. The intuition behind this measure was to analyze the trajectory of pointer movements (footprints) of each interaction technique. In other words, since the pointer was mapped with eye gaze, we could detect which interaction technique would select targets with less eye movements (see Figure 3). This measure was specifically designed to test the hypothesis whether dwell-time requires less eye movements than multimodal techniques due to pointer drift caused by synchronization between *pointing* and *selection* phases. To compare the shapes of the generated paths, we used the dynamic time warping (DTW) algorithm [57, 58, 59]. Since DTW works on a time-value domain the paths produced by the eye tracker were decomposed into their horizontal and vertical values and compared with their associated shortest path models' X and Y values. We applied the built-in DTW function in the Python DTW 1.3.3 module [60] to measure the deviations of each path from the shortest path model.

4.2.3. Results

A two-way repeated measures ANOVA (methods × difficulty levels) was performed to examine the effect of interaction type on: (1) *completion time* and



(a) (b)

Figure 3: The path cost overview of (a) dwell-time, and (b) EyeTAP on the screen.

(2) *path costs of target selection* for each method and difficulty levels. We also analysed the distribution of each measure since it indicates the consistency of each interaction technique on most users.

Completion time: We found a significant effect of interaction method on completion time (F(12,384)=8.51, $p < .001$). A posthoc Bonferroni comparison test showed a significant difference between mouse ($M = 1.04$ *sec*, $SE = 0.02$ *sec*) and all other eye tracking methods (see Figure 4a). In addition, EyeTAP ($M = 2.57$ *sec*, $SE = 0.12$ *sec*), dwell-time ($M = 1.40$ *sec*, $SE = 0.06$ *sec*) and voice recognition ($M = 3.20$ *sec*, $SE = 0.25$ *sec*) are significantly different ($p < .05$). Figure 4a illustrates the overall completion time per method for each target.

We also looked at the distribution of values for completion time, and found a large range for both EyeTAP ($range = 8.69$ *sec*, $IQR = 0.90$ *sec*) and voice recognition ($range = 7.71$ *sec*, $IQR = 1.39$ *sec*) comparing to the mouse (0.70 *sec*, $IQR = 0.14$ *sec*) and dwell-time (1.80 *sec*, $IQR = 0.84$). The interquartile range comparison was the narrowest for mouse and highest for voice recognition, but there was a similar variability between EyeTAP and dwell-time.

Path costs of target selections: To examine the paths produced by selecting targets we compared the original locations of the targets and the shortest path (ideal path model), as described earlier. For each method, we had a $\frac{distance}{cost}$ measure to the shortest path. This metric can be regarded as the *footprint* of each interaction technique on the display. A two-way repeated measures

ANOVA (methods × difficulty levels) showed that there was a significant effect of interaction type on path cost (F(12,384)=2.57, $p < .05$). A Bonferroni posthoc test showed that dwell-time ($M = 76.73\ pixels$, $SE = 5.09\ pixels$) produced the shortest path among all other interaction techniques, even better than the mouse interaction ($M = 109.25\ pixels, SE = 3.82\ pixels$) with $p < .05$. There were no significant differences between dwell-time ($M = 76.73\ pixels$, $SE = 5.09\ pixels$), EyeTAP ($M = 84.80\ pixels$, $SE = 3.59\ pixels$) and voice recognition ($M = 82.03\ pixels$, $SE = 4.41\ pixels$). Figure 4b, which shows the path costs for all interaction methods, reveals that eye tracking movements produce significantly lower movements than mouse on a large screen. We found the highest variability in paths for dwell-time ($range = 126.81\ pixels$, $IQR = 43.13\ pixels$) and the lowest for mouse ($range = 79.21\ pixels$, $IQR = 33.26\ \ pixels$). Voice recognition ($range = 111.11\ pixels$, $IQR = 29.91\ pixels$) showed a larger range compared to EyeTAP ($range = 88.88\ pixels$, $IQR = 22.76\ pixels$). All eye tracking techniques reached a significantly lower median than the mouse which reflects a shorter path for eye gaze pointing on the screen than mouse pointing. EyeTAP reached the narrowest interquartile range for gaze path on screen among all interaction techniques which represents similar performance for most users comparing to other interaction techniques. The dwell-time method showed the highest variability and voice recognition reached the second highest variability based on the interquartile range measure.

Errors in target selections: To measure the effectiveness of each Midas touch solution we need to consider a penalty for wrongly selected neighboring targets. These targets are shown in red on the screen (see Figure 2). We projected the locations of errors per each interaction method, since difficulty level 5 has the highest number of targets (12 targets) on the screen, we illustrate the locations for this difficulty level in Figure 5. EyeTAP has the highest number of errors, however the figure reveals the potential regions of the screen which are more error prone. As shown in the figure, most errors occurred from the center towards the right side of the screen. In fact, the right side of the screen produces more errors than the left side. Moreover, the lower side produces more errors than the top side. This is similar to Feit *et al.*'s finding showing that the bottom and right regions of the screen have lower accuracy [56]. We confirm their results and also demonstrate that the same regions are also more error prone.

4.3. User Study 2: Dart-based Test

The purpose of this user study was to measure the accuracy of EyeTAP in comparison to the previously proposed eye-based interaction methods. Specifically, we wanted to focus on target selection accuracy. The task of the subject was to select, as accurately as possible, the bull's-eye of a dart target using each interaction method. In this test, the eye tracker was used for the pointing phase for each of the interaction methods, however selection of the target was triggered by different methods, i.e. dwell-time, voice command or EyeTAP acoustic signal. In order to take into consideration the fact that eye tracking has different accuracy in different regions of the monitor, we computed an average value based on five trials for each interaction method where the stimulus was shown at different areas of the screen near the center of the screen randomly. Each new randomly chosen trial began two seconds after selection of the previous target, allowing users time to change their gaze and to focus on the new target. For the dwell-time method, a countdown (from 5 to 0) representing the time left in milliseconds until the target selection was displayed and after each selection visual feedback was given to the user by showing the achieved distance to target.

4.3.1. Stimulus

The stimulus for this test consisted of a dart-like target with three circles, green (0 to 30 pixels radius), blue (30 to 60 pixels radius) and red (60 to 90 pixels radius) as shown in Figure 7a. Points within the center area i.e. green have the lowest range of distances to the bulls-eye; each other co-centric circle has a larger range of distance values. Any point lying outside the three co-centric circular areas is considered as having a fixed maximum distance of 90 pixels. For this test, a cross-hair icon was used.

4.3.2. Measures

The purpose of this test was to measure the selected point's distance on the dart target to the center of the core circle (in green), thus the accuracy (i.e. dependent variable) is measured in pixels. Since the measured trials are chosen randomly, the average is calculated to compare different methods based on accurate selection.

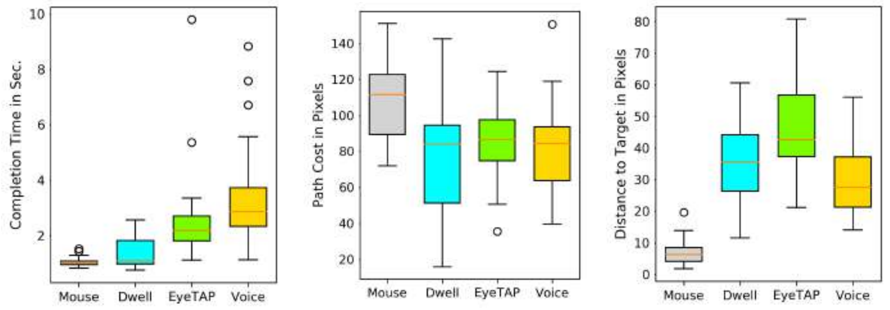
4.3.3. Results

We performed a one-way repeated measures ANOVA to compare the effect of the different interaction methods on accuracy. The results of the ANOVA showed all eye tracking methods have statistical difference (F(3,96)=104.92, $p < .001$) on selection accuracy. In fact, the mouse interaction has the lowest distance to target (highest accuracy) compared to eye tracking techniques. EyeTAP ($M = 45.11$ $pixels$, $SE = 2.28$ $pixels$) achieved the highest mean pixel accuracy compared to dwell-time ($M = 35.30$ $pixels$, $SE = 2.11$ $pixels$) and voice recognition ($M = 29.27$ $pixels$, $SE = 2.07$ $pixels$). Figure 4c depicts the results of the accuracy test.

We found the highest variability for EyeTAP on both measures ($range =$ 59.62 $pixels$, $IQR = 19.42$) among eye tracking techniques whereas the voice recognition technique reached the lowest distribution ($range = 41.05$ $pixels$, $IQR = 15.87$) and lowest distance to the target, and dwell-time ($range =$ 48.96 $pixels$, $IQR = 17.91$) showed a higher distribution than mouse ($range =$ 17.76 $pixels$, $IQR = 4.39$).

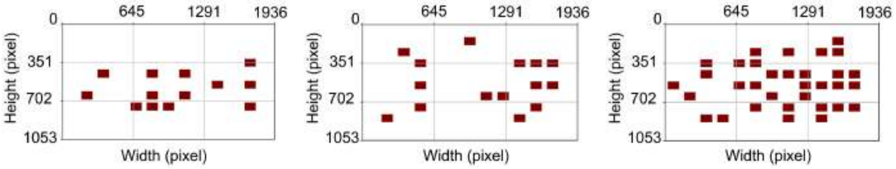
4.4. User Study 3: Ribbon-shaped Test

In order to compare our method to other gaze-based techniques, we measured the performance target selection based on the Fitts' law [61]. This study is used to analyze pointing interaction methods in accordance to well-established academic standards. As part of this study, we measured three metrics to compare the performance of all interaction techniques for point-and-select tasks, (1)



(a) (b) (c)

Figure 4: (a) Completion time of point-and-select tasks for each target ($p < .001$). (b) Path cost comparison calculated using the dynamic time warping (DTW) algorithm. All eye tracking techniques have shorter path lengths than mouse interaction for traversing items on a screen for matrix-based user study ($p < .05$). (c) The distance to target in pixels for dart-based test ($p < .001$).



(a) Dwell-Time (b) Voice Recognition (c) EyeTAP

Figure 5: The locations of errors on the screen during the matrix-based user study (see Figure 2) for difficulty level 5. The right side of the screen as well as bottom side are more error prone than the left and top sides.

throughput (how good a selection technique operates), (2) *movement time* and (3) *error rates* for ribbon-shaped targets (see Figure 7b).

The intuition of this test was to test interaction techniques based on the Fitts' law with rectangular buttons ('FittsStudy' application [62]).

4.4.1. Stimulus

The stimulus for this test consisted of two ribbon-shaped buttons to be selected on the left and right sides of the screen with random widths and distances as shown in Figure 7b. The test sessions includes three distances (256, 384, 512) pixels, and two widths (96, 128) pixels.

4.4.2. Measures

The following dependent variables were recorded: *movement time*, *throughput*, and *error rates* for this test. We applied the 'FittsStudy' application by Wobbrock *et al.* [62] for this test.

4.4.3. Results

A one-way repeated measures ANOVA was performed to examine the effect of interaction type on: (1) *movement time*, (2) *throughput* and (3) *error rates* for each interaction method. We also analysed the distribution of each measure since it indicates the consistency of each interaction technique on most users.

Movement time: We found a significant effect of the interaction method on movement time (F(3,96)=69.42, $p < .001$). A posthoc Bonferroni comparison test showed a significant difference between mouse ($M = 684.15$ ms, $SE = 16.80$ ms) and all other eye tracking methods (Figure 6a). In addition, among all eye tracking methods, dwell-time ($M = 599.39$ ms, $SE = 18.76$ ms) achieved significantly lower movement time than EyeTAP ($M = 1794.89$ ms, $SE = 170.90$ ms) and voice recognition ($M = 2014.20$ ms, $SE = 89.28$ ms) techniques. However, there is no statistical significance between EyeTAP and voice recognition. The lower movement time of dwell-time method compared to mouse interaction is associated with the low activation time (500 ms).

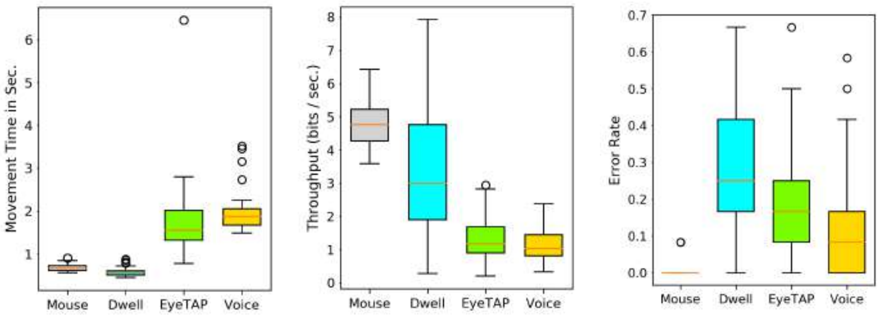
We found the highest variability for EyeTAP ($range = 5.67$ sec, $IQR = 0.69$ sec) among all interaction techniques, whereas dwell-time ($range = 0.42$ sec, $IQR = 0.09$ sec) and voice recognition ($range = 2.03$ sec, $IQR = 0.37$ sec) reached lower distributions among eye tracking techniques. The mouse reached the narrowest range ($range = 0.34$ sec) but larger interquartile range ($IQR = 0.11$ sec) than dwell-time. We found the dwell-time as the best interaction technique based on the movement time measure for the ribbon-shaped test as illustrated in Figure 6a.

Throughput: We found a significant effect of the interaction method on throughput ($F(3,96) = 75.13$, $p < .001$). A posthoc Bonferroni comparison test showed a significant difference between dwell-time ($M = 3.30$ $bits/sec$, $SE = 0.36$ $bits/sec$) and all eye tracking methods (Figure 6b). The mouse ($M = 4.81$ $bits/sec$, $SE = 0.11$ $bits/sec$) achieved higher throughput than the eye tracking methods. However, there is no statistical difference between voice recognition ($M = 1.15$ $bits/sec$, $SE = 0.09$ $bits/sec$) and EyeTAP ($M = 1.34$ $bits/sec$, $SE = 0.12$ $bits/sec$).

We found that EyeTAP ($range = 2.73$ $bits/sec$, $IQR = 0.78$ $bits/sec$) had the narrowest range of values for throughput, and dwell-time ($range = 7.64$ $bits/sec$, $IQR = 2.86$ $bits/sec$) the highest variability based on both measures among all interaction techniques. The voice recognition ($range = 2.043$ $bits/sec$, $IQR = 0.63$ $bits/sec$) reached lower variability than mouse ($range = 2.83$ $bits/sec$, $IQR = 0.95$ $bits/sec$) on both measures. However, both EyeTAP and voice recognition reached lower throughput than dwell-time on average, dwell-time reached the highest variability due to having a sparse distribution compared to the other interaction techniques.

Error rates: We found a significant effect of interaction method on error rates ($F(3,96) = 27.15$, $p < .001$). A posthoc Bonferroni comparison test showed a significant difference between mouse ($M = 0.01$ $errors$, $SE = 0.005$ $errors$) and all eye tracking interactions (see Figure 6c). In addition, dwell-time ($M = 0.28$ $errors$, $SE = 0.03$ $errors$) reached a higher error rate than EyeTAP ($M = 0.18$ $errors$, $SE = 0.02$ $errors$) and voice recognition ($M = 0.10$ $errors$, $SE = 0.02$ $errors$).

We also analysed the distribution of errors among users and found that EyeTAP ($range = 0.66$ $errors$, $IQR = 0.16$ $errors$) had a similar range compared to dwell-time ($range = 0.66$ $errors$, $IQR = 0.25$ $errors$) but lower variability based on the interquartile range measure. The voice recognition technique ($range = 0.58$ $errors$, $IQR = 0.16$ $errors$) showed a narrower range than EyeTAP but similar variability based on the interquartile range measure. The mouse ($range = 0.08$ $errors$, $IQR = 0.00$ $errors$) reached the lowest variability based on both measures among all interaction techniques. The voice recognition technique reached the lowest distribution of errors among eye tracking techniques based on error rates as illustrated in Figure 6c.



(a)

(b)

(c)

Figure 6: (a) Calculated movement time, (b) throughput, and (c) the error rates per method for the ribbon-shaped test. For all measures $p < .001$.

4.5. User Study 4: Circle-shaped Test

This test is similar to the Ribbon-shaped test, however, contains different target shapes. Figure 7c illustrates the screenshots of this test which contains uni-variate endpoint deviation (SD_x) through X axis and bi-variate endpoint deviation ($SD_{x,y}$) through both $X,\ Y$ axes for throughput calculations which results in better Fitts' law model [62]. The 'FittsStudy' application by Wobbrock *et al.* [62] was used for this test.

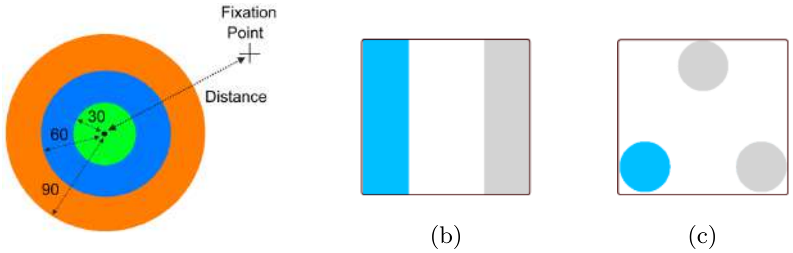
The intuition of this test was to test the interaction techniques based on the Fitts' law with circular buttons provided by the 'FittsStudy' application [62].

4.5.1. Stimulus

The stimulus for this test consisted of three circle-shaped buttons to be selected located in the middle of the screen with random widths and distances as shown in Figure 7c. The test sessions includes three distances (256, 384, 512) pixels, and two widths (96, 128) pixels.

4.5.2. Measures

The following dependent variables were recorded: *movement time*, *throughput* (with two variations), and *error rates* for this test.



(a)

(b)

(c)

Figure 7: (a) Shows the Dart-based test stimuli: the accuracy is highest in the green area. The cross-hair icon indicates the correct eye gaze location, (b) Illustrates the ribbon-shaped stimuli, and (c) shows the circle-shaped stimuli of the 'FittsStudy' application [62]. Targets highlighted in blue represent active targets to be selected.

4.5.3. Results

A one-way repeated measures ANOVA was performed to examine the effect of interaction type on: (1) *movement time*, (2) *throughput* and (3) *error rates* for each interaction method. This test is similar to ribbon-shaped test but contains an extra metric to measure throughput of each method.

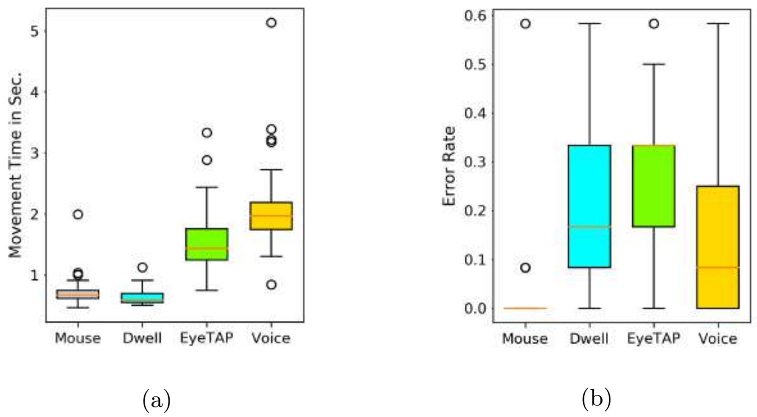
Movement time: We found a significant effect of the interaction method on movement time (F(3,96)=67.48, $p < .001$). A posthoc Bonferroni comparison test showed a significant difference between EyeTAP ($M = 1578.95\ ms$, $SE = 95.34\ ms$), dwell-time ($M = 638.80\ ms$, $SE = 24.35\ ms$), voice recognition ($M = 2123.35\ ms$, $SE = 132.42\ ms$) and mouse ($M = 727.91\ ms$, $SE = 46.12\ ms$). However, there is no statistical difference between mouse ($M = 727.91\ ms$, $SE = 46.12\ ms$) and dwell-time ($M = 638.80\ ms$, $SE = 24.35\ ms$). Figure 8a illustrates the mean movement time per method for the circle-shaped test.

We found that dwell-time ($range = 0.62\ sec$, $IQR = 0.15\ sec$) has the narrowest, and voice recognition ($range = 4.29\ sec$,$IQR = 0.44\ sec$) the largest range. EyeTAP ($range = 2.58$ sec, $IQR = 0.51$ sec) showed a narrower range than voice recognition but larger interquartile range than voice recognition, dwell-time and mouse ($range = 1.53$ sec, $IQR = 0.12$ sec). This analysis shows higher consistency for dwell-time compare to the other interaction techniques.

Error rates: We found a significant effect of the interaction method on error rates ($F(3, 96) = 18.25$, $p < .001$). A posthoc Bonferroni comparison test showed a significant difference between mouse ($M = 0.02$ $errors$, $SE = 0.01$ $errors$), dwell-time ($M = 0.23$ $errors$, $SE = 0.03$ $errors$), voice recognition ($M = 0.13$ $errors$, $SE = 0.02$ $errors$) and EyeTAP ($M = 0.28$ $errors$, $SE = 0.02$ $errors$). Voice recognition ($M = 0.13$ $errors$, $SE = 0.02$ $errors$) reached the lowest error rate among eye tracking methods, however, there is no statistical difference between dwell-time ($M = 0.23$ $errors$, $SE = 0.03$ $errors$) and EyeTAP ($M = 0.28$ $errors$, $SE = 0.02$ $errors$). Figure 8b illustrates the calculated error rates for the circle-shaped test.

We found that mouse ($range = 0.58$ $errors$, $IQR = 0.0$ $errors$), dwell-time ($range = 0.58$ $errors$, $IQR = 0.25$ $errors$), voice recognition ($range = 0.58$ $errors$, $IQR = 0.25$ $errors$), and EyeTAP ($range = 0.58$ $errors$, $IQR = 0.16$ $errors$) showed the same variability based on range measure, but EyeTAP reached a lower distribution based on the interquartile range among eye tracking techniques.

Throughput: Since the circle-shaped test contains two variations (uni-variate, bi-variate) to measure throughput [62], we ran a two-way repeated measures ANOVA (throughput × variation) and found a significant effect of the interaction method on throughput (F(3,96)=19.75, $p < .001$). A posthoc Bonferroni comparison test showed a significant difference between mouse ($M = 4.16$ $bits/sec$, $SE = 0.18$ $bits/sec$), dwell-time ($M = 3.20$ $bits/sec$, $SE = 0.25$ $bits/sec$), voice-recognition ($M = 1.24$ $bits/sec$, $SE = 0.07$ $bits/sec$) and EyeTAP ($M = 1.04$ $bits/sec$, $SE = 0.13$ $bits/sec$). However, there is no statistical difference between voice-recognition ($M = 1.24$ $bits/sec$, $SE = 0.07$ $bits/sec$) and EyeTAP ($M = 1.04$ $bits/sec$, $SE = 0.13$ $bits/sec$). Figure 9a shows uni-variations of throughput, and Figure 9b shows the bi-variations of



(a)

(b)

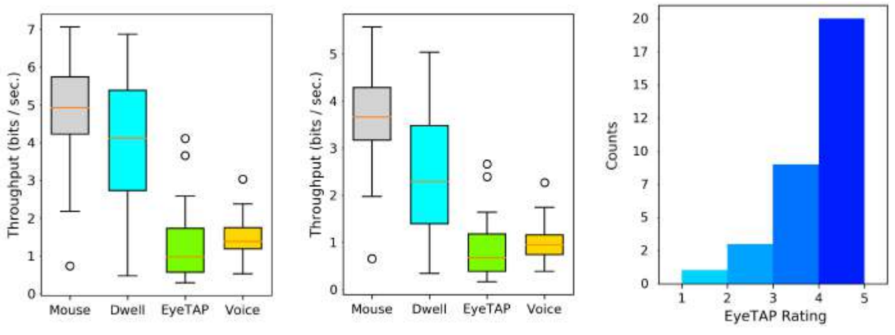
Figure 8: (a) Calculated movement time, and (b) error rates per method for the circle-shaped test. For all measures ($p < .001$).

throughput per interaction method.

We found that dwell-time ($range = 6.40\ bits/sec$, $IQR = 2.66\ bits/sec$) showed the highest variability among all interaction techniques based on both measures, range and interquartile range for uni-variation throughput measure. Whereas, voice recognition ($range = 2.50\ bits/sec$, $IQR = 0.55\ bits/sec$) showed the lowest variability. EyeTAP ($range = 3.81\ bits/sec$, $IQR = 1.16\ bits/sec$) showed lower variability than mouse ($range = 6.32\ bits/sec$, $IQR = 1.51\ bits/sec$) on both measures as illustrated in Figure 9a.

We found that dwell-time ($range = 4.69\ bits/sec$, $IQR = 2.08\ bits/sec$) and mouse ($range = 4.91\ bits/sec$, $IQR = 1.11\ bits/sec$) showed the highest variability on both range and interquartile range measures. Whereas voice recognition ($range = 1.88\ bits/sec$, $IQR = 0.42\ bits/sec$) and EyeTAP ($range = 2.49\ bits/sec$, $IQR = 0.79\ bits/sec$) showed lower variability for the bi-variate throughput measure as illustrated in Figure 9b.

This analysis confirms that EyeTAP has the lowest throughput based on mean value, and voice recognition has the lowest distribution (higher consistency) among all interaction techniques for throughput measure based on both uni-variation and bi-variation of the circle-shaped user study (see Figure 9).



(a) (b) (c)

Figure 9: (a) Calculated throughput for uni-variate, (b) throughput for bi-variate per method for the circle-shaped test, and (c) shows the ratings of EyeTAP from 1 (worst) to 5 (best) for 33 participants. For all measures in (a) and (b) ($p < .001$).

5. Results

5.1. EyeTAP Rating by Users

We asked participants to evaluate the overall performance of EyeTAP in the post-test questionnaire on a scale from 1 (worst) to 5 (best). EyeTAP reached the average rate of 3.64 (SD = 0.99) by 33 users. Figure 9c illustrates the subjective ratings obtained from the post-test questionnaire.

5.2. NASA TLX Scores

Figure 10 shows the NASA TLX scores for all interaction methods obtained during the user study. The overall workload is the average of scale values since we assume all scales equally important and therefore eliminated the weighting calculation to apply a simplified version [63] of the basic NASA TLX ratings [52]. According to our findings, the dwell-time method has the lowest workload among other eye tracking techniques. However, EyeTAP shows relatively lower workload compared to the voice recognition technique.

5.3. Comparative Scores

We analyzed the results of the eye tracking techniques based on (1) the analysis of variance (ANOVA), and (2) the descriptive statics based on dispersion

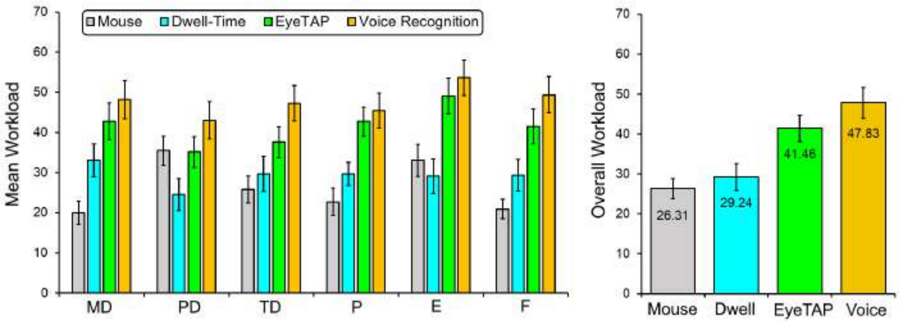
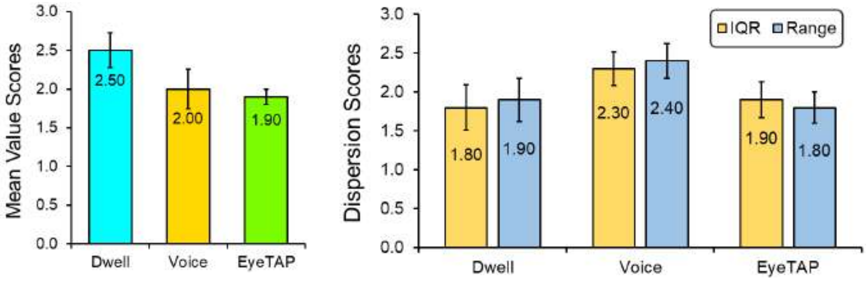


Figure 10: The NASA TLX scores for the interaction methods. (Left) Comparison of each method based on different scales. (Right) The overall mean workload of tested interaction methods. Error bars represent standard error.

of data, as illustrated earlier in this section. Since we measured the interaction techniques based on various criteria, we need to obtain a single measure comprised of all reviewed measures for comparison. Therefore, we applied a simple scoring technique and assigned an integer value in the set of {1 (worst), 2 (medium), 3 (best)} to eye tracking techniques based on their performance and calculated the arithmetic average for each interaction techniques of the entire criteria. Furthermore, we assigned the value of 2 (medium) to interaction techniques when they showed statistically similar or very close performance. Table 1 shows the details of this scoring technique for the ANOVA-based measures, and Table 2 contains the details of dispersion analysis scoring. The higher the calculated average score shows the better performance of the entire measures.

Figure 11a illustrates the results of Table 1 and Figure 11b shows the calculated average of both measures (range and IQR measures) of Table 2. The dwell-time reached the highest score (the best performance) based on the average value of objective measures of our user studies, although the difference between voice recognition and EyeTAP is not significant. However, EyeTAP and voice recognition reached relatively higher scores (higher consistency) than the dwell-time method based on dispersion analysis, however, the differences are not statistically significant. We showed that dwell-time performs very well for some participants, but shows sparse distribution on some criteria. Furthermore, EyeTAP may be considered as an interaction technique that has potential for improvement and can be adapted for most participants with sufficient training.



(a)

(b)

Figure 11: (a) Calculated scores from 1 (worst) to 3 (best) on all objective measures for eye tracking techniques shown in Table 1. The dwell-time method shows the highest scores based on ANOVA analysis results. (b) The calculated scores of average of both dispersion analysis results (range and IQR measures) shown in Table 2. Higher scores are better in both figures.

6. Discussion

Regarding the experiments with the reviewed Midas touch solutions, we found several benefits and disadvantages of each method. We discuss each method individually.

6.1. EyeTAP

We found several benefits of using EyeTAP in comparison to the other interaction techniques. First of all, it has no dependent features, rather it requires only an acoustic pulse (making a sound) near a microphone to send a signal. In fact, the output of EyeTAP in a noisy environment can appear deterministic after a number of repetitions. According to the results of our study, it achieved faster completion time in the matrix-based test, and faster movement time in the circle-shaped test than voice recognition. In addition, it showed a similar path cost (pointer footprint on display) with the other eye tracking techniques. It also achieved lower cognitive workload in comparison to the voice

	Mean Values		
Criteria	Dwell	Voice	EyeTAP
Comp. Time	3	1	2
	(1.40)	(3.20)	(2.57)
Path Costs	2	2	2
	(76.73)	(82.03)	(84.80)
Distance	2	3	1
	(35.30)	(29.27)	(45.11)
MT_{Ribbon}	3	1	2
	(0.59)	(2.01)	(1.79)
TP_{Ribbon}	3	2	2
	(3.30)	(1.15)	(1.15)
ER_{Ribbon}	1	3	2
	(0.28)	(0.10)	(0.18)
MT_{Circle}	3	1	2
	(0.63)	(2.12)	(1.57)
$\mathrm{TP}_{Circle-uni}$	3	2	2
	(3.90)	(1.48)	(1.24)
$\mathrm{TP}_{Circle-bi}$	3	2	2
	(2.50)	(1.00)	(0.84)
ER_{Circle}	2	3	2
	(0.23)	(0.13)	(0.28)
Average	2.50	2.00	1.90

Table 1: Summary of scores per interaction techniques based on comparison of their mean values. Scores are integer values from 1 (worst) to 3 (best). Statistically similar mean values ($p > .05$) were assigned the value of 2. Values represented in parenthesis denote the mean values of each measure. MT, TP, and ER represent movement time, throughput, and error rates.

	R			**IQR**		
Criteria	Dwell	Voice	EyeTAP	Dwell	Voice	EyeTAP
Comp. Time	3	2	1	3	1	2
	(1.80)	(7.71)	(8.69)	(0.84)	(1.39)	(0.90)
Path Costs	1	2	3	1	2	3
	(126.81)	(111.11)	(88.88)	(43.13)	(29.91)	(22.76)
Distance	2	3	1	2	3	1
	(48.96)	(42.05)	(59.62)	(17.91)	(15.87)	(19.42)
MT_{Ribbon}	3	2	1	3	2	1
	(0.42)	(2.03)	(5.67)	(0.09)	(0.37)	(0.69)
TP_{Ribbon}	1	3	2	1	3	2
	(7.64)	(2.04)	(2.73)	(2.86)	(0.63)	(0.78)
ER_{Ribbon}	2	3	2	1	2	2
	(0.66)	(0.58)	(0.66)	(0.25)	(0.16)	(0.16)
MT_{Circle}	3	1	2	3	2	1
	(0.62)	(4.29)	(2.58)	(0.15)	(0.44)	(0.51)
$\mathrm{TP}_{Circle-uni}$	1	3	2	1	3	2
	(6.40)	(2.50)	(3.81)	(2.66)	(0.55)	(1.16)
$\mathrm{TP}_{Circle-bi}$	1	3	2	1	3	2
	(4.69)	(1.88)	(2.49)	(2.08)	(0.42)	(0.79)
ER_{Circle}	2	2	2	2	2	3
	(0.58)	(0.58)	(0.58)	(0.25)	(0.25)	(0.16)
Average	1.90	2.40	1.80	1.80	2.30	1.90

Table 2: Summary of scores per interaction techniques based on comparison of dispersion on both measures (1) range (R), and (2) interquartile range (IQR) values. Scores are integer values from 1 (worst) to 3 (best). We assigned value of 2 for similar mean values. Values represented in parenthesis denote the actual values of each measure. MT, TP, and ER represent movement time, throughput, and error rates.

recognition technique. Furthermore, EyeTAP was a popular choice of interaction (36.4%) compared to voice recognition (9.1%). However, EyeTAP showed relatively lower accuracy and higher error rates than voice recognition, perhaps due to the fact most users had no prior experiences with this kind of interaction. Suggesting that with more training the performance of EyeTAP could be improved.

EyeTAP achieved the lowest variability for path cost of pointer movements on screen for the matrix-based test. In addition, it showed lower variability than dwell-time and mouse on throughput measures of both ribbon-shaped and circle-shaped test. The low variability of EyeTAP reflects the predictability of its performance on subjects, thus this method can be adopted for different users or different case scenarios. In general, EyeTAP allows for point-and-select interaction because it separates the actions of *pointing* and *selecting* to two different modalities while relaxing the requirement for accurate voice recognition. The results of our user study demonstrate that EyeTAP is a feasible alternative interaction technique. Moreover, it is a viable and effective solution to the Midas touch problem for eye tracking platforms and can be regarded as an alternative to voice recognition technique. EyeTAP showed the similar dispersion on average based on both measures *range*, and *interquartile range* (IQR) with dwell-time as shown in Table 2 and Figure 11b.

However, the range of activation threshold for the dwell-time method is reported in the range of (300-1100 ms) in the literature [54]. Compared to a 500 ms dwell-time, EyeTAP showed acceptable results. To our surprise however, EyeTAP did not generally outperform dwell-time in terms of either time or errors. This may suggest that a well-tuned dwell-time method even on commercial hardware components does not suffer greatly from the Midas touch problem.

EyeTAP showed a lower error rate than the dwell-time in the ribbon-shaped test (see Figure 6c) with relatively large targets. We posit that with larger size targets, the eyes to move around the target causing the dwell-time method to have more errors. Conversely, target size should not impact EyeTAP as much, as the selection is multimodal so as soon as the eye is on target the user can confirm the selection with a sound. These features caused the reduction of wrong selections by users to select relatively large targets in a left-right shift of movements applying the EyeTAP technique. In contrast, selecting smaller-sized targets in different orientations on the screen (360 degrees) of the circle-shaped test (see Figure 7c) caused a larger number of errors for EyeTAP compared to dwell-time and voice recognition. These show that EyeTAP is more suitable to select larger targets with eye movements in opposite directions (left-right, up-down) based on error rates.

EyeTAP is an effective and robust alternative to previous gaze-based interaction techniques. It may be more robust than voice-based techniques and cause less fatigue than the dwell-time method. Based on our study results, we believe it would be particularly useful when there is ambient noise, or users feel uncomfortable speaking out loud, such as the case in a communal workplace.EyeTAP showed a lower variability than the voice recognition technique, and a comparable variability to the dwell-time technique based on dispersion analysis (see Figure 11b) when applied on participants which is beneficial to apply EyeTAP on different users.

Another advantage of EyeTAP relies on its dual-purpose applications for able-bodied and severely disabled users who may not use a voice recognition engine to send their commands and has also difficulties using a dwell-time technique for their basic interaction needs.

Finally, the interesting advantage of EyeTAP lies in its fundamental auditory technique which is based on the Morse code [47] which enables a series of commands based on binary input variables. This feature provides an extension of new commands from simple to complex functionalities which offers a design flexibility for future applications and case scenarios. Although currently, EyeTAP is designed for selection tasks only, its functionalities can be extended. EyeTAP can be considered as a competitive alternative to speech recognition techniques for selection tasks. Furthermore, when users are uncomfortable using a mouth sound (and having the physical capacity to do so), they can tap the microphone to initiate the required acoustic pulse for selection.

6.2. Voice Recognition

This interaction method showed relatively acceptable results but suffers from some limitations. In general, a voice recognition engine depends on the user's voice, gender, language, and accent. Additionally, it is not applicable to users with speech impediments. Another drawback is the need of prior training samples to detect words correctly. Furthermore, similar words may lead to false recognition as we experienced during our user study. The quality of the microphone and its distance to the user is also another factor to be considered for this kind of interaction. Regarding the accuracy of recognition, the choice of recognition software plays an important role. Finally, speaking commands out loud may not be suitable in certain working environments.

In general, voice recognition presented some challenges for the users in terms of wrongly recognized words, need for action word repetition, and delay between input and feedback. The subjects' rating of this technique was very low (9.1%) in our user study. Voice recognition showed the highest completion time in the matrix-based test and highest movement time in the circle-shaped test and reached the highest cognitive workload among all interaction techniques.

The lowest error rates in both Fitts' studies reflect that the voice recognition technique is easier to control than EyeTAP and dwell-time to select targets (see Figures 6c and 8b). Voice recognition had the highest selection accuracy measured by the dart-based test. This suggests that it may be a well-suited interaction technique when on small screens and/or with small-sized targets. In addition, the voice recognition technique reached the lowest variability based on our dispersion analysis on distance to target (as shown in Figure 4c and Table 2), and throughput measures (shown in Figures 6b, 9a, and 9b and Table 2) among all eye tracking techniques. The voice recognition technique achieved the highest score based on dispersion analysis as shown in Table 2, and Figure 11b. These show its adaptability on different users which is a useful feature to apply it on a larger population with a predictable performance for suitable case scenarios.

Beelders *et al.* stated that using the dwell-time technique should be more efficient than speech commands [39]. However, we have shown that speech commands have better performance for error rates (see Figures 6c, 8b), selection accuracy (see Figure 4c), and higher consistency on users based on dispersion analysis (see Figure 11b). Zhao *et al.* experienced issues with their voice recognition engine such as speaking words loudly [41], we also had the same difficulties in our experiments. This is one of the challenges of voice recognition engines.

6.3. Dwell-Time

The dwell-time method showed the fastest completion time in the matrix-based test, and fastest movement time and highest throughput in both Fitts' experiments due to the low amount of activation time (500 ms). In addition, it reached the lowest amount of cognitive workload. However, it showed the highest error rates in the ribbon-shaped test and with EyeTAP in the circle-shaped test. Moreover, some users complained about eye fatigue after a while during test sessions. Since the dwell-time method relies on the activation time, any changes may produce different results.

We believe that the reason for faster completion time for dwell-time relates to the fact that it has a singular activation function which demands significantly lower cognitive workload (see Figure 10) to select targets at different locations, whereas the multimodal technique relies on mental coordination between both modalities to point and select a target. We posit that the synchronization of these modalities was a major factor in dwell-time outperforming the EyeTAP technique on most measures.

The dwell-time technique showed the lowest variability on task completion time and movement time measures among all eye tracking techniques, but the highest variability on path cost of target selection, throughput of both ribbon-shaped and circle-shaped tests and the highest variability on error rates of the ribbon-shaped test. This method reached similar variability as EyeTAP based on both measures *range*, and *interquartile range* (IQR) as shown in Table 2 and Figure 11b. Except the high error rates for the dwell-time method, it has been shown to be comparable with the mouse interaction for target selections in our studies which makes it still a superb eye tracking interaction technique. However, the EyeTAP technique showed competitive performance compared to the voice recognition technique with promising results. Pfeuffer *et al.* found the dwell-time the fastest technique in their study [9]. We confirm their findings regarding the completion time in our user studies for the dwell-time technique. However, they found dwell-time eye tiring and the least favorable technique by users due to relatively high activation time (1 sec). In contrast, we found the lowest workload for the dwell-time based on the NASA TLX scores (see Figure 10) but had similar feedback about eye fatigue. Since we employed half of the activation threshold used in Pfeuffer *et al.*'s experiment, dwell-time was found to be the easiest and fastest technique among eye tracking techniques in our user studies. In another work for head mounted displays (HMDs), Esteves *et al.* found a dwell-time of 400 ms a faster interaction technique than applying a clicker and speech commands [37]. We confirm their findings based on our user studies' results. Moreover, they found the dwell-time and clicker the most popular interaction techniques by users. We found relatively high error rates for dwell-time in our studies. Esteves *et al.* showed that increasing the activation threshold for dwell-time (400 ms to 1 sec) can decrease error rates to zero. These confirm that the choice of activation threshold is a key factor in applying the dwell-time method which is a trade-off between performance and error rates.

Miniotas *et al.* applied a dwell-time of 1500 ms in their experiments and showed the lowest error rate for that threshold [38]. However, although increasing the dwell-time may reduce error rates, it may also cause eye fatigue as we experienced in our user studies, especially during long-time sessions. The dwell-time method with 500 ms threshold is regarded as the best performing version of dwell-time [55].

6.4. The Mouse

We applied the mouse interaction as a baseline technique for comparison with the gaze-based techniques. Overall, we found higher performance for mouse interaction, however, it showed higher pointer movements on the screen (see

Figure 4b) than eye tracking techniques. Beelders *et al.* found that mouse interaction has significantly higher performance than eye tracking techniques in the case of throughput and completion time. We confirm these findings, however we also found that in the case of completion time, the dwell-time technique reached similar performance (see Figures 6a, 8a). These show the potentials of a fine-tuned dwell-time technique as an alternative for the mouse.

7. Conclusion and Future Work

In this paper, we proposed EyeTAP (Eye tracking point-and-select by Targeted Acoustic Pulse), an eye tracking interface that addresses the Midas touch problem with acoustic input detection capabilities. The performance of the prototype was measured in four user studies with 33 participants based on eight criteria: (1) *completion time*, (2) *path cost of target selection*, (3) *error rate*, (4) *error locations on screen*, (5) *accuracy of target selection*, (6) *movement time*, (7) *throughput*, and (8) *cognitive workload*.

In addition, we performed a statistical analysis based on (1) variance, and (2) dispersion of data. The results of our user studies showed that the dwell-time method outperformed other eye tracking techniques, including EyeTAP on most criteria based on an analysis of variance (ANOVA), but suffers from a high level of distribution on some criteria. At the same time we found that EyeTAP, in comparison to the other tested methods provides a faster task completion time, faster movement time and lower workload than voice recognition. In addition, EyeTAP showed similar performance compared to the dwell-time method and a lower error rate in the ribbon-shaped test.

Moreover, our study showed that eye tracking has a lower footprint (eye gaze mapped with mouse pointer) on the screen compared to a mouse pointer in time scale. Additionally, we confirmed that center regions towards the right and bottom side of the screen are more error prone than the left and top sides. Finally, we developed two user tests (Matrix-based, and Dart-based tests) that would be effective in studying different target selection in gaze-based interaction techniques.

Although we only developed the left mouse click event, EyeTAP demonstrates a completely contact-free alternative to mouse interaction for users with disabilities and users who need to avoid physical contact with input devices considering their workplace or situation. Thus, we believe EyeTAP can be regarded as a competitive technique to both dwell-time, specifically in cases where users may experience physical disabilities or restrictions, and voice recognition, particularly when dealing in workplaces, accents or speech disabilities. EyeTAP showed a higher consistency (lower variability) based on the dispersion analysis, thus it may be more easily accessible to a larger diverse population (e.g. children, users with disabilities, and elderly users).

The global outbreak of COVID-19 showed the importance of contact-free interactions, specifically in public places and for healthcare personnel. The potential of EyeTAP can be considered on public devices such as ATM machines and self check-in platforms at airports. We hope, that EyeTAP inspires researchers into developing contact-free interaction techniques for emerging case scenarios and equipment. In future work, we will apply the EyeTAP technique on AR/VR headsets to measure its usability in different case scenarios for able-bodied and participants with motor disabilities.

Acknowledgement

This research is supported in part by the Natural Sciences and Engineering Research Council of Canada Grants DG-N01670 and DG-N06722.

References

[1] R. J. K. Jacob, What you look at is what you get: Eye movement-based interaction techniques, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '90, ACM, New York, NY, USA, 1990, pp. 11–18. doi:10.1145/97243.97246.
URL http://doi.acm.org/10.1145/97243.97246

[2] R. J. Jacob, K. S. Karn, Eye tracking in human-computer interaction and usability research: Ready to deliver the promises, in: The mind's eye, Elsevier, 2003, pp. 573–605.

[3] H. Istance, R. Bates, A. Hyrskykari, S. Vickers, Snap clutch, a moded approach to solving the midas touch problem, in: Proceedings of the 2008 symposium on Eye tracking research & applications, 2008, pp. 221–228.

[4] J. Pi, B. E. Shi, Probabilistic adjustment of dwell time for eye typing, in: 2017 10th International Conference on Human System Interactions (HSI), IEEE, 2017, pp. 251–257. doi:10.1109/HSI.2017.8005041.

[5] B. B. Velichkovsky, M. A. Rumyantsev, M. A. Morozov, New solution to the midas touch problem: Identification of visual commands via extraction of focal fixations, Procedia Computer Science 39 (2014) 75–82.

[6] L. E. Sibert, R. J. K. Jacob, Evaluation of eye gaze interaction, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 00, Association for Computing Machinery, New York, NY, USA, 2000, p. 281288. doi:10.1145/332040.332445.
URL https://doi.org/10.1145/332040.332445

[7] R. Bednarik, T. Gowases, M. Tukiainen, Gaze interaction enhances problem solving: Effects of dwell-time based, gaze-augmented, and mouse interaction on problem-solving strategies and user experience, Journal of Eye Movement Research 3 (1). doi:10.16910/jemr.3.1.3.
URL https://bop.unibe.ch/JEMR/article/view/2287

[8] R. Vertegaal, A fitts law comparison of eye tracking and manual input in the selection of visual targets, in: Proceedings of the 10th International Conference on Multimodal Interfaces, ICMI 08, Association for Computing Machinery, New York, NY, USA, 2008, p. 241248. doi:10.1145/1452392.1452443.
URL https://doi.org/10.1145/1452392.1452443

[9] K. Pfeuffer, L. Mecke, S. Delgado Rodriguez, M. Hassib, H. Maier, F. Alt, Empirical evaluation of gaze-enhanced menus in virtual reality, in: 26th ACM Symposium on Virtual Reality Software and Technology, 2020, pp. 1–11.

[10] O. Špakov, D. Miniotas, On-line adjustment of dwell time for target selection by gaze, in: Proceedings of the Third Nordic Conference on Human-Computer Interaction, NordiCHI 04, Association for Computing Machinery, New York, NY, USA, 2004, p. 203206. doi:10.1145/1028014.1028045.
URL https://doi.org/10.1145/1028014.1028045

[11] P. Majaranta, I. S. MacKenzie, A. Aula, K.-J. Räihä, Effects of feedback and dwell time on eye typing speed and accuracy, Universal Access in the Information Society 5 (2) (2006) 199–208.

[12] G. R. Barnes, Rapid learning of pursuit target motion trajectories revealed by responses to randomized transient sinusoids, Journal of Eye Movement Research 5 (3).
URL https://bop.unibe.ch/JEMR/article/view/2337

[13] M. Vidal, A. Bulling, H. Gellersen, Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets, in: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13, ACM, New York, NY, USA, 2013, pp. 439–448. doi:10.1145/2493432.2493477.
URL http://doi.acm.org/10.1145/2493432.2493477

[14] E. Velloso, M. Wirth, C. Weichel, A. Esteves, H. Gellersen, Ambigaze: Direct control of ambient devices by gaze, in: Proceedings of the 2016 ACM Conference on Designing Interactive Systems, DIS '16, ACM, New York, NY, USA, 2016, pp. 812–817. `doi:10.1145/2901790.2901867`.
URL `http://doi.acm.org/10.1145/2901790.2901867`

[15] S. Schenk, M. Dreiser, G. Rigoll, M. Dorr, Gazeeverywhere: Enabling gaze-only user interaction on an unmodified desktop pc in everyday scenarios, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, ACM, New York, NY, USA, 2017, pp. 3034–3044. `doi:10.1145/3025453.3025455`.
URL `http://doi.acm.org/10.1145/3025453.3025455`

[16] A. Esteves, E. Velloso, A. Bulling, H. Gellersen, Orbits: Gaze interaction for smart watches using smooth pursuit eye movements, in: Proceedings of the 28th Annual ACM Symposium on User Interface Software &#38; Technology, UIST '15, ACM, New York, NY, USA, 2015, pp. 457–466. `doi:10.1145/2807442.2807499`.
URL `http://doi.acm.org/10.1145/2807442.2807499`

[17] R. J. Jacob, Eye movement-based human-computer interaction techniques: Toward non-command interfaces, Advances in human-computer interaction 4 (1993) 151–190.

[18] H. Drewes, A. Schmidt, Interacting with the computer using gaze gestures, in: C. Baranauskas, P. Palanque, J. Abascal, S. D. J. Barbosa (Eds.), Human-Computer Interaction – INTERACT 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 475–488.

[19] M. Bâce, T. Leppänen, D. G. de Gomez, A. R. Gomez, ubigaze: Ubiquitous augmented reality messaging using gaze gestures, in: SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications, SA '16, ACM, New York, NY, USA, 2016, pp. 11:1–11:5. `doi:10.1145/2999508.2999530`.
URL `http://doi.acm.org/10.1145/2999508.2999530`

[20] A. Hyrskykari, H. Istance, S. Vickers, Gaze gestures or dwell-based interaction?, in: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12, ACM, New York, NY, USA, 2012, pp. 229–232. doi:10.1145/2168556.2168602.
URL http://doi.acm.org/10.1145/2168556.2168602

[21] H. Istance, A. Hyrskykari, L. Immonen, S. Mansikkamaa, S. Vickers, Designing gaze gestures for gaming: An investigation of performance, in: Proceedings of the 2010 Symposium on Eye-Tracking Research &#38; Applications, ETRA '10, ACM, New York, NY, USA, 2010, pp. 323–330. doi:10.1145/1743666.1743740.
URL http://doi.acm.org/10.1145/1743666.1743740

[22] W. Delamare, T. Han, P. Irani, Designing a gaze gesture guiding system, in: Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, 2017, pp. 1–13.

[23] V. Rajanna, T. Hammond, A gaze-assisted multimodal approach to rich and accessible human-computer interaction, CoRR abs/1803.04713. arXiv:1803.04713.
URL http://arxiv.org/abs/1803.04713

[24] Y. K. Meena, H. Cecotti, K. Wong-Lin, G. Prasad, A multimodal interface to resolve the midas-touch problem in gaze controlled wheelchair, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2017, pp. 905–908. doi:10.1109/EMBC.2017.8036971.

[25] N. Sidorakis, G. A. Koulieris, K. Mania, Binocular eye-tracking for the control of a 3d immersive multimedia user interface, in: 2015 IEEE 1st Workshop on Everyday Virtual Reality (WEVR), IEEE, 2015, pp. 15–18. doi:10.1109/WEVR.2015.7151689.

[26] P. Biswas, P. Langdon, Multimodal intelligent eye-gaze tracking system,

International Journal of Human-Computer Interaction 31 (4) (2015) 277–294.

[27] K. Pfeuffer, H. Gellersen, Gaze and touch interaction on tablets, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16, ACM, New York, NY, USA, 2016, pp. 301–311. doi:10.1145/2984511.2984514.
URL http://doi.acm.org/10.1145/2984511.2984514

[28] K. Pfeuffer, J. Alexander, H. Gellersen, Partially-indirect bimanual input with gaze, pen, and touch for pan, zoom, and ink interaction, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, ACM, New York, NY, USA, 2016, pp. 2845–2856. doi:10.1145/2858036.2858201.
URL http://doi.acm.org/10.1145/2858036.2858201

[29] S. Stellmach, R. Dachselt, Look & touch: gaze-supported target acquisition, in: Proceedings of the SIGCHI conference on human factors in computing systems, 2012, pp. 2981–2990.

[30] S. Stellmach, R. Dachselt, Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets, in: Proceedings of the sigchi conference on human factors in computing systems, 2013, pp. 285–294.

[31] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, M. Billinghurst, Pinpointing: Precise head-and eye-based target selection for augmented reality, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1–14.

[32] D. Rozado, J. Niu, M. Lochner, Fast human-computer interaction by combining gaze pointing and face gestures, ACM Transactions on Accessible Computing (TACCESS) 10 (3) (2017) 10.

[33] B. Martinez, M. F. Valstar, Advances, challenges, and opportunities in automatic facial expression recognition, in: Advances in face detection and facial image analysis, Springer, 2016, pp. 63–100.

[34] S. Mayer, G. Laput, C. Harrison, Enhancing mobile voice assistants with worldgaze, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 110. doi:10.1145/3313831.3376479.
URL https://doi.org/10.1145/3313831.3376479

[35] T. R. Beelders, P. J. Blignaut, The usability of speech and eye gaze as a multimodal interface for a word processor, Speech Technologies (2011) 386–404.

[36] C. Acartürk, J. Freitas, M. Fal, M. S. Dias, Elderly speech-gaze interaction, in: M. Antona, C. Stephanidis (Eds.), Universal Access in Human-Computer Interaction. Access to Today's Technologies, Springer International Publishing, Cham, 2015, pp. 3–12.

[37] A. Esteves, Y. Shin, I. Oakley, Comparing selection mechanisms for gaze input techniques in head-mounted displays, International Journal of Human-Computer Studies 139 (2020) 102414. doi:https://doi.org/10.1016/j.ijhcs.2020.102414.
URL https://www.sciencedirect.com/science/article/pii/S1071581920300185

[38] D. Miniotas, O. Špakov, I. Tugoy, I. S. MacKenzie, Speech-augmented eye gaze interaction with small closely spaced targets, in: Proceedings of the 2006 symposium on Eye tracking research & applications, 2006, pp. 67–72.

[39] T. R. Beelders, P. J. Blignaut, Using eye gaze and speech to simulate a pointing device, in: Proceedings of the Symposium on Eye Tracking Research and Applications, 2012, pp. 349–352.

[40] K. Sengupta, M. Ke, R. Menges, C. Kumar, S. Staab, Hands-free web browsing: enriching the user experience with gaze and voice modality, in: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, 2018, pp. 1–3.

[41] D. G. Zhao, N. D. Karikov, E. V. Melnichuk, B. M. Velichkovsky, S. L. Shishkin, Voice as a mouse click: Usability and effectiveness of simplified hands-free gaze-voice selection, Applied Sciences 10 (24) (2020) 8791.

[42] I. Chatterjee, R. Xiao, C. Harrison, Gaze+ gesture: Expressive, precise and targeted free-space interactions, in: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, 2015, pp. 131–138.

[43] K. Pfeuffer, B. Mayer, D. Mardanbegi, H. Gellersen, Gaze+ pinch interaction in virtual reality, in: Proceedings of the 5th Symposium on Spatial User Interaction, 2017, pp. 99–108.

[44] J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, P. Irani, Consumed endurance: a metric to quantify arm fatigue of mid-air interactions, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2014, pp. 1063–1072.

[45] J. Hild, P. Petersen, J. Beyerer, Moving target acquisition by gaze pointing and button press using hand or foot, in: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, 2016, pp. 257–260.

[46] M. Kumar, A. Paepcke, T. Winograd, Eyepoint: practical pointing and selection using gaze and keyboard, in: Proceedings of the SIGCHI conference on Human factors in computing systems, 2007, pp. 421–430.

[47] T. E. of Encyclopaedia Britannica, Morse code, [Online; accessed September 14, 2018] (March 2018).
URL https://www.britannica.com/topic/Morse-Code

[48] S. N. Patel, G. D. Abowd, Blui: Low-cost localized blowable user interfaces, in: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, UIST '07, ACM, New York, NY, USA, 2007, pp. 217–220. doi:10.1145/1294211.1294250.
URL http://doi.acm.org/10.1145/1294211.1294250

[49] A. K. Dey, R. Hamid, C. Beckmann, I. Li, D. Hsu, A cappella: Programming by demonstration of context-aware applications, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, ACM, New York, NY, USA, 2004, pp. 33–40. doi:10.1145/985692.985697.
URL http://doi.acm.org/10.1145/985692.985697

[50] B. Hartmann, L. Abdulla, M. Mittal, S. R. Klemmer, Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07, ACM, New York, NY, USA, 2007, pp. 145–154. doi:10.1145/1240624.1240646.
URL http://doi.acm.org/10.1145/1240624.1240646

[51] Oracle, Audioformat (java platform se 7), [Online; accessed March 30, 2021] (June 2020).
URL https://docs.oracle.com/javase/7/docs/api/javax/sound/sampled/AudioFormat.html

[52] N. H. P. R. Group, Nasa task load index (tlx) paper and pencil package (1986).
URL https://humansystems.arc.nasa.gov/groups/TLX/downloads/TLX.pdf

[53] JASP Team, JASP (Version 0.12.2)[Computer software], accessed on 16.Oct.2020 (2020).
URL https://jasp-stats.org/

[54] O. Špakov, D. Miniotas, On-line adjustment of dwell time for target selection by gaze, in: Proceedings of the third Nordic conference on Human-computer interaction, ACM, 2004, pp. 203–206.

[55] I. S. MacKenzie, Evaluating eye tracking systems for computer input, in: Gaze interaction and applications of eye tracking: Advances in assistive technologies, IGI Global, 2012, pp. 205–225.

[56] A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, M. R. Morris, Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design, in: Proceedings of the 2017 Chi conference on human factors in computing systems, ACM, 2017, pp. 1118–1130.

[57] R. Bellman, R. Kalaba, On adaptive control processes, IRE Transactions on Automatic Control 4 (2) (1959) 1–9. doi:10.1109/TAC.1959.1104847.

[58] C. Myers, L. Rabiner, A. Rosenberg, Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 28 (6) (1980) 623–635. doi:10.1109/TASSP.1980.1163491.

[59] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1) (1978) 43–49. doi:10.1109/TASSP.1978.1163055.

[60] pierre rouanet, pierre-rouanet/dtw, accessed on 16.Oct.2020 (2020).
URL https://github.com/pierre-rouanet/dtw

[61] P. M. Fitts, The information capacity of the human motor system in controlling the amplitude of movement., Journal of experimental psychology 47 (6) (1954) 381.

[62] J. O. Wobbrock, K. Shinohara, A. Jansen, The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on

pointing measures and models, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2011, pp. 1639–1648.

[63] S. G. Hart, Nasa-task load index (nasa-tlx); 20 years later, Proceedings of the Human Factors and Ergonomics Society Annual Meeting 50 (9) (2006) 904–908. arXiv:https://doi.org/10.1177/154193120605000909, doi:10.1177/154193120605000909.
URL https://doi.org/10.1177/154193120605000909