EyeShadows: Peripheral Virtual Copies for Rapid Gaze Selection and Interaction

Jason Orlosky* Augusta University
Chang Liu† Kyoto University
Kenya Sakamoto‡ Osaka University
Ludwig Sidenmark§ University of Toronto
Adam Mansour¶ Augusta University

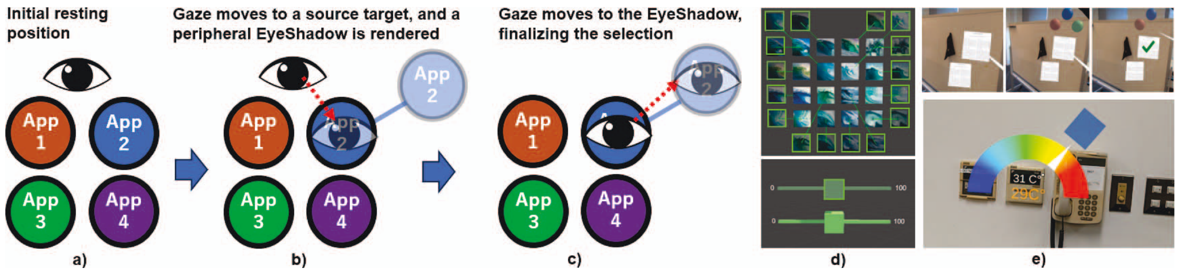

Figure 1: Basic concept diagrams of the EyeShadows selection system including a) eye gaze position and selection targets at rest, b) gaze upon a target, which renders a secondary peripheral copy, i.e., an *EyeShadow*, c) subsequent movement of the eye to the EyeShadow, confirming the user's intended selection. The remaining two figures show d) layouts of EyeShadows in VR for search and select (top) and an analog slider (bottom), with green outlines highlighting the EyeShadow element positions and e) an implementation of EyeShadows for document selection in the Varjo XR-3 (top) and in-situ temperature control in the HoloLens 2 (bottom).

Abstract

In eye-tracked augmented and virtual reality (AR/VR), instantaneous and accurate hands-free selection of virtual elements is still a significant challenge. Though other methods that involve gaze-coupled head movements or hovering can improve selection times in comparison to methods like gaze-dwell, they are either not instantaneous or have difficulty ensuring that the user's selection is deliberate. In this paper, we present *EyeShadows*, an eye gaze-based selection system that takes advantage of peripheral copies (shadows) of items that allow for quick selection and manipulation of an object or corresponding menus. This method is compatible with a variety of different selection tasks and controllable items, avoids the Midas touch problem, does not clutter the virtual environment, and is context sensitive. We have implemented and refined this selection tool for VR and AR, including testing with optical and video see-through (OST/VST) displays. Moreover, we demonstrate that this method can be used for a wide range of AR and VR applications, including manipulation of sliders or analog elements. We test its performance in VR against three other selection techniques, including dwell (baseline), an inertial reticle, and head-coupled selection. Results showed that selection with EyeShadows was significantly faster than dwell (baseline), outperforming in the select and search and select tasks by 29.8% and 15.7%, respectively, though error rates varied between tasks.

Keywords: Eye Tracking, Virtual Reality, Augmented Reality, Selection, Hands-free, Manipulation

*e-mail: jorlosky@augusta.edu
†e-mail: liuchang@kuhp.kyoto-u.ac.jp
‡e-mail: sakamoto.kenya@lab.ime.cmc.osaka-u.ac.jp
§e-mail:lsidenmark@dgp.toronto.edu
¶e-mail: duskkazuno@gmail.com

1 Introduction

In augmented and virtual reality (AR/VR), hands-free selection and manipulation have long been important goals for assistive and multimodal interfaces. With recent advances in integrated eye tracking, research has produced a number of improved methods for making selections with the assistance of controllers, head gaze, and other secondary confirmations. These techniques are incredibly useful for improving the speed and accuracy of selection, but they often require the use of an individual's hands or other limbs or some other clutch-based mechanism for confirming the selection target. Methods that utilize head tracking or dwell-based selection may require additional time for the head or limbs to catch up to the gaze target, which still inhibits selection performance in many cases.

In this paper, we present EyeShadows, a pure eye-gaze method for fast selection in virtual and augmented environments. This strategy makes use of a secondary selection using a peripheral virtual copy of the target, in which the user looks at the copy to finalize a selection, as shown in Figure 1. Though secondary selection has been proposed for other applications, such as reverse-crossing for gaze typing [3] and the dual-gaze system [12], neither of these methods extend well to other use cases such as analog control, nor do they provide contextual information during the selection process. Moreover, they are not always evaluated against more recent state-of-the art selection methods such as gaze-activated head-crossing [23] for VR tasks.

The key differences of EyeShadows in comparison to other work is that it 1) provides context-specific virtual copies of target content, 2) places those copies in peripheral locations to avoid the Midas touch problem, and 3) includes view management principles to facilitate easy selection for a variety of tasks. In terms of selection speed, another reason that this works well is that the speed of fine grained selections is limited only by a secondary saccade, which occurs on the order of 200 milliseconds (ms) or less.

Although systems like dual-gaze [12] and the inertial reticle method called "Duo-Reticles" [19] also try to take advantage of the speed of saccades for this purpose, they still do not significantly outperform conventional dwell-based selection. One reason why these techniques might not have outperformed dwell may be that both dual-gaze and Duo-Reticles place the secondary selection mechanism near the central field of view, which can distract the user or

979-8-3503-7402-5/24/$31.00 ©2024 IEEE
DOI 10.1109/VR58804.2024.00088

occlude content. A second reason may be that the type of reticles used do not provide contextual information about the gaze target. Other methods such as reverse-crossing and head-gaze crossing require multiple eye movements, countdown timers, or head-based activation, adding to selection time.

EyeShadows addresses these problems since the copy of the virtual target contains information about the target itself and the targets are in positions that are easy to select, even if eye tracking accuracy is poor. In cases where multiple targets are nearby or could potentially clutter the environment, leader lines connect the targets and their respective EyeShadows to improve ease of selection. We also refined parameters such as the distance to the virtual copy, display time, and fade-out to improve selection times and reduce clutter.

Once we developed and refined the EyeShadows system, we set up an experiment to test its performance against dwell-based selection (baseline), head-crossing, and Duo-Reticles. Results from 17 participants showed that EyeShadows significantly outperformed Dwell (baseline) selection for basic selection and search and select tasks by margins of 29.8% and 15.7%, respectively. Eyeshadows also outperformed head-crossing and Duo-Reticles in some other respects, though error was higher in some cases. These results suggest that EyeShadows is promising for use as a selection tool in virtual and augmented reality.

In addition, we implemented mixed reality (MR) versions of EyeShadows to test its use in practice for OST and VST displays. These were tested with the Hololens 2 and the Varjo XR-3, and we include details of these tests to show how functionality works in these devices. To summarize, the contributions of this work include:

- the development of EyeShadows, a method for using digital copies of virtual elements to enable fast, deliberate control via eye gaze,
- adaptation of EyeShadows to various use cases, including selection, search, and analog control,
- experiments testing the performance of EyeShadows against a baseline and two other recent selection methods,
- practical testing of EyeShadows in VR and both OST- and VST-AR to demonstrate its usability in multiple environments and on different sets of hardware.

2 Prior Work

Related research can generally be divided into three areas, including gaze selection techniques that make use of an external clutch, time-based method that use a waiting period, and methods that are specific to Augmented and Virtual Reality. In addition to discussing these three areas in the following sections, we have summarized the most closely related research in Table 1, including the reference and characteristics specific to each method.

2.1 Gaze Selection using an External Clutch

Since gaze lacks an intrinsic confirmation mechanism (i.e., a "click") and is always on, it is commonly combined with other modalities that act as a clutch to trigger interaction. These techniques are motivated to lower the required body movements by relying more on gaze [33], or to allow more granular and efficient interaction with gaze [9]. As such, gaze has been combined with multiple modalities such as hands [7, 9, 14, 21], head [9, 23, 24], and voice [30] in contexts such as menus [25], travel [18], object manipulation [18], and pointing refinement [9]. Conversely, some methods use gaze as the clutch to improve selection times via mouse, such as the combination of gaze with Ninja Cursors [20].

This prior work show how combining gaze with other modalities to act as a clutch can improve the affordances of gaze-based interaction. However, multimodal techniques can be disadvantaged when modalities are occupied (i.e., the user already interacts with their hands). Especially in the case of the eyes, such as for blink interaction, the eyelids will often occlude a target and disrupt the user's vision [2, 16]. Furthermore, the user has fewer interaction techniques at their disposal at any time, as a single technique occupies multiple modalities, limiting their expressiveness and efficiency. These situations create value for single-modality techniques where the interaction technique uses only a single modality. Our work therefore focuses on gaze-only interaction, and we show its capabilities for fast, accurate selection with during continuous interaction.

2.2 Time-based Selection

Time based selection methods generally consist of a wait period in which the user must keep his or her eyes or head steady while a timer counts down. Perhaps the most well-known of these methods is "dwell," which has a short learning curve, is relatively efficient, and has a low error rate for selection tasks in which detailed inspection is not necessary [6]. Traditional dwell techniques have been improved over the years using automated adjustment, such as the work by Vspakov et al. [26]. Other more recent techniques for virtual manipulation make use of a series of dwells for both selection and manipulation in combination with gaze [11].

In addition to dwell, some other methods make use of a similar time based principle that can also alleviate the Midas touch problem. For example, the Pursuits mechanism by Vidal et al. allows a user to follow a moving target in a unique shape that is attached to a particular target [22, 31]. Though following the target takes time, the unique target shape prevents accidental selection and can be in close proximity to the selection target. One disadvantage of many time-based techniques is that they use a timer of 500 ms to 2 seconds to initiate a selection, which prevents detailed inspection of the object for greater than the timer. EyeShadows mitigates this problem since the trigger mechanism is separated from the gaze target, allowing for indefinite inspection time and low risk of accidental selection.

2.3 Augmented and Virtual Reality Specific Methods

In addition to much of the work that has been done on gaze selection alone, much work has been dedicated to solving selection tasks in AR and VR, where virtual feedback and spatial logic can assist with the selection process. For example, in the work by Piumsomboon et al., the authors proposed several selection methods specific to VR that could alleviate the midas touch problem and also handle occluded objects [19]. Fernandes, et al. also recently tested eye tracking against head- and controller-based selection in Virtual Reality [4], finding that eye tracking outperformed head selection and matched controllers in targeting and button-press selections.

Other work has tested various methods such as dwell and goal-crossing to analyze saccade performance: [13]. Other gaze-based work in VR have made use of techniques such as adjusting the environment via object mirrors [10] or by adjusting targets' positions for easier selection [1]. The closest work to ours is likely DualGaze [12], which renders a small confirmation square immediately adjacent to the corresponding virtual element for selection. Similarly, a method called reverse-crossing renders a small icon next to keys on a keyboard for selection, though this is only implemented and tested for PC-based typing, and the icons are often in-between the other keys. [8]. Some other work goes so far as to clone a monitor itself to improve interaction via gaze, but the purpose of that method is not for fine-grained selection [27].

EyeShadows have several important improvements that differentiate them from DualGaze and Reverse-Crossing. First, we can render EyeShadows not just for selection, but for analog controls such as sliders or dials. Moreover, rendering is possible in cramped or crowded spaces since we include integrated view management, and we have also tested with multiple challenging use case scenarios in both AR and VR. Lastly, the virtual content contained in each EyeShadow also contains information about the selection target, allowing users to take advantage of their peripheral view to assist with selection decisions.

Table 1: A review of gaze-based control methods for XR with a summary of the features of each, including the use of the eyes, the head, an external clutch, wait time, and estimated robustness to accuracy and calibration error.

Method [citation]	Eye Used	Head Used	Clutch Required	Wait/Dwell Required	Error-Robust
OrthoGaze [11]	yes	no	no	yes	mid
Look&Cross [23]	yes	yes	yes	no	high
Duo-Reticle [19]	yes	no	no	no	mid
Pursuits [31]	yes	no	no	yes	high
Blink [2, 16]	yes	no	no	no	mid
Head-coupled [11]	yes	yes	yes	no	mid
Bimodal [24]	yes	yes	yes	yes	mid
Radii [25]	yes	yes	yes	no	mid
Pinpointing [9]	yes	yes	yes	yes	mid
Dual-gaze [12]	yes	yes	yes	no	low
EyePointing [21]	no	yes	yes	no	mid
EyeShadows (ours)	yes	no	no	no	high

2.4 Further Motivation

Of the prior work mentioned previously, the closest work to ours is likely DualGaze, the system by Mohan et al. that places a small, square icon next to the gaze target [12]. While this method employs a secondary selection, the secondary item is small and can suffer from calibration error, and it is not adapted for other use cases such as analog control or manipulation. Moreover, experiments testing the DualGaze mechanism were limited to a single selection task and compared only to a dwell-based mechanism (referred to "fixed gaze" in that paper) [12]. Though our work is similar, we have made several important improvements, including the placement of shadows in the periphery. Whereas central vision roughly takes up the 1.5 - 2 degrees in the center of the human FoV, shadows range from 5 to 30 degrees in placement, as outlined in Figure 2. In addition, we include context-coding of shadows with numerical, image, or scalar information, and a more thorough experiment testing two other selection techniques as well as two other tasks.

Moreover, we implemented EyeShadows for both the Hololens 2 and the Varjo XR-3, demonstrating its practical use in addition to formal evaluation in a VR environment.

3 System Design

The EyeShadows selection system is based on the combination of a primary coarse selection and a secondary peripheral confirmation that takes advantage of the fact that users can perceive peripheral content without directly attending to it. The basic interaction mechanism and samples of use are shown in Figures 1, 4, and 5.

This approach is advantages for two main reasons.

1. Because each EyeShadow is present in the periphery, the user can both focus on items in central vision while determining the position of the relevant selector (EyeShadow).

2. Copies of the targets for selection contain visual information related to the target. I.e. decisions about a selection can be made using only color or shape information from the EyeShadow without gazing at it directly.

For an objective comparison with EyeShadows, we have also implemented three other methods, including dwell (baseline), Look&Cross, and Duo-Reticles, which served as a diverse comparison for performance. The details of the EyeShadows interaction system as well as the implementation of the three other methods are described below.

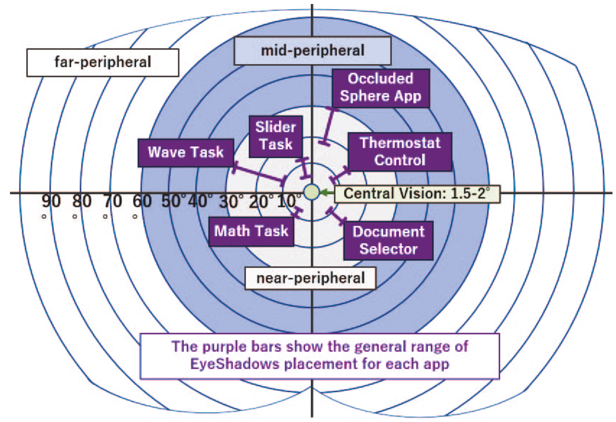

Figure 2: Diagram showing the human field of view (FoV) and the general area of placement for peripheral shadows in the EyeShadows algorithm, which vary from 5 to 30 degrees in our implementation. Each bar represents the range of degrees within the periphery in which a shadow may be placed for the corresponding application.

3.1 EyeShadows Framework

In order to implement the EyeShadows concept, we had to overcome a number of technical challenges associated with the timing of gaze and saccade movements and refine the rendering techniques needed to display secondary selection objects (also referred to as *"shadows"* throughout this paper). The first step in this process was to determine when and where to render a Shadow.

To answer the "when" question, we determined that the fastest way to present information would be to render the secondary confirmation shadow instantaneously, but as a context-dependent activation. In other words, when a user looks at a potential gaze target, the shadow for that target, and only that target, is rendered immediately.

Answering the "where" question was more difficult. In or review of prior work, we found that a majority of methods try to render virtual content within the central field of view. Though this places selectors nearby for easy access, it can distract the user or incur the Midas touch problem. It occurred to us that the periphery, i.e. greater than 2 degress from the center of vision, though farther away from the gaze target, would be out of the way and less distracting, yet could be reached by a single saccade if the target was large enough. As such, we tested shadows of multiple sizes in the periphery, and found that as long as the shadow was of adequate size, the secondary selection could be made with a single saccade rather than a primary and corrective saccade; Most fine-gaze control tasks require two or more saccades to accurately align gaze direction with a small target.

Testing this concept produced results that appeared to be much quicker than the traditional dwell-based mechanism, so we set out to improve and refine the interaction mechanism for other use cases.

3.2 General Interaction Mechanism

The principal behind EyeShadows is that any object that is selectable or manipulable in a virtual environment receives a secondary copy of itself that functions as a secondary confirmation when gazed upon. Examples are shown in figures 1, 3, 4, and 5. The location of the *shadow* is placed in the periphery just far enough that it avoids common Midas-touch problems, but close enough that a user can reach the shadow with a single saccade of the eye. The approximate ranges of angular positions of shadows, defined as the angle between the source icon and corresponding shadow relative to the HMD's origin, are as follows: Math task: 8 to 10 degrees, Wave task: 10 to 28 degrees, Slider Task: 5 - 12 degrees, Occluded sphere application: 17 to 30 degrees, and Thermostat/Document applications: 8 - 15

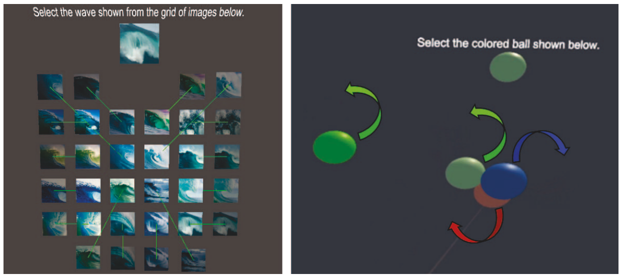

Figure 3: The left image shows a screenshot of the wave task from the experiment with all shadows and leader lines rendered for reference. *Note that during use, both the shadow (copy) and leader line are only activated when each individual gaze target is viewed. The right image shows an additional task used for practical testing involving moving/occluded spheres and the shadow corresponding to the green sphere, rendered in the left periphery. Arrows showing example travel directions are overlaid for reference in this image, but are not rendered during use.

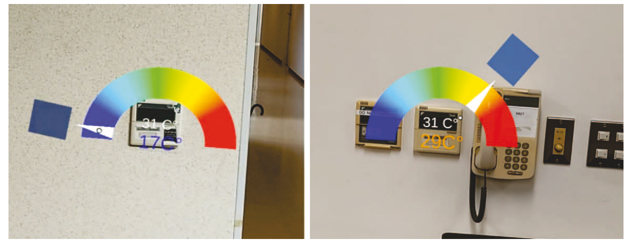

Figure 4: Images of EyeShadows on the Hololens 2, including gaze-controlled thermostats for temperature adjustment. The user can adjust the temperature by gazeing at the blue square on the border of the widget and finalize the selection by looking at the center temperature reading.

degrees, as detailed in Figure 2. An example of the peripheral orientation of the shadows for a search-and-select task is shown in D) of Figure 1. Another benefit of this placement is that a saccade to the peripheral location of shadows requires minimal additional effort on the part of the user.

In addition to basic selections, shadows are effective at controlling analog elements such as knobs, sliders, or other linear controls. Similar to selection, a secondary copy of the slider or knob to be controlled is rendered near the target control, as can be seen in D) of Figure 1 and in Figure 4. The secondary copy of the controllable element renders upon gaze of the primary target, is set upon gaze of the shadow, and is finalized upon return to the target. This allows the user to examine the target for as long as he or she wishes, make a selection via saccade, and return to the target to verify the accuracy of the selection. In this scheme, color coding can be used in place of an exact copy of the target for easy-to-remember shadows (icons).

3.2.1 Gaze-contingent Render and Fade Parameters

By design, the EyeShadows system places additional virtual components into the scene, which runs the risk of cluttering the user's view or obstructing content. As such we needed a way to ensure that shadows were visible when there was a possibility that the target needed to be selected, but not rendered when a target is not intended for selection. To do so, the rendering of all shadows are gaze contingent. In other words, shadows will only render when a target has been viewed. In addition to keeping the environment clutter free, this also has the benefit of helping the user associate a target with a given shadow since its virtual copy renders at the moment of gaze. Though a few frames of latency exist in the update cycle of the eye tracker, users can easily make this association despite the shadow rendering slightly after their gaze intersects the target.

Secondly, we needed a way to turn off the renderers for each shadow to prevent additional clutter. Initially, we simply attached a one-second timer to each shadow and turned off the renderer after one second after it appeared. However, this resulted in significant distraction during testing since the instantaneous disappearance of the shadow created a high-contrast effect that often triggered the user's esotaric motion perception. To mitigate this effect, we added a fade-out effect that decreased the opacity of the shadow from one to zero over the one-second rendering. This smooth transition was less distracting but still preserved contextual information. For shadows that control analog effects such as the slider task, knob or dial (see d) and e) of Figure 1, the fade-out effect is postponed until the user's gaze has left the corresponding shadow.

3.2.2 Scan-Path Filter

During initial testing, we found that every so often, a user's gaze would pass over a shadow and trigger it accidentally. This mostly happened during the wave task when gaze started at the target wave and passed over the leftmost or rightmost target. Similarly, in practical application or layered applications such as that of Huynh et al. [5], users will be glancing between different contexts, each with their own set of gaze targets, and will not always intend to make a selection between the application switch.

To address this problem, each shadow has an internal frame counter that requires at least five frames (roughly 100ms) of gaze intersection to trigger, which we logically selected through repeatedly testing each application. Consequently, any saccades that happen to pass over a shadow will only do so for one or two frames, which will avoid the counter trigger for that shadow and prevent unintentional selection. The trigger resets when no gaze intersection is present in a subsequent frame. Note that while this may seem similar to dwell interactions, after the has completed a first coarse saccade during a purposeful glance, the user needs at least 100 milliseconds to start a second corrective saccade. In other words, no matter where a user's gaze ends up after a saccade, as long as it is intersecting a portion of the shadow, the interaction will incur no additional time overhead.

3.2.3 Shadow View Management

Similar to the Dualgaze [12], Duo Reticles [19], and Radii [25] techniques, EyeShadows makes use of view management principles to ensure easy selection and avoid clutter. Placement of a selection element in between the answers or waves seen in the left two images Figure 7 would very likely result in the user's gaze passing through an unintended selection element. The peripheral positions of EyeShadows alleviate this problem, but in situations where two target elements are separated such that neither can fit into the central or near-peripheral field of view simultaneously, even a peripheral icon might be placed between the two elements.

As such, we have coupled eye shadows with several layouts that reduce the chances of icons that may tend to overlap, as shown in Figure 3. For example, in the wave task, icons are spaced relatively evenly around the search grid, but those that would have bisected the position between the target and start of the grid have been moved to the side, similar to the strategies presented in Hedgehog labels [28] and Halo Content [15]. The size and placement of each shadow for these applications was designed based on several factors, including the size of the source target, the expected distance of interaction for the context of the task, and information density. Moreover, leader lines are included in cases when the group of targets may result in visual confusion. The right-hand image of Figure 3 shows a set of three spheres that are constantly moving. For this type of dynamic task, the shadow corresponding to each sphere moves with the sphere itself off to the side.

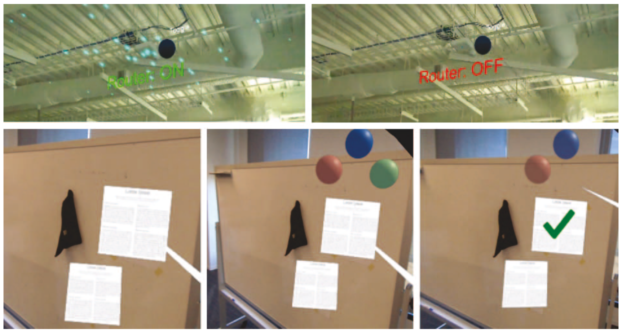

Figure 5: This figure shows images of EyeShadows on the Varjo XR-3 in video see-through (VST) mode. The top row includes a toggle switch for a ceiling-mounted router. To engage the toggle, users look at the circular icon on the right of the router and then at the text to confirm the selection. The bottom row shows a whiteboard document selection tool, in which users can confirm documents by selecting one of the three colored icons on top of each page.

3.3 Multiple device implementations

To test whether the EyeShadows mechanism worked in practice and to determine what refinements might need to be made for other displays, we developed several additional applications. These included an analog thermostat control designed for the Hololens 2 and a router toggle system and whiteboarding application for the Varjo XR-3.

3.3.1 HoloLens 2 Implementation

We also implemented EyeShadows in an Optical See-through (OST) display, the HoloLens 2. In both AR and MR, EyeShadows will appear visually similar. However, several adjustments were necessary, for example to compensate for the HoloLens 2's slower eye tracking in comparison to the Vive Pro Eye. While Hololens 2 might not offer as rapid selection, it allows for the display position of the *shadow* to be registered in a local environment, enabling a more context-relevant user experience. Figure 4 shows our implementation for adjusting air conditioner temperature, which was adapted from the analog slider mechanism in the VR-based experiment.

In this system, the temperature is visualized with a colored semicircle that uses the blue-green-red rainbow scheme that is traditionally used for temperature information. When the user gazes at the outer ring of the semicircle, a conical bar with a square (the shadow) at its endpoint rotates with the user's gaze. Much like the VR implementation, the adjustment is disabled when the user looks back at the temperature to confirm a selection, which avoids accidental selection. The current room temperature is displayed in the center, along with the newly set temperature corresponding to the bar's position.

3.3.2 XR-3 Implementation

For our implementation on the Varjo XR-3, we made use of the mixed reality capabilities of the HMD. The Varjo API provides eye tracking data that is calibrated to the video pass-through, and the gaze ray is passed into the EyeShadows implementation with relatively few changes. We also used the Varjo Marker system and inside-out tracking to affix the targets to real-world objects in the mixed reality view. Two demos show examples of how a user may interact with their environment via gaze, including a router which can be turned on/off and a whiteboard application in which documents can be confirmed or cancelled (shown in Figure 5).

3.4 Other methods for comparison

To ensure a robust comparison of EyeShadows with other methods, we implemented three other gaze-based selection techniques and adapted each one to our experimental tasks. The descriptions and implementations of each of these methods are as follows.

Figure 6: Images of the three other methods that we implemented for comparison, shown relative to the math task. These include Look&Cross (left [23]), Gaze-dwell (middle [11]), and Duo-Reticles (right [19]).

3.4.1 Look&Cross

The "Look&Cross" technique is based on the selection method proposed by Sidenmark et al. [23, 25], as shown on the left of Figure 6. Look & Cross is designed as a generic interaction technique based on the natural misalignment between the eye and head directions, and that gaze naturally precedes the head during gaze shifts [23]. The gaze is used to explore the interface, trigger hover interaction, and enable widgets for selection. The user then selects a widget by moving a head-attached cursor across the boundary of the gaze-activated widget. The technique can also be used for continuous interaction (i.e., slider interaction) by moving the position of the head within the boundary of the object after crossing.

3.4.2 Dwell (Baseline)

Dwell is an implementation of a standard time-based selection using a fixation point, as shown in the middle of Figure 6. Dwell-based selection is a very commonly used technique that performs well, so we considered this to be the the baseline for our experiment. When the user stares directly at an object, a small timer begins to increment, and a circular reticle starts to turn in a clockwise direction. Once the reticle forms a full circle, the dwell executes. This threshold for gaze dwell was set to 900 milliseconds considering the item sizes and locations used in our experiment [17]. While we considered adding DualGaze [12] to the conditions, we thought this would be unnecessary since the original paper found that DualGaze performance was equivalent to Dwell for average time taken, and therefore we included Dwell as our baseline.

3.4.3 Duo-Reticles

The Duo-Reticles method is based off of the work by Piumsomboon et al. [19], as shown on the right of Figure 6. To confirm a selection, users are required to use their eye gaze to align two reticles. The first reticle, the eye gaze reticle, appears at the user's current eye gaze location in real-time. The second reticle, the inertial reticle, indicates a moving-average gaze location from the near past. Over time, positions of the inertial reticle are updated and move toward the eye reticle, but stops short at a minimal Euclidean distance threshold [19]. In short, the Duo-Reticles takes advantages of eye saccades and allows users to make a confirmation by looking back at a certain position of their past eye gaze without requirement of gaze dwell.

4 Experiments

To test EyeShadows, we set up an experiment in which we tested various selections and manipulations against three other well recognized selection methods. The experiment consisted of three tasks, including basic answer selection for a mathematics task, a search and select task where participants selected from a grid of visually similar wave images, and an analog (slider) manipulation task in

Figure 7: Images of the three tasks used in the experiment, including selecting the correct answers for math problems (left), searching for and selecting a wave image from a randomly arranged set of other waves (center), and manipulating a slider to a number from zero to nine (right).

which participants had to set a slider to a given number. The tasks we implemented draw from previous work in eye-tracking selection, ranging from search [23], to selection [19] to moving a target laterally [11]. These tasks also provided a wide range of challenges that were designed to resemble real world tasks such as selection of an icon or setting an analog dial to a randomly generated value.

We recruited 17 participants, with an average age of 30.12 (stdev 9.58, 4 Female and 13 Male). This experiment was approved by a university Institutional Review Board (IRB). The tasks, methods for selection, and experiment conditions are described in detail below.

4.1 Tasks and Conditions

Each task represents a different operation modality, which consequently represents a different difficulty. These tasks included:

Multiple choice selection (easy) - This is a basic math problem in which users must select the correct answer from a list of six choices. Out of the selections, there is always only one correct answer. The orientation of answers in this task were placed in a semi-circular orientation around the problem itself, which is common for Fitt's law studies [29].

Search, compare, and select (intermediate) - This problem represents the more difficult task of searching for and selecting a target image from a randomized board of images, as used in [19] and [23].

Selection on analog scales (intermediate) - The purpose of this task is to move a slider to one of ten positions on an analog scale. Though the success condition for the task is to move the indicator into the correct slot, the slider is actually an analog scale with float values that are evaluated to three significant digits. This allowed us to measure fine-grained accuracy in addition to success rate, i.e. how far the final selection is from perfect (accuracy) and how much time it took for a selection to fall within the target range (selection time).

Each of these tasks was completed with four conditions, where one condition was one gaze selection method (either Dwell, DuoReticles, Look&Cross or EyeShadows). The order of the conditions was selected from a shuffled list prior to starting the experiment such that no order was repeated for any of the tasks.

4.2 Procedures

Prior to the experiment, participants read about the general content of the experiment and filled out a consent form. The experimenters then explained how to wear the HMD and adjust the bands for a comfortable fit. Next, the participant underwent the standard eye tracking calibration procedure that is part of the HTC Vive SteamVR package. We asked participants to gaze at several targets during this process to verify a successful calibration.

The participant then started the main experiment, which began with the math task. Since we did not set out to measure the difficulty of the tasks themselves, all participants completed the math, wave, and slider tasks in that order. Participants were simply instructed to select the correct answers, with no emphasis on speed or accuracy.

Each of the four selection methods were then given to the participant in shuffled order. Before starting a trial for a given method, participants went through a one minute tutorial on how to make selections with that particular gaze selection method (condition). Other than having a different duration, the tutorial task was exactly the same as the corresponding task in which we recorded data. Participants were free to ask questions during practice and were also monitored by an experimenter to ensure they understood how to use the method. If the participant seemed to be confused about a method's operation, for example if they did not understand that the head reticle corresponded to head movements, the experimenter provided a brief explanation as needed.

Once the minute of tutorial and practice was complete, participants then carried out as many selections as possible within a two-minute time period for each condition. Data was recorded for these two-minute sessions, and the experimenter refrained from providing any feedback unless the participant asked for help. Ten-second breaks were provided between each selection method, between the one-minute practices, and the two-minute tasks to ensure participants had time to rest their eyes prior to starting the next trial or task. Conditions were randomly selected from a Latin-square generated list, so no condition order was repeated for any of the four tasks for an individual participant. If learning occurred between tasks, this would still have little effect on each condition.

Once all four conditions were completed for all three tasks, the participant removed the display and answered a brief Likert scale questionnaire with three questions on the ease of use, eye fatigue, and efficiency of each selection method across all tasks. Questionnaires were administered after the entire experiment to prevent the need for recalibration. All participants completed the entire experiment within one hour.

4.2.1 Metrics for evaluation

The most important metric between the selection methods was the average time necessary to make a selection. To measure this, we recorded the total number of selections made during each two-minute trial and divided the time over the number of selections, though we did not remove individual outliers for trials. Note that we divided by the total number of selections rather than the number of correct selections since it was not possible to separate general user error, i.e. not recognizing the correct wave or making arithmetic error, from other errors resulting from unintentional selection. Though the simple nature of the tasks might make it seem that most mistakes were interactive, participants often vocalized logical errors during the experiment, e.g. missing a math problem or matching the wrong picture. Selection times were measured this way for all tasks, though the slider task technically did not have correct answers since the task was to continue manipulating the slider until the selection matched the number displayed.

In addition to selection times, we measured correct answer rates, defined as the percentage of correct answers out of the total number of selections made. These were recorded only for the first two tasks since the mathematics and wave selection tasks were the only ones that had definitively correct or incorrect answers. For the Slider task, we measured accuracy in terms of the absolute distance of the slider's final position to the center of the target number. This was done to highlight how well each respective method could be used to set a specific value. Note that participants were not told that they would be judged for accuracy on the slider task.

4.3 Results

The most relevant result from these experiments was that EyeShadows outperformed Dwell (baseline) for the math and wave selection tasks by the most significant margins, 29.8% and 15.7%, respectively, though error rates were higher in some cases.

Table 2: An overview of performance results, highlighting the best performing methods (highlighted in green) in terms of selection times and accuracy (or virtual distance to target in the case of the slider task) according to task. Significance values are represented by * for p < 0.05, ** for p < 0.01, and *** for p < 0.001, which are included if that method statistically outperformed at least one of the others.

Method	Task	Time (ms)	Accuracy
Duo-Reticles [19]	Math (select)	4513	100%***
Dwell	Math (select)	3194	99.52%
Look&Cross [23]	Math (select)	2742	97.06%
EyeShadows (Ours)	Math (select)	2477***	96.77%
Duo-Reticles [19]	Wave (search)	6581	99.12%***
Dwell	Wave (search)	4796	97.48%
Look&Cross [23]	Wave (search)	5643	84.12%
EyeShadows (Ours)	Wave (search)	4144***	94.65%
Duo-Reticles [19]	Slider (analog)	4705	14.43 cm
Dwell	Slider (analog)	2369	14.68 cm
Look&Cross [23]	Slider (analog)	2115	13.8 cm
EyeShadows (Ours)	Slider (analog)	1929***	13.64 cm

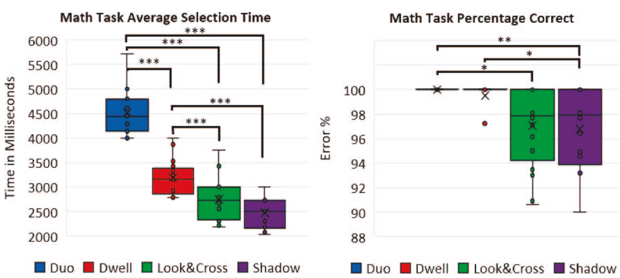

Figure 8: Box plots showing relative performance of all four selection methods in aggregate for the math task. The left chart shows the average speed of selection (lower is better) and the right chart shows percentage of correct answers out of the total answered (higher is better). Significance values are * for p < 0.05, ** for p < 0.01, and *** for p < 0.001.

Detailed results are divided into two categories, including Quantitative (speed and error) and Subjective (Likert ratings) metrics. Quantitative data was analyzed with Shapiro-Wilk tests for normality, F-tests for Analysis of Variance (ANOVA), and then with post-hoc t-tests if the F-tests were found to be significant. T-tests were corrected with Bonferronni correction for the (6) multiple comparisons between the four gaze-based selection conditions on a per-task basis. Subjective data and non-normally distributed performance data were analyzed using Kruskal-Wallis tests.

In addition to the sections that follow, performance results on a per-task basis are summarized in Table 2 and Figures 8, 9, and 10. Subjective ratings are summarized in Figure 11.

4.3.1 Quantitative Performance by Task

Quantitative data and significance values are listed below in the same order as the Figures. Note that EyeShadows is shortened to *Shadow* and Duo Reticles to *Duo* for brevity.

With regards to average **selection time for the math task** (Figure 8, left), ANOVA revealed a significant effect of condition ($F_{(3,64)}$ = 90.829, P < 0.001). A Shapiro-Wilk test for each condition did not show evidence of non-normality (W_{Duo} = 0.903, p = 0.076; W_{Dwell} = 0.906, p = 0.087; $W_{LookAndCross}$ = 0.939, p = 0.31; W_{Shadow} = 0.918, p = 0.138;). Two-tailed t-tests between pairs of selection methods (Bonferroni corrected) revealed significant differences between Duo and Dwell (t_{stat} = 23.612, P < 0.001), Duo and Look&Cross (t_{stat} = 20.199, P < 0.001), Duo and Shadow (t_{stat} = 29.318, P < 0.001), Dwell and Look&Cross (t_{stat} = 4.947, P < 0.001), and Dwell and Shadow (t_{stat} = 12.775, P < 0.001). No difference was found between Look&Cross and Shadow (t_{stat} = 2.922, P = 0.059).

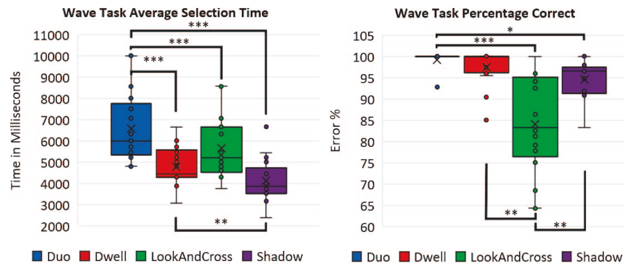

Figure 9: Box plots showing the relative accuracy of all four selection methods for the wave task. Significance values are represented by *s, where * is p < 0.05, ** is p < 0.01, and *** is p < 0.001.

With regards to average **error rate for the math task** (Figure 8, right), A Shapiro-Wilk test revealed evidence of non-normality (W_{Duo} > 1, p < 0.001; W_{Dwell} = 0.476, p = 0.001; $W_{LookAndCross}$ = 0.833, p < 0.01; W_{Shadow} = 0.918, p < 0.05.) Subsequently, a Kruskal-Wallis test revealed a significant effect of condition (χ^2 = 17.421, P < 0.001). Two-tailed t-tests between pairs of selection methods (Bonferroni corrected) revealed significant differences between Duo and Look&Cross (t_{stat} = 3.659, P < 0.05), Duo and Shadow (t_{stat} = 4.320, P < 0.01), and Dwell and Shadow (t_{stat} = 3.758, P < 0.05). No differences were found between Look&Cross and Shadow (t_{stat} = 0.255, P = 0.801), Duo and Dwell (t_{stat} = 1.851, P = 0.496), and Dwell and Look&Cross (t_{stat} = 2.784, P = 0.0796).

With regards to average **selection time for the wave task** (Figure 9, left), a Shapiro-Wilk test revealed evidence of non-normality (W_{Duo} > 0.862, p < 0.05; W_{Dwell} = 0.956, p = 0.558; $W_{LookAndCross}$ = 0.893, p = 0.0519; W_{Shadow} = 0.934, p = 0.254.) Subsequently, a Kruskal-Wallis test revealed a significant effect of condition (χ^2 = 26.299, P < 0.001). Two-tailed t-tests between pairs of selection methods (Bonferroni corrected) revealed significant differences between Duo and Dwell (t_{stat} = 5.248, P < 0.001), Duo and Look&Cross (t_{stat} = 3.978, P < 0.01), Duo and Shadow (t_{stat} = 8.526, P < 0.001), Dwell and Shadow (t_{stat} = 3.431, P < 0.05), and Look&Cross and Shadow (t_{stat} = 5.796, P < 0.001). No difference was found between Dwell and Look&Cross (t_{stat} = -2.833, P = 0.0719)

With regards to average **error rate for the wave task** (Figure 9, right), a Shapiro-Wilk test revealed evidence of non-normality (W_{Duo} > 0.389, p < 0.001; W_{Dwell} = 0.672, p = 0.001; $W_{LookAndCross}$ = 0.925, p = 0.181; W_{Shadow} = 0.896, p = 0.058) Subsequently, a Kruskal-Wallis test revealed a significant effect of condition (χ^2 = 30.178, P < 0.001). Two-tailed t-tests between pairs of selection methods (Bonferroni corrected) revealed significant differences between Duo and Look&Cross (t_{stat} = 5.791, P < 0.001), Duo and Shadow (t_{stat} = 3.580, P < 0.05), Dwell and Look&Cross (t_{stat} = 4.195, P < 0.01), and Look&Cross and Shadow (t_{stat} = -4.522, P < 0.01). No differences were found between Duo and Dwell (t_{stat} = 1.469 , P = 0.161) or Dwell and Shadow (t_{stat} = 1.921, P = 0.073).

With regards to average **selection time for the slider task** (Figure 10, left), a Shapiro-Wilk test revealed evidence of non-normality (W_{Duo} > 0.824, p < 0.01; W_{Dwell} = 0.839, p < 0.01; $W_{LookAndCross}$ = 0.874, < 0.05; W_{Shadow} = 0.835, < 0.01) Subsequently, a Kruskal-Wallis test revealed a significant effect of condition (χ^2 = 35.503, P < 0.001). Two-tailed t-tests between pairs of selection methods (Bonferroni corrected) revealed significant differences between Duo and Dwell (t_{stat} = 5.657, P < 0.001), Duo and Look&Cross (t_{stat} = 5.695, P < 0.001), and Duo and Shadow (t_{stat} = 6.337, P < 0.001). No differences were found between Dwell and Look&Cross (t_{stat} = 1.142, P = 1.619), Dwell and Shadow (t_{stat} = 1.848, P = 0.499), and

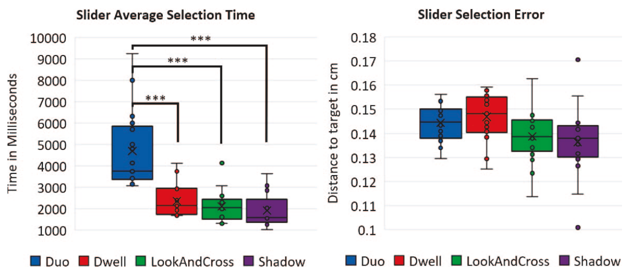

Figure 10: Box plots showing the slider results, where significance values are * for $p < 0.05$, ** for $p < 0.01$, for *** is $p < 0.001$.

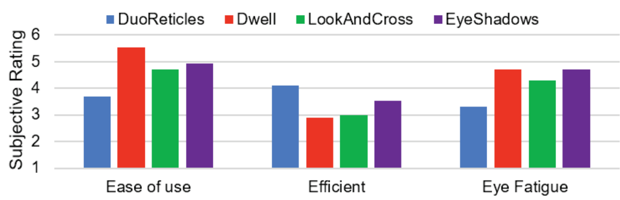

Figure 11: Results from the subjective questionnaire administered at the end of the experiment, which was rated on a 7-point Likert scale, with 1 being Strongly Disagree and 7 being Strongly Agree. No significant differences were found.

Look&Cross and Shadow ($t_{stat} = 1.011$, P = 1.96).

With regards to average **error for the slider task** (Figure 10, right), A Shapiro-Wilk test for each condition did not show evidence of non-normality ($W_{Duo} = 0.834$, p = 0.971; $W_{Dwell} = 0.929$, p = 0.215; $W_{LookAndCross} = 0.964$, p = 0.724; $W_{Shadow} = 0.948$, p = 0.425;). ANOVA revealed a significant effect of condition ($F_{(3,64)} = 3.144$, $P < 0.05$). Two-tailed t-tests between pairs of selection methods (Bonferroni corrected) did not reveal significant differences between Duo and Dwell (t_{stat} = -.918, P = 0.2.232), Duo and Look&Cross ($t_{stat} = 2.029$, $P < 0.356$), Duo and Shadow ($t_{stat} = 1.813$, $P < 0.531$), Dwell and Shadow ($t_{stat} = 2.056$, $P < 0.338$), Look&Cross and Shadow ($t_{stat} = 0.477$, P = 3.835), and Dwell and Look&Cross ($t_{stat} = 2.759$, P = 0.084).

4.3.2 Subjective Data

Subjective metrics were rated on a 7-point Likert scale and included the following statements on a scale of Strongly Disagree (1) to Strongly Agree (7): 1) I felt this method was easy to use. 2) I felt eye fatigue when using this method. 3) I felt this method was efficient. A summary of subjective results is shown in Figure 11. None of the three subjective questions were significant, including ease of use ($H_{stat(3,68)} = 6.583$ p = 0.087), eye fatigue ($H_{stat(3,68)} = 5.263$ p = 0.153), and efficiency ($H_{stat(3,68)} = 6.361$ p = 0.095).

5 Discussion and Future Work

Taking all of our results into account, EyeShadows demonstrated the potential to outperform baseline dwell selections for several tasks. Moreover, we demonstrated that it can be used in a wide variety of application scenarios, including fundamental virtual reality selection and manipulation and practical tasks in AR and MR. Though some adaptation is necessary, EyeShadows also works for both OST and VST display types, further supporting evidence for its versatility as an interactive tool. For the tasks other than Slider, EyeShadows error was somewhat higher in many cases. As such the size and location of shadows should be optimized to further improve accuracy.

Much like the Radii technique [25] leverages a circular, multi-level menu system for its head-crossing technique, we would like to explore shadow-specific multi-level menus for EyeShadows. It may be possible to cascade several menu options to deepen menu granularity and allow for additional context without clutter, similar to the EyeSQUAD technique [32]. Other tests to further refine parameters like size, position, and color would be beneficial to optimize performance. Judging from the better selection speeds of Look&Cross and EyeShadows, it's likely that certain applications would benefit from one technique over the other. For example, search and select tasks where information is densely arranged may benefit from the precision of head selection, whereas EyeShadows might be better suited for tasks with a larger number of sparsely located targets. Regarding color, certain tasks that are repetitive in nature might be even faster than as observed in experiments if a user can memorize the color for a selection. For example, if green is always yes, red is always no, and blue represents a neutral state (similar to the whiteboard application in Figure 5, users could make selections via peripheral examination alone.

5.1 Sources of Error

During the experiment, we notice that the eye tracker experienced jitter from time to time, especially for participants that wore glasses. This may have reduced the statistical power of results to some extent. In addition, our implementation of Duo Reticles was a best effort approach, and the results in our experiment did not match those of Piumsomboon et al. [19], as their method matched the performance of dwell-based selection in their paper. In that paper, the authors stated that DuoReticles performance was equivalent to Dwell, i.e., for 11 trials, Dwell took 92.73 sec (8.43 sec/trial) and DuoReticles took 94.68 sec (8.61 sec/trail) [19]. For the same type of task, our results were actually faster than the original DuoGaze on an absolute basis: Dwell: 4.79 sec/trail and DuoReticles: 6.58 sec per trial, though slower on a relative basis. Despite the different results, we reason that Dwell should still be an adequate baseline. Assuming Duo Reticles would have performed on par with dwell as demonstrated in [19], EyeShadows would still likely have performed best in terms of selection times.

Lastly, a significant portion of our experiment participants took breaks between the second and third tasks and reported minor eye fatigue towards the end of the experiment. This may have affected the slider task results, though the effect on our four selection conditions would be negligible since order was randomized and learning effects between task type were not studied.

5.2 Further Testing in VR

Another use case we tested was selection from a moving set of spheres that would constantly occlude each other. As shown in Figure 3, this task represents a difficult use case by including three randomly moving colored spheres that constantly occlude each other. The spheres are constantly and randomly moving in a confined 3D space, movement is defined by a randomly chosen vector, and the spheres bounce back into the test area when they reach a threshold boundary. Due to this boundary, spheres always stay within a cubic space that occupies approximately 10 degrees of visual field. Though we considered adding this task to the list of experiments, we decided to leave it out to ensure all participants would complete the tasks within one hour. Practical testing amongst the experimenters suggested that the EyeShadows mechanism also works for this type of occluded selection.

6 Conclusion

In this paper, we presented EyeShadows, a new way to quickly select and manipulate targets in virtual environments. By rendering a virtual copy of the target content in a peripheral location, we allow the user to grasp contextual information while avoiding the Midas touch problem, which can improve the way we make selections of virtual elements. Moreover, EyeShadows significantly outperformed dwell selection speeds for both basic selection and search and select tasks, demonstrating its potential to outperform conventional methods for selection. In addition, our system enables purely gaze-based interactions such as slider or dial control, extending its use case to manipulations in use cases outside of target selection.

We hope that this method will improve selection and manipulation times for tasks in virtual environments and encourage further research into gaze-based selection and interaction.

7 Acknowledgements

This research was funded in part by JSPS grant #21H03482 and NSF grant #2223035.

References

[1] S. Deng, J. Chang, S.-M. Hu, and J. J. Zhang. Gaze modulated disambiguation technique for gesture control in 3d virtual objects selection. In *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, pp. 1–8. IEEE, 2017.

[2] S. Esaki, Y. Ebisawa, A. Sugioka, and M. Konishi. Quick menu selection using eye blink for eye-slaved nonverbal communicator with video-based eye-gaze detection. In *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.'Magnificent Milestones and Emerging Opportunities in Medical Engineering'(Cat. No. 97CH36136)*, vol. 5, pp. 2322–2325. IEEE, 1997.

[3] W. Feng, M. Chen, and M. Betke. Target reverse crossing: a selection method for camera-based mouse-replacement systems. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 1–4, 2014.

[4] A. S. Fernandes, T. S. Murdison, and M. J. Proulx. Leveling the playing field: A comparative reevaluation of unmodified eye tracking as an input and interaction modality for vr. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2269–2279, 2023.

[5] B. Huynh, A. Wysopal, V. Ross, J. Orlosky, and T. Höllerer. Layerable apps: Comparing concurrent and exclusive display of augmented reality applications. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 857–863. IEEE, 2022.

[6] R. J. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems (TOIS)*, 9(2):152–169, 1991.

[7] S. Jalaliniya, D. Mardanbegi, and T. Pederson. Magic pointing for eyewear computers. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, ISWC '15, p. 155–158. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2802083.2802094

[8] A. Kurauchi, W. Feng, A. Joshi, C. Morimoto, and M. Betke. Eyeswipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pp. 1952–1956.

[9] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, and 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*, pp. 1–14, 2018.

[10] J.-J. Lee and J.-M. Park. 3d mirrored object selection for occluded objects in virtual environments. *IEEE Access*, 8:200259–200274, 2020.

[11] C. Liu, A. Plopski, and J. Orlosky. Orthogaze: Gaze-based three-dimensional object manipulation using orthogonal planes. *Computers & Graphics*, 89:1–10, 2020.

[12] P. Mohan, W. B. Goh, C.-W. Fu, and S.-K. Yeung. Dualgaze: Addressing the midas touch problem in gaze mediated vr interaction. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 79–84. IEEE, 2018.

[13] A. Mutasim, A. U. Batmaz, M. Hudhud Mughrabi, and W. Stuerzlinger. Performance analysis of saccades for primary and confirmatory target selection. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–12, 2022.

[14] A. K. Mutasim, A. U. Batmaz, and W. Stuerzlinger. Pinch, click, or dwell: Comparing different selection techniques for eye-gaze-based pointing in virtual reality. In *Acm symposium on eye tracking research and applications*, pp. 1–7, 2021.

[15] J. Orlosky, K. Kiyokawa, T. Toyama, and D. Sonntag. Halo content: Context-aware viewspace management for non-invasive augmented reality. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pp. 369–373, 2015.

[16] J. Orlosky, T. Toyama, K. Kiyokawa, and D. Sonntag. Modular: Eye-controlled vision augmentations for head mounted displays. *IEEE transactions on visualization and computer graphics*, 21(11):1259–1268, 2015.

[17] A. M. Penkar, C. Lutteroth, and G. Weber. Designing for the eye: Design parameters for dwell in gaze interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, p. 479–488. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2414536.2414609

[18] K. Pfeuffer and H. Gellersen. Gaze and touch interaction on tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, p. 301–311. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2984511.2984514

[19] T. Piumsomboon, G. Lee, R. W. Lindeman, and M. Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE symposium on 3D user interfaces (3DUI)*, pp. 36–39. IEEE, 2017.

[20] K.-J. Räihä and O. Špakov. Disambiguating ninja cursors with eye gaze. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1411–1414, 2009.

[21] R. Schweigert, V. Schwind, and S. Mayer. Eyepointing: A gaze-based selection technique. In *Proceedings of Mensch und Computer 2019*, pp. 719–723. 2019.

[22] L. Sidenmark, C. Clarke, X. Zhang, J. Phu, and H. Gellersen. Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality. In *Proceedings of the 2020 chi conference on human factors in computing systems*, pp. 1–13, 2020.

[23] L. Sidenmark and H. Gellersen. Eye&head: Synergetic eye and head movement for gaze pointing and selection. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*, pp. 1161–1174, 2019.

[24] L. Sidenmark, D. Mardanbegi, A. R. Gomez, C. Clarke, and H. Gellersen. Bimodalgaze: Seamlessly refined pointing with gaze and filtered gestural head movement. In *ACM Symposium on Eye Tracking Research and Applications*, pp. 1–9, 2020.

[25] L. Sidenmark, D. Potts, B. Bapisch, and H. Gellersen. Radi-eye: Hands-free radial interfaces for 3d interaction using gaze-activated head-crossing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2021.

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

[27] K. Takada, N. Kumasaki, T. Froese, K. Shibata, and S. Kasahara. Shadow clones: an unattended visual interface to enhance the task performance in multiple space using gaze-switching. In *Symposium on Spatial User Interaction*, pp. 1–2, 2022.

[28] M. Tatzgern, D. Kalkofen, R. Grasset, and D. Schmalstieg. Hedgehog labeling: View management techniques for external labels in 3d space. In *2014 IEEE Virtual Reality (VR)*, pp. 27–32. IEEE, 2014.

[29] R. J. Teather, W. Stuerzlinger, and A. Pavlovych. Fishtank fitts: a desktop vr testbed for evaluating 3d pointing techniques. In *CHI'14 Extd. Abstracts on Human Factors in Computing Systems*, pp. 519–522.

[30] J. van der Kamp and V. Sundstedt. Gaze and voice controlled drawing. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, NGCA '11. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/1983302.1983311

[31] M. Vidal, A. Bulling, and 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*, pp. 439–448, 2013.

[32] Y. Wang and R. Kopper. Efficient and accurate object 3d selection with eye tracking-based progressive refinement. *Frontiers in Virtual Reality*, 2:607165, 2021.

[33] R. C. Zeleznik, A. S. Forsberg, and J. P. Schulze. Look-that-there: Exploiting gaze in virtual reality interactions. *Brown Univ., Providence, RI, USA, Tech. Rep. CS-05*, 2005.