frontiers in Virtual Reality

ORIGINAL RESEARCH
published: 17 June 2021
doi: 10.3389/frvir.2021.607165

Efficient and Accurate Object 3D Selection With Eye Tracking-Based Progressive Refinement

Yunhan Wang[1] *and Regis Kopper*[2]*

[1]*Department of Mechanical Engineering & Materials Science, Duke University, Durham, NC, United States,* [2]*Department of Computer Science, University of North Carolina at Greensboro, Greensboro, NC, United States*

Selection by progressive refinement allows the accurate acquisition of targets with small visual sizes while keeping the required precision of the task low. Using the eyes as a means to perform 3D selections is naturally hindered by the low accuracy of eye movements. To account for this low accuracy, we propose to use the concept of progressive refinement to allow accurate 3D selection. We designed a novel eye tracking selection technique with progressive refinement–Eye-controlled Sphere-casting refined by QUAD-menu (EyeSQUAD). We propose an approximation method to stabilize the calculated point-of-regard and a space partitioning method to improve computation. We evaluated the performance of EyeSQUAD in comparison to two previous selection techniques–ray-casting and SQUAD–under different target size and distractor density conditions. Results show that EyeSQUAD outperforms previous eye tracking-based selection techniques, is more accurate and can achieve similar selection speed as ray-casting, and is less accurate and slower than SQUAD. We discuss implications of designing eye tracking-based progressive refinement interaction techniques and provide a potential solution for multimodal user interfaces with eye tracking.

Keywords: selection, progressive refinement, eye tracking, 3D user interfaces, interaction

OPEN ACCESS

Edited by:
Thierry Duval,
IMT Atlantique Bretagne-Pays de la Loire, France

Reviewed by:
Frank Steinicke,
University of Hamburg, Germany
Ken Pfeuffer,
Munich University of the Federal Armed Forces, Germany

****Correspondence:***
Regis Kopper
kopper@uncg.edu

Specialty section:
This article was submitted to Technologies for VR, a section of the journal Frontiers in Virtual Reality

Received: *16 September 2020*
Accepted: *25 May 2021*
Published: *17 June 2021*

Citation:
Wang Y and Kopper R (2021) Efficient and Accurate Object 3D Selection With Eye Tracking-Based Progressive Refinement. Front. Virtual Real. 2:607165. doi: 10.3389/frvir.2021.607165

1 INTRODUCTION

Interaction techniques in immersive virtual environments are essential means to offer the ability for users to affect the virtual scene and improve the experience. Object manipulation is a very common class of 3D interaction techniques which consists of several subtasks such as positioning, rotation and scaling (LaViola et al., 2017). As one of the most basic tasks, selection requires users to perform a “target acquisition task” (LaViola et al., 2017). Without selection techniques, virtual contents are impossible to be manipulated in the first place.

Ray-casting (Mine, 1995) is one of the most commonly used selection techniques in virtual reality (VR) due to its ease of implementation and use. With ray-casting, a ray is cast into the environment, and the object hit by the ray is selected when the user presses a button. It is easy to learn as well as easy to perform. However, ray-casting becomes inefficient when targets are small or remote due to their small visual size (Poupyrev et al., 1998; Kopper et al., 2010). Researchers have presented many possible solutions and new techniques in order to solve the precision issues of ray-casting (De Haan et al., 2005; Frees et al., 2007; Vanacken et al., 2007; Kopper et al., 2011).

In some edge situations, for example, in a cluttered virtual environment, even improvements to traditional ray-casting may not be sufficient, as targets may be too many, too small or too occluded to allow usable immediate interaction. Kopper et al. (2011) proposed to address these cases with the concept of selection by progressive refinement. Contrary to direct selection, performing selection by progressive refinement breaks down the task into multiple refinement steps, where each step does not require high precision and reduces the set of selectable targets, effectively allowing accurate, but low precision selection.

With the availability of built-in eye tracking on consumer head-mounted displays, leveraging its potential for improving interaction techniques is a logical step. However, eye movements are not always consistent with the focus of attention but occur in rapid movements known as saccades (Robinson, 1964), often not leading to predictable results (Findlay, 1982). This nature of eye movements have caused previous attempts at using the eyes as a means to control VR interaction unsuccessful (Cournia et al., 2003; Smith and Graham, 2006).

In this work, we leverage the low accuracy required by progressive refinement techniques and apply it in a multimodal eye tracking-based 3D selection technique, called Eye-controlled Sphere-casting followed by QUAD-menu (EyeSQUAD). With EyeSQUAD, users first roughly select a set of objects in the vicinity of the target with a selection sphere whose center is determined by the eye movements. We used an approximation method to stabilize the point-of-regard calculated from the eye ray data. After that, similar to SQUAD, the initial objects selected by the selection sphere are evenly and randomly distributed on a quad-menu which consists of four quadrants. The refinements are achieved by gazing in the direction of the quadrant containing the target object. In our current implementation, each refinement phase is triggered by a button press. Similar to SQUAD, EyeSQUAD presents the tradeoff between speed and accuracy. In highly cluttered environments, using a direct eye-gaze based technique would lead to very low accuracy. On the other hand, multiple refinements increase the selection time, while maintaining the accuracy high. We also note that EyeSQUAD scales quite well, allowing the reduction of 1,024 candidate objects into a single target in only five refinement phases (Kopper et al., 2011).

Importantly, usable interaction through eye-tracking can serve as a viable assistive interface to allow users with low large muscle group mobility to achieve a high quality experience when interacting in VR environments. We believe that designers should strive for universal usability in every type of user interface. Due to the nature of spatial input in VR, this is especially challenging, and our proposed solution of progressively refined eye tracking-based 3D selection is a step towards universal access for VR interfaces.

In the remainder of the paper, we review previous high precision hand-based selection techniques, provide a background on eye-tracking-based research in human-computer interaction, cover research on eye-tracking as an assistive interface and finally discuss other uses of eye tracking in VR. We then present EyeSQUAD, the eye-tracking based selection technique we have developed and discuss technical aspects of its implementation, such as calculating the nearest target and partitioning the selection space. We then describe the user study we conducted to evaluate EyeSQUAD, followed by the description of the results. The paper continues with a discussion of the results, followed by a list of the limitations in our work and we end by presenting the conclusions of our work and future research directions.

2 RELATED WORK

In this section, we cover research related to our work, in terms of its motivation and related techniques. One of the main goals of EyeSQUAD is to offer high precision selection with low required accuracy, and we cover related research on **section 2.1**. The goal of EyeSQUAD is to offer a high usability human-computer interaction technique that uses eye tracking, and we cover related research on **section 2.2**. Additionally, one of the applications of EyeSQUAD is to allow users with upper limb mobility impairments to successfully interact in Virtual Reality; thus, we cover related literature on eye tracking in assistive technologies in **section 2.3**. Finally, we cover related literature on virtual reality eye-tracking-based techniques and contrast it with EyeSQUAD on **section 2.4**.

2.1 High-Precision Hand-Based Selection Techniques

Many researchers have addressed solutions and proposed techniques to solve precision issues with ray-casting.

Frees et al. (2007) designed the Precise and Rapid Interaction through Scaled Manipulation (PRISM) interaction technique that can improve the accuracy and precision of standard ray-casting by changing the control/display ratio with respect to hand speed. Vanacken et al. (2007) presented the depth ray and 3D bubble cursor techniques that can select even invisible targets occluded by other objects. The IntenSelect technique by De Haan et al. (2005), a revised version of ray-casting, is combined with a dynamic rating system for all objects in the environment which can dynamically generate a banded ray between the highest score target and the controller.

Kopper et al. (2011) introduced a new concept of selection by progressive refinement. Here, a set of objects that contains the target is coarsely selected and then low precision refine steps are done until only the target remains. Based on this concept, the authors designed the sphere-casting refined by quad-menu (SQUAD) selection technique for selection in cluttered environments. The user first selects a set of objects in the environment with a selection sphere; in the subsequent steps, the objects contained in the initial sphere are evenly distributed and refined through coarse selection of a quadrant in an out-of-context quad-menu. SQUAD outperforms ray-casting in speed when targets are small and leads to virtually no errors.

All the selection techniques mentioned above depend on hand interaction. Eye tracking has been described as “an attractive input for VR” (Pfeiffer, 2008). Selection with eye movements can be more intuitive and faster than hand-based selection as eye fixation at the target is a prerequisite for these previous selection techniques. Thus, implementing eye tracking could potentially save selection time and provide a more intuitive experience.

2.2 Eye Tracking in Human-Computer Interaction

Eye tracking applications are categorized as diagnostic and interactive depending on whether eye tracking is regarded as an input or analytical tool (Duchowski, 2007). Interactive eye tracking techniques can be further categorized into selective and gaze-contingent subtypes (Duchowski, 2007). Selective type of eye tracking techniques, as the main focus in our work presented here, replace usual inputs such as mouse and controller with the point-of-regard. Gaze-contingent eye tracking applications utilize the gaze data to improve the quality of display rendering with techniques such as foveated rendering (Guenter et al., 2012; Patney et al., 2016).

The "Midas Touch" problem is one of the main issues with control by eye movements (Jacob and Karn, 2009). It is characterized by the uncertainty on whether there is intention to activate or just look at an object. In our implementation, we avoid the "Midas Touch" by using an explicit command (a button click) to trigger the action.

Eye tracking technology is widely used in a broad variety of disciplines such as neuroscience (Khushaba et al., 2013), psychology (Iacono et al., 1982), marketing/advertising (Wedel and Pieters, 2008) and human factors (Dishart and Land, 1998), providing objective diagnostic evidence. Eye Movements can also be utilized in usability research, where Jacob and Karn (2009) described their use as "rising from the ashes" rather than "taking off like wildfire". They noted that eye tracking had been impeded from being widely implemented due to technical problems in usability studies, labor-intensive data extraction and difficulties in data interpretation. With new displays that incorporate eye tracking technology, such as the FOVE (FOVE, 2018), along with new systems and toolkits to process eye-tracking data, such as PyGaze (Dalmaijer et al., 2014), these technical challenges have been reduced.

However, directly implementing eye tracking as a human-computer interaction input method is still challenging because even current technology cannot ensure a robust performance of both tracking and calculation of correct point-of-regard without constant re-calibration, especially with head-mounted eye-tracking systems (Fuhl et al., 2016). Due to these limitations, progressive refinement seems like a viable choice to achieve high quality eye tracking-based interaction.

2.3 Eye Tracking in Assistive Technologies

Eye tracking has been applied as an assistive technology for the physically disabled. Dv et al. (2018) proposed a gaze-controlled interface (GCI) for individuals with physical impairment, and found that selection time can be significantly reduced by a GCI task in pointing and selection tasks as compared to traditional hand-based input. However, their GCI activates the selection by gazing at a button and waiting for a period of time which can potentially increase significant selection time, and also is subject to "Midas Touch" artifacts. Meena et al. (2016, 2017) proposed a solution to this problem through a soft-switch. They designed multimodal graphical user interfaces (GUIs) to control a power wheelchair (Meena et al., 2017) and a virtual keyboard (Meena et al., 2016) using a touch pad as a soft switch that adresses the "Midas Touch" problem. In both the wheelchair and virtual keyboard studies, the eye-tracker combined with a soft-switch control outperformed the eye-tracker only technique. In our implementation, we address this issue by employing a finger trigger button that can be replaced by other types of input in future implementations.

Although eye tracking for assistive interfaces has been proven feasible and useful in previous work, it can also be frustrating if it is not carefully designed. For example, Creed (2016) leveraged the Tobii EyeX tracker for assisting disabled artists. In his experiments, users were asked to perform four 2D tasks including selection with eye tracking. Even though the results of this effort were mixed, and the user experience was frustrating at times, we can derive important insight from these low usability experiences. The high precision required due to high cursor sensitivity can lead to severe usability issues for direct gaze-based control. Significant physical issues such as eye strain, head tension and tiredness were reported due to the high precision required by the tasks. For this reason, narrowing down the precision requirement of eye tracking techniques is necessary for a comfortable and usable experience of a gaze-based interaction technique. Progressive refinement can be a promising solution which can break a high precision demanding task into several low precision subtasks. In easy tasks, eye tracking techniques can even outperform manual techniques (Meena et al., 2017; DV et al., 2018).

2.4 Eye Tracking Techniques in Virtual Reality

Control by eye movement can be implemented in VR as virtual environments usually contain far-away objects spread out in a large three-dimensional space (Poole and Ball, 2006), making them easily switched with saccadic eye movements.

Tanriverdi and Jacob (2000) designed an eye tracking navigation technique and evaluated its on spatial memory compared with a hand-pointing technique. In this study, targets were salient, large and relatively close to the user, and the hand-based pointing technique utilized the same selection mechanism as the eye gaze variant. They found that eye tracking could be faster than the hand-pointing technique especially in distant virtual environments although hand-pointing yielded better spatial memory. Their study showed that interaction with eye movements can be feasible even with relatively low accuracy eye tracking hardware.

Pfeiffer (2008) tested the precision of an eye tracker in a CAVE-type VR system. They achieved a precision of about 1° horizontally and 2° vertically. This level of precision is sufficient for selection of large targets, but it is not sufficient in situations where targets have small visual sizes (Kopper et al., 2011). In such situations, strategies to overcome the precision limitations of gaze-based control should be studied.

Miniotas et al. (2004) coupled the idea of expanding targets with an eye-based selection technique. With the benefit of a "grab-and-hold" algorithm, they achieved a 9.9% error rate. The algorithm has reduced 57% error rates from pure eye selection in their experiments.

Ashmore et al. (2005) utilized a fisheye lens to perform pointing and selection with eyes with different interaction styles (i.e., "MAGIC" (Zhai et al., 1999) and the "grab-and-hold" (Miniotas and Špakov, 2004) styles). Their study shows the technique with MAGIC style has the best performance on speed (3.359 s) and accuracy (18% abort rate and average 14.26 pixels deviation from center of the target).

Kumar et al. (2007) presented an eye tracking selection technique called EyePoint with a look-press-look-release pattern by using eye gaze and keyboard triggers, taking advantage of progressive refinement to compensate the accuracy of the eye trackers. Their work proved combining progressive refinement with eye tracking for virtual selection is practical although its error rate was non negligible (13%).

Pfeuffer et al. (2017) designed a gaze + pinch interaction technique that supports unimanual/bimanual and single/two objects. They conducted an informal evaluation on the technique under four interesting application cases (e.g. building molecules, zooming gallery) and collected qualitative user feedback. Feedback shows users were generally positive about the technique. However, users did encounter the "Midas Touch" problem and the usability of the technique was harmed by the accuracy and stability of the hand and eye tracking.

Piumsomboon et al. (2017) presented three novel eye-gaze-based interaction techniques: 1) Duo-reticles, represents current eye-gaze position and a near past moving-average gaze location with two reticles, and achieves the selection when the two reticles align; 2) Radial pursuit, hits an object with eye-gaze reticle, expands all objects within a selection sphere, and pursues the target object; 3) Nod and roll–combining head gestures with eye gaze, prevents influence of head nodding on eye fixation with benefit of vestibulo-ocular reflex. The authors conducted an initial user study for the first two techniques with a small sample size and found the performance of the two techniques are similar to Gaze-Dwell (Majaranta et al., 2009).

Khamis et al. (2018) combined the concept of pursuits with an eye tracking selection technique which finds the target whose movement correlates most with eye movements with comparison between eye movements and the movements of targets in the virtual environment. Their user study showed 79% as the highest accuracy.

Sidenmark et al. (2020) Proposed a gaze-based selection technique that allow the selection of partially occluded objects in virtual environments. They leverage the human ability to achieve smooth eye movements during pursuit of objects by animating the rendering of the object outline during the selection process. Although the technique had high selection time and error rates as compared to a hand-based equivalent technique, it required fewer movements than standard ray-casting in highly occluded conditions. The gaze-based Outline Pursuit technique addresses the issue of occlusion, but it is not appropriate for highly cluttered environments, which would make the selection of candidate objects unfeasible.

Recently, there has been intense research activity involving the use of eye-tracking in virtual reality. Pfeuffer et al. (2020) evaluated the use of gaze for the selection of menu items in virtual environments. By nature, menus consist of few and uncluttered items, near the user sight. Sidenmark and Gellersen (2019) looked at head and gaze coordination on selection of targets 4° in diameter in a circular display. There have been also research on the use of eye gaze in social VR settings (Rivu et al., 2020, e.g.), determining behavior (e.g., Pfeuffer et al., 2019) and for selection of tools through a see-though interface (Mardanbegi et al., 2019).

Prior work discussed the design and evaluation of four eye-gaze-based selection techniques, combining aspects of progressive refinement selection with eye tracking (Stellmach and Dachselt, 2012). In this work, the authors propose a set of techniques that use gaze as the coarse indicator of selection region, and the refinement is done by a manual action. Their work shows that performance is improved over simple gaze cursor control, and they did not evaluate the technique on highly cluttered environments.

Even though there has recently been intense research on eye-tracking based selection for virtual environments, we failed to identify an eye-gaze technique that achieves high accuracy in 3D cluttered environments. We achieve this with EyeSQUAD, whose design is detailed in the following section.

3 EyeSQUAD SELECTION

We designed a novel selection technique with eye tracking–Eye-controlled Sphere-casting refined by QUAD-menu (EyeSQUAD). EyeSQUAD combines eye tracking with SQUAD, a hand-based progressive refinement selection technique proposed by Kopper et al. (2011). Selection with Progressive refinement is an indirect method of selection which allows users to first select a set of objects including the target and then reduce the number of selectable objects through step-by-step refinements until only the target remains. As with the original SQUAD technique, EyeSQUAD is divided into two subtasks: sphere-casting and quad-menu refinement.

Sphere-casting For the sphere-casting subtask, rather than casting a sphere by a controller, EyeSQUAD allows the user to control the selection sphere with eyes by calculating the convergence point from the user's eye ray data. We set the diameter of the selection sphere to be 26.3° which is consistent with the angular size of the selection bubble in SQUAD. The size of selection sphere is visually constant to prevent the sphere from being too small or oversized when it is far away or too close. The objects inside of the selection sphere will be chosen as the initial set of objects that need to be further refined when user performs the initial selection (**Figure 1**).

QUAD-menu Refinements Once the sphere-casting selection has been triggered, the set of objects inside it are evenly and randomly distributed on an out-of-context quad-menu (**Figure 1**). Users then refine the set of selectable objects by gazing in the direction of the quadrant that contains the target and trigger the selection again. The objects on that quadrant will be distributed again across the quad-menu (**Figure 1**), reducing the number of total candidate objects by a factor of approximately

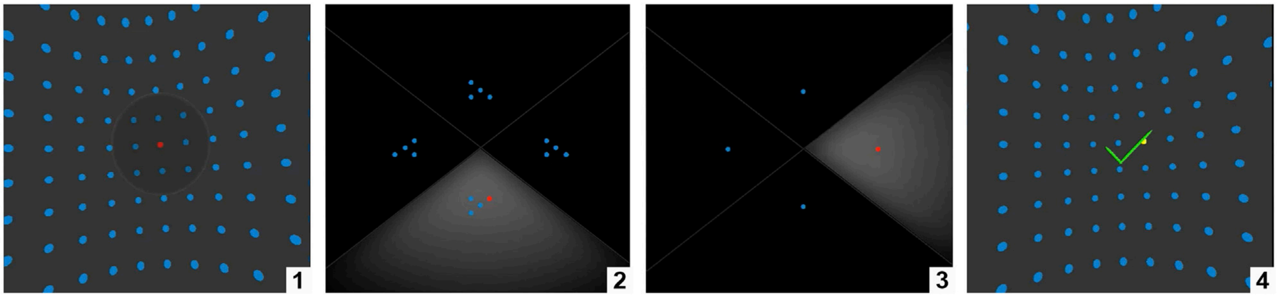

FIGURE 1 | EyeSQUAD selection process (monocular view): 1. main scene (red dot: target, blue dot: distractor), 2. first QUAD-menu progressive refinement, 3. later QUAD-menu progressive refinement(s), 4. back to the original scene.

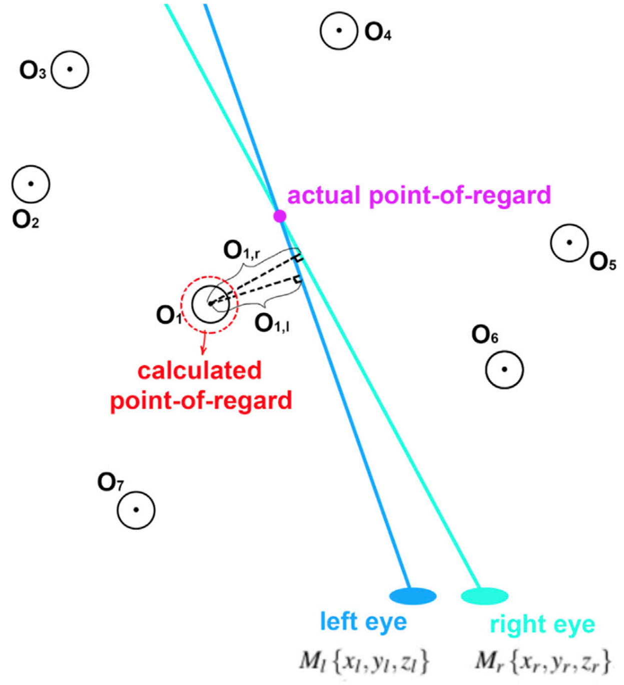

FIGURE 2 | Schematic of the closest target approximation method.

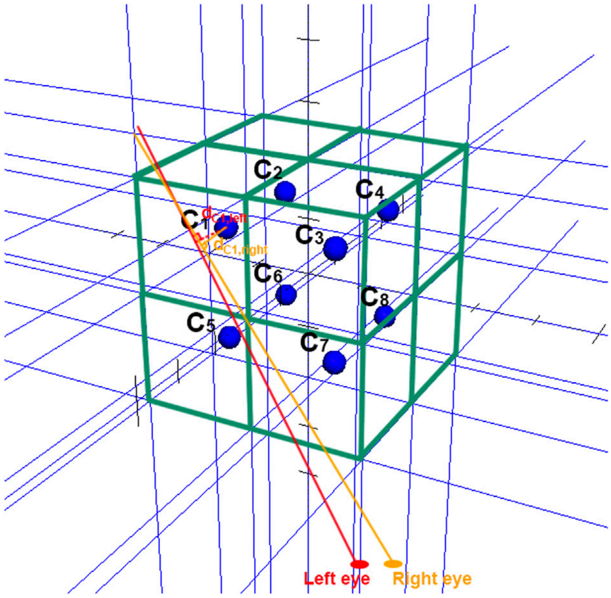

FIGURE 3 | Schematic of the space partition method.

4. This process progressively refines the set of candidate objects until the only object in a quadrant is the target. Once the task is completed, the user is transferred back to the original scene (**Figure 1**).

Triggering Mechanism In order to avoid "Midas Touch" artifacts, the refinement triggers are activated by the depression of a button in an HTC Vive controller. Our focus with EyeSQUAD is in the quality of the eye-based pointing performance, and we leave to future work the investigation of hands-free triggering mechanisms.

3.1 Closest Target Approximation

We developed a "closest target approximation" method to stabilize the calculated point-of-regard. Instead of directly controlling the selection sphere by the calculated convergence point, the selection sphere always moves to the closest target in the environment which is determined by calculating the minimum summation of the distance from the target to the two eye rays. This approximation of constraining the selection sphere center to the nearest object reduces jitter and ensures that at least one potential target is contained by it.

As shown in **Figure 2**, we suppose the positions of the left and right eyes in the environment are respectively $M_l\{x_l, y_l, z_l\}$ and, while the directions of two eyes are respectively $S_l\{\alpha_l, \beta_l, \gamma_l\}$ and $S_r\{\alpha_r, \beta_r, \gamma_r\}$. The position of a certain object i in the environment is written as $M_{O_i}\{x_i, y_i, z_i\}$.

$$\overline{M_{O_i}M_l} = \{x_i - x_l, y_i - y_l, z_i - z_l\} \tag{1}$$

$$\overline{M_{O_i}M_l} \times \overline{S_l} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ x_i - x_l & y_i - y_l & z_i - z_l \\ \alpha_l & \beta_l & \gamma_l \end{vmatrix} \tag{2}$$

We then calculate the distance from the object *i* to the left eye ray,

$$d_{O_i,l} = \frac{\left|\overline{M_{O_i}M_l} \times \overline{S_l}\right|}{\left|\overline{S_l}\right|} \tag{3}$$

Similarly, for the distance from the object *i* to the right eye ray. We now sum the distances from object *i* to two eye rays.

$$d_{O_i,sum} = d_{O_i,l} + d_{O_i,r} = \frac{\left|\overline{M_{O_i}M_l} \times \overline{S_l}\right|}{\left|\overline{S_l}\right|} + \frac{\left|\overline{M_{O_i}M_r} \times \overline{S_r}\right|}{\left|\overline{S_r}\right|} \tag{4}$$

Finally, we find the closest target by

$$t = \underset{i \in \{1,\cdots,n\}}{argmin}\left(d_{O_i,sum}\right) \tag{5}$$

The calculated point-of-regard is the position of the closest object $M_{O_t}\{x_t, y_t, z_t\}$. The sphere will always move to closest target with a constant speed of 6 m/s. Since the movements of eyes are saccadic (Deubel and Schneider, 1996), we decided to have a continuous motion rather than instant transform to the calculated closest target to ensure a visually smooth motion of the sphere.

3.2 Space Partition

In order to save computational time and optimize the closest target approximation algorithm, we partition the space (**Figure 3**) with regards to the magnitude of target positions in the environment using octrees (Meagher, 1980). We then find the closest center of a partitioned part that contains targets (e.g., C1 in **Figure 3**). Empty parts are just ignored.

We only apply the closest target approximation method in the part nearest to the eye rays. If the size of space is small enough (10^{-3} m^3 in our study), the method returns the position calculated from the closest target approximation which will be regarded as the current calculated point-of-regard. Otherwise, the method will partition the space C into several parts (8 parts in our study). Then, by regarding the centers of those partitioned parts as the objects (e.g., C1 to C8 in **Figure 3**), a closest partitioned part (e.g. C1) can be obtained with the closest target approximation method, which will be used as the input to recursively call this space partition algorithm. Thus, the whole space can be partitioned into smaller parts until it finally finds the closest target. All objects being static, pre-processing provided constant time accessing the objects within a specific small space. This way, time complexity was improved to the order of $O(logN)$ assuming evenly distributed objects in the space.

4 METHODS

4.1 Experimental Design

We evaluated the performance of EyeSQUAD and compared it to ray-casting (Mine, 1995) and to SQUAD (Kopper et al., 2011) for a selection task–acquisition of a target surrounded by several distractors in a virtual environment. The size of these objects and density of the distractors vary across different conditions in the experiment.

4.1.1 Goals and Hypotheses

The purpose of the experiment was to determine the tradeoffs between EyeSQUAD and two other previous selection techniques–ray-casting and SQUAD. Ray-casting only requires a single but accurate selection while SQUAD and EyeSQUAD allow the user to select with little precision in each step but requires several steps. We estimated that EyeSQUAD would enhance accuracy and speed of SQUAD as selection with the eyes do not require large muscle groups engagement and could be performed faster.

This study aimed to answer two research questions:

1) Can EyeSQUAD outperform previous selection techniques such as ray-casting and SQUAD?
2) Will target size or distractor density have influence on the performance of selection techniques?

With the consideration of the tradeoffs and the research questions, we hypothesized that.

H1) The time of selecting a target with SQUAD or EyeSQUAD will not be affected by the target size while ray-casting will be slow with small targets and fast with large targets.

H2) The time of selecting a target with SQUAD or EyeSQUAD will be proportional to the number of distractors in the virtual environment while the performance of ray-casting will not be influenced by the distractor density.

H3) EyeSQUAD will outperform ray-casting when number of distractors is small with respect to speed and accuracy.

H4) EyeSQUAD will outperform SQUAD in all conditions with respect to speed and accuracy.

H5) SQUAD and EyeSQUAD will have virtually no errors due to their low requirement of precision while ray-casting will increase errors with decreasing the target size.

4.1.2 Design

Since individual differences with eye tracking methods is significant (Goldberg and Wichansky, 2003), we used a factorial within-subject design with repeated measures. There are three independent variables: technique (ray-casting, SQUAD, EyeSQUAD), target size (small: radius 0.01 m or 0.26, medium: radius 0.015 m or 0.40°, large: radius 0.04 m or 1.06°), and distractor density (sparse: 16, medium: 64, dense: 256). This design is thus 3 × 3 × 3. The dependent variables in the study were time to complete a task, error rate and subjective ratings of each technique. Regarding the error, we count any failure during the selection process as an error. For ray-casting, error is counted when a wrong object is selected. For EyeSQUAD and SQUAD, an error occurs when the target is absent from the initial selection sphere or when selecting a quadrant that does not include the target.

The order of the presentation of technique was counterbalanced while each of the nine conditions of target size vs. distractor density was repeated 8 times and presented in random order.

4.2 Apparatus

We used the FOVE head-mounted display (HMD) (weight: 520 g) (FOVE, 2018) which offers a built-in eye tracker. Since our research required subjects to stand at a fixed point within a

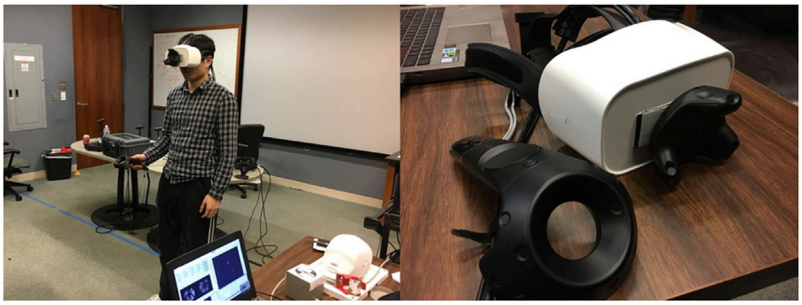

FIGURE 4 | One of the authors showing the experimental setup and the FOVE headset (with an HTC Vive tracker) and an HTC Vive Controller.

room tracking space during the experiment, we used the HTC Vive tracking system, with an incorporated Vive Tracker (weight: 300 g) onto the FOVE headset and a Vive Controller for interaction (**Figure 4**). To achieve that, one laptop connected with the FOVE headset provides the display of virtual contents and eye tracking while one desktop connected to the HTC headset provides headset and controller tracking.

For the software, the FOVE Unity plugin v1.3.0 was used driven by FOVE Driver Version 0.13.0 on an ASUS GL502V Quad-Core Processor (2.8 GHz), 16.0 GB RAM, with NVIDIA GeForce GTX 1070 running Windows 10. An Alienware X51 R3 Edition, with Quad Core Processor (2.7 GHz), 8GB RAM, NVIDIA GeForce GTX 970 running Windows 10, was used to drive the HTC SteamVR plugin v1.2.3 to support the HTC Vive tracking deviced with 6DOF position and orientation. A local server was built for supporting real-time data transfer between the two PCs through UDP. All positional tracking was done by the HTC Vive Tracker and the Unity application directly applied the transformations in world coordinates coming through the socket connection to the camera view. The eye-gaze data was captured with the FOVE API, and had no relation to the positional tracking data.

The virtual environment was made with Unity 3D Engine (version 2017.1) and the scripts were written with C#. All virtual objects including a target and other distractors were circular, static and located on the surface of an invisible sphere with a radius of 2.155 m whose center was the position of the participant. The user was positioned in the center of this sphere, which ensured that all objects had the same visual size from the user's perspective (Kopper et al., 2011). The target was chosen within an inner sphere (radius 1.1 m) which ensured the target fell within the initial field-of-view. The selection bubble would at most cover constant number of objects (e.g., 16, 64, 256) in one selection. This ensured a constant number of refinement steps (e.g., 2, 3, 4) under each target density condition.

Before each selection task, since participant height varied, a reset session was included to ensure that participants began from the same position relative to the objects, as the difficulty of a pointing task is positively correlated with the visual size of the target and the amplitude of movement to accomplish the task (Fitts, 1954; Kopper et al., 2010). Participants could take a small break in the reset session if they needed. During the reset session, they used the controller or eyes to move a dot into a large circle in the center of the screen and press the touch pad on the Vive controller to proceed. Timing only started after the reset session.

4.3 Participants

24 voluntary unpaid participants (12 male, 12 female) were recruited for the experiment whose age ranged from 21 to 32 years old with a median age of 24 years old. All of the participants were graduate students except for one, who was a postdoctoral scholar.

4.4 Procedure

Participants were first welcomed by the experimenter and given background information of the study. Then participants read and signed an informed consent form. After that, they were asked to complete a subset of the Ishihara color blindness test (Birch, 1997) and a background survey online. No participants were excluded from the experiment due to color blindness.

Participants were emphasized to perform the trials as quickly as possible while making as few mistakes as they could, giving priority to making few mistakes. Then the experimenter explained how to complete the selection task. They were notified that they should hold the controller with their dominant hand and could not use the other hand to steady the controller through all trials. Once the experimenter finished the explanation of usage of the Vive controller, the participants were asked to move to the experiment area which was marked by a dot on the floor of the room-sized tracking space and fitted with the FOVE HMD. There was also a red starting point in the virtual environment which was consistent with the starting point in the real world and would change to green if the participant was close enough to it (<0.1 m).

Participants then started learning their first technique in a corresponding training session which tought them how to use the technique and allowed them to try all nine combinations of target size and distractor density conditions once. During the training session, participants would be told to accomplish at least one correct selection and one error selection to see both results. (For correct selection, a check mark was displayed. Otherwise a cross appeared.) After the training session, participants performed the corresponding experimental condition of the technique which contained 72 trials with all 9 combinations repeated 8 times in random order.

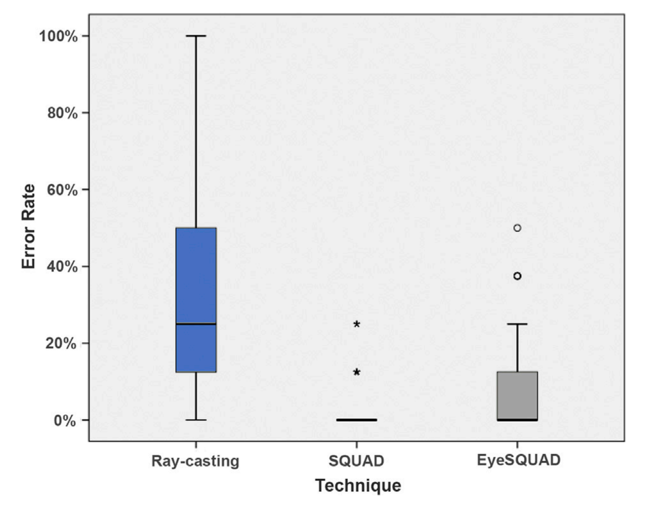

FIGURE 5 | Mean error rate with different techniques.

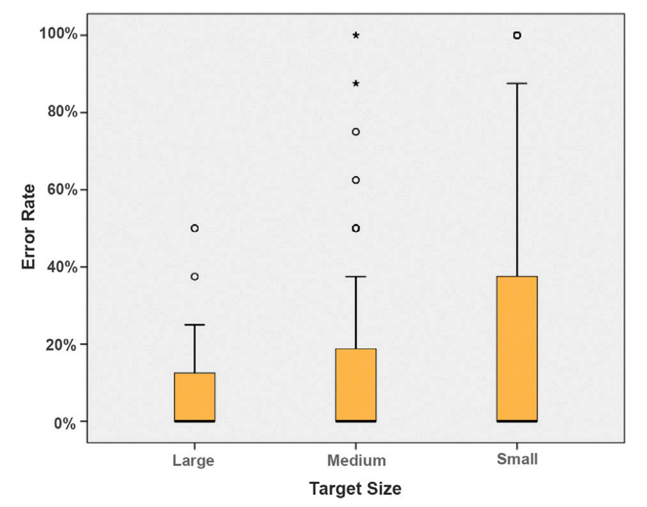

FIGURE 6 | Mean error rate with different target sizes.

Once completed all of these trials, participants filled in a technique rating questionnaire for the technique they just performed. All participants accomplished all three techniques in a specific order that was counterbalanced. After finishing all of these conditions, they filled a post-study overall performance questionnaire.

5 RESULTS

We used a repeated-measures multi-variate ANOVA (MANOVA) model with $\alpha = 0.05$ to evaluate mean error rate and average time to complete selection. There are three independent variables: technique (Ray-casting, SQUAD, EyeSQUAD), target size (small, medium, large) and distractor density (sparse, medium, dense).

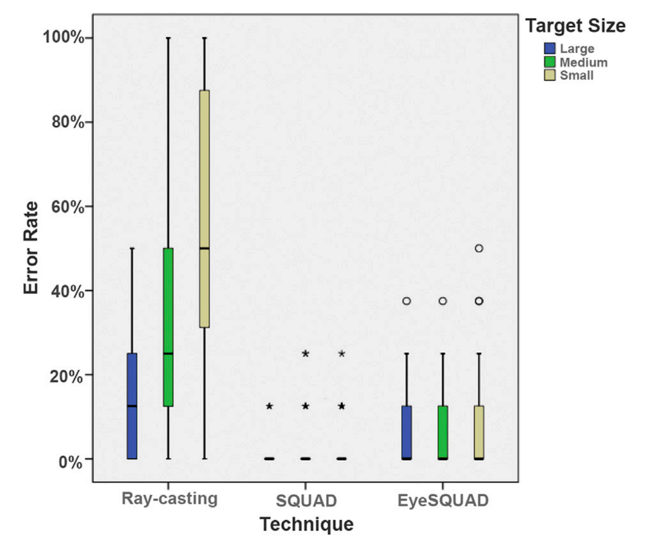

FIGURE 7 | The interaction between technique and target size on errors.

We checked the statistical power of results to ensure that the significance of the effect of a factor was not exaggerated. We accepted significance when power was larger than 0.8, which indicates "sufficient power to detect effects" (Field, 2013). From the results, we find that our sample size is enough since sufficient power were observed in all the results where $p < .05$. We used the Bonferroni correction to account for multiple comparisons. The Bonferroni correction was used to lower the alpha threshold with respect to the multiple comparisons and avoid spurious positives.

5.1 Mean Error Rate

Overall, technique had a significant effect ($F_{0.95;2,36} = 97.986, p < .001, power > .999$) on the error. As shown in **Figure 5**, ray-casting leads a significantly higher mean error rate of 34.2% compared with the mean error rate of 0.9% of SQUAD ($p < .001$) and the mean error rate of 6.2% of EyeSQUAD ($p < .001$). SQUAD has a significantly lower mean error rate than EyeSQUAD ($p < .001$).

Apart from technique, target size (**Figure 6**) had a significant effect on error ($F_{0.95;2,36} = 36.762, p < .001, power > .999$) while distractor density was not significant ($F_{0.95;2,36} = 0.432, p = .654, power = .112$). This shows that the effect of target size on ray-casting is so significant that even averaged on overall techniques it is still significant. Performing pairwise comparisons with the Bonferroni test, large target size had significant lower error rate than medium target size ($p < .0167$) and medium target size had significant lower error rate than small target size ($p < .001$). Contrary to target size, the effect of distractor density is not significant once averaged on overall techniques.

Furthermore, the interaction of technique and target size ($F_{0.95;4,72} = 29.874, p < .001, power > .999$) had a significant effect (**Figure 7**). Consistent with the results of the study of Kopper et al. (2011), there is a significant effect of target size with ray-casting. The Bonferroni test indicates that the medium size had significantly higher error rate than the large size ($p < .0167$) while the small size had significantly higher error rate than the medium size ($p < .001$). As expected, however, the target size had no significant effects on techniques with progressive refinement, that is, SQUAD ($p = .534$) and EyeSQUAD ($p = .817$) techniques. This is consistent with the latter part of H5, which states that ray-casting increases errors with smaller targets.

After examining the contribution of target size on the interaction effect between technique and target size, the impact of the technique variable can be unveiled blocking by target size. When the target size was medium and small, significant differences could be found among these techniques. SQUAD was significantly more accurate than ray-casting ($p < .001$) and EyeSQUAD ($p < .0167$). EyeSQUAD was significantly more accurate than ray-casting when target was medium or small size ($p < .001$). However, EyeSQUAD was less accurate than SQUAD regardless of target size because SQUAD yielded lower error rate also when target was large ($p < .0167$). This also contradicts H4, revealing that SQUAD outperformed EyeSQUAD with respect to precision in all target sizes. When target was large, although ray-casting also had higher error rate than the SQUAD ($p < .001$), there was no significance found between ray-casting and EyeSQUAD ($p = .143$).

The effect of the interaction of technique and distractor density ($F_{0.95;4,72} = 3.745, p < .0167, power = .854$) was found significant on overall. Examining the interaction of technique and distractor density when blocking by distractor density, the Bonferroni test shows that ray-casting had significantly higher error rate than both SQUAD ($p < .001$) and EyeSQUAD ($p < .001$), and SQUAD was significantly more accurate than EyeSQUAD ($p < .0167$) for any distractor density.

When blocking by technique, there was no significant effect found with either ray-casting or SQUAD. With EyeSQUAD, the sparse distractor density yielded significantly lower error rate than medium distractor density ($p < .0167$), although no other significant difference was found. This suggests that errors are more possible to yield when increasing the number of refinement steps from two to three during selection with EyeSQUAD. In other words, the quad-menu selection process also had an effect on the errors which contributes potential amount of system errors.

A constant amount of error rate could be observed for EyeSQUAD from the interaction between technique and target size blocked by technique (**Figure 7**). This constant amount of error rate was mainly caused by system aspects such as losing accuracy of eye tracking with time proceeds, by human aspects such as losing attention, or by design aspects such as distractions on the quad-menu. For instance, the accuracy of EyeSQUAD had been greatly improved from the pilot study to the actual experiment (17.4–6.2%) due to usability improvement mainly caused by carefully removing some distractions on the quad-menu (i.e., moving the distribution of objects away from the margin of the quadrants on quad-menu to prevent the calculated point-of-regard from shaking around margins).

Finally, under our experimental settings, no significance was found in the interaction of technique, target size and distractor density ($F_{0.95;8,144} = 0.935, p = .492, power = .412$). No other significant differences could be detected on the error rate.

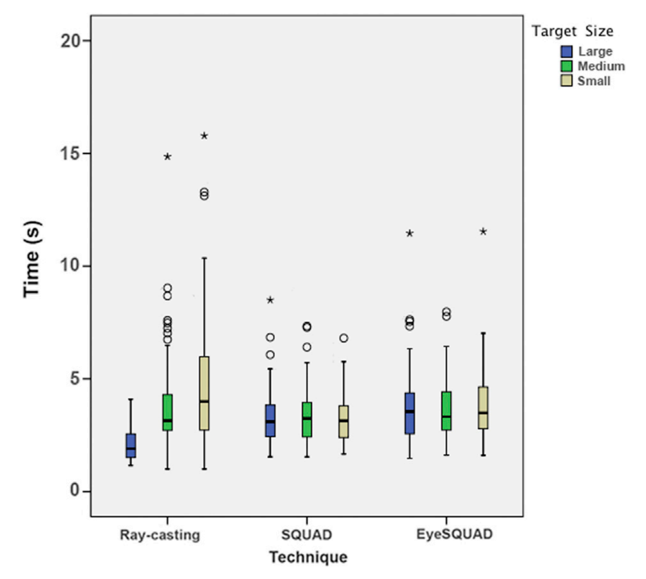

FIGURE 8 | Interaction between technique and target size on time.

We checked whether there was a significant order effect in the study which attributed to the completely within-subject experiment design. As a between-subjects variable, the order was found not significant in the study on the error ($F_{0.95;5,18} = 1.915, p = .165, power = .451$).

5.2 Average Selection Time

Overall, target size ($F_{0.95;2,36} = 35.768, p < .001, power > .999$) and distractor density ($F_{0.95;2,36} = 67.810, p < .001, power > .999\delta$) had significant effects, compared with no significance of technique ($F_{0.95;2,36} = 2.404, p = .112, power = .437$). These suggest the effect of target size on the selection time of ray-casting and the effect of distractor density on the speed of progressive refinement techniques were so significant which were still significant even averaged on overall.

Delving into the pairwise comparisons of techniques first, the Bonferroni test suggests that EyeSQUAD had significantly lower selection speed than SQUAD ($p < .0167$) although no significant difference was observed either between ray-casting and SQUAD ($p = .781$), or between ray-casting and EyeSQUAD ($p = 1.000$).

For the target size, the small target size had significantly longer selection time than the large target size ($p < .0167$) and the medium target size ($p < .001$) by checking the pairwise comparisons with the Bonferroni test. Besides, the medium target size had significant longer selection time than the large target size ($p < .0167$).

On overall, the distractor density also had significant effect on time. With increasing the density of distractor in the virtual environment, the selection time was significantly increased. Since sparse density had the lowest selection time compared with medium ($p < .001$) and dense ($p < .001$) densities with dense density being slower than medium density ($p < .001$).

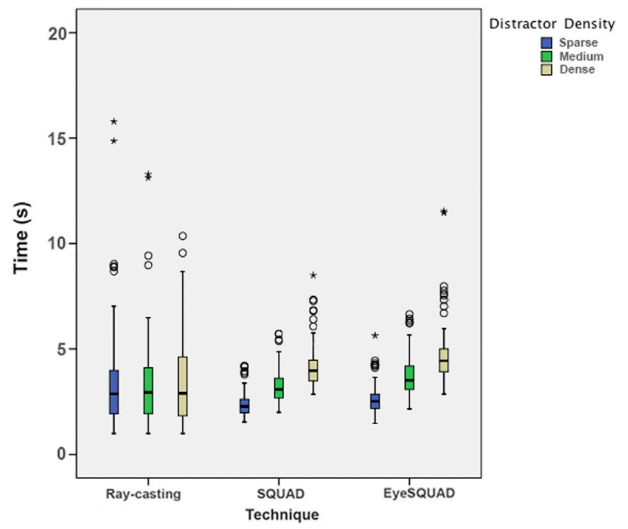

FIGURE 9 | Interaction between technique and distractor density on time.

Examining the interaction of technique and target size ($F_{0.95;4,72} = 34.545, p < .001, power > .999$) yielded a significant effect. First, the impact of target size on the interaction could be revealed when blocking by technique (**Figure 8**). With ray-casting, the large target size was significantly faster than medium target size ($p < .001$) and small target size ($p < .001$). Ray-casting had faster speed when target size was medium than when target was small ($p < .0167$). However, no significance was observed by changing the target size in either SQUAD or EyeSQUAD. These confirm H1.

Furthermore, we evaluated the interaction effect between technique and target size, blocked by target size. Under large target size condition, ray-casting was significantly faster than SQUAD ($p < .001$) and EyeSQUAD ($p < .001$), and SQUAD was faster than EyeSQUAD ($p < .0167$). When target was small, SQUAD was significantly faster than ray-casting ($p < .0167$) while no significant difference was found either between ray-casting and EyeSQUAD ($p = .100$) or between SQUAD and EyeSQUAD ($p = .021$). No significant difference was found when target was medium. This indicates that ray-casting outperformed two other techniques with respect to selection speed only when the target was large, and SQUAD outperformed ray-casting when the target was small. Since SQUAD was faster than EyeSQUAD when target was large, H4 is weakened.

At first glance, the interaction of technique and density ($F_{0.95;4,72} = 15.487, p < .001, power > .999$) also shows a significant effect on time overall. When we delved into details of the interaction, significant differences could be found for SQUAD and EyeSQUAD under different distractor density scenarios when blocking by technique (**Figure 9**). For SQUAD, sparse distractor density yielded significantly less average selection time than medium distractor density ($p < .001$). Also, medium distractor density had significantly less average selection time than dense distractor density ($p < .001$). Similarly, for EyeSQUAD, significant differences could be found among different distractor density conditions (sparse and medium ($p < .001$), medium and dense ($p < .0167$), dense and sparse ($p < .001$)). This coincides with H2 that the selection time of techniques based on progressive refinement (i.e., SQUAD and EyeSQUAD) depends on the density of distractor in the environment which directly related to the number of selection steps. Besides, as expected, the distractor density had no significant effect on selection time when using ray-casting which also supports H2.

Moreover, blocking by distractor density, we can evaluate the importance of technique on its interaction with distractor density. In any distractor density, no significant difference of selection speed could be observed between ray-casting and EyeSQUAD (sparse: $p = .087$, medium: $p = .751$, dense: $p = .139$). However, significance was found when looking into other pairwise comparisons. When the distractor density was sparse, ray-casting was significantly slower than SQUAD ($p < .0167$) although no significance was observed between SQUAD and EyeSQUAD ($p = .032$). Furthermore, SQUAD was significantly faster than EyeSQUAD when distractor density was medium ($p < .0167$) and dense ($p < .0167$). This indicates that significantly more time was spent on the quad-menu refinement process when using the EyeSQUAD compared with SQUAD. Previous results of error demonstrated that the SQUAD was more accurate than EyeSQUAD in any distractor density. The nature of varying the distractor density is actually changing the refinement steps. Hence, the impact of the quad-menu selection process needs to be examined for explaining the differences of performance between EyeSQUAD and SQUAD. Since the essential difference between EyeSQUAD and SQUAD in the quad-menu process is whether using a hand-control metaphor or an eye tracking metaphor, eye tracking technology from both system and human aspects could contribute to the differences in performance.

No significance was found in the interaction of technique, target size and distractor density ($F_{0.95;8,144} = .764, p = .636, power = .335$). However, significance could be observed that ray-casting was faster than SQUAD ($p < .001$) and EyeSQUAD ($p < .001$) when target size was large and distractor density was dense. On the contrary, when target size was small and distractor density was sparse, SQUAD was faster than ray-casting ($p < .0167$) though no significant difference was found between ray-casting and EyeSQUAD ($p = .026$).

Under our experimental settings, no other significance of interactions was found. The order was found also not significant in the study on the time ($F_{0.95;5,18} = 2.426, p = .097, power = .558$).

5.3 User Preference

All 24 participants were asked to complete a post-study survey which was an overall performance questionnaire. Among all participants, 15 participants preferred SQUAD and eight participants favored EyeSQUAD, while only one participant would choose ray-casting if needed to perform additional selection tasks.

All participants rated each technique based on levels of one–7 (1 to be very bad, seven to be very good) right after they had finished all trials of each technique. We performed a one-way repeated measure ANOVA on the ratings. The effect of technique was found significant ($F_{0.95;2,36} = 47.502, p < .001, power > .999$) though the effect of order was found insignificant ($F_{0.95;5,18} = 1.039, p = .425, power = .285$). The mean ratings for ray-casting is 2.75 which was lower compared with 5.92 of SQUAD ($p < .001$) and 5.33 of EyeSQUAD ($p < .001$) with the Bonferroni test. However, no significant difference was found between SQUAD and EyeSQUAD ($p = .408$) which suggests that participants had similar favors for SQUAD and EyeSQUAD. These ratings coincide with the above overall preference of techniques which SQUAD led the favors and ray-casting was least preferable while from rating perspective, EyeSQUAD and SQUAD had similar preferences. Collecting oral feedbacks after the experiments, we found that many participants preferred SQUAD instead of EyeSQUAD since they were more familiar with distal pointing tasks. For example, most individuals had been using remote controllers such as a TV controller since childhood. In contrast, eye tracking interaction was novel to all of them. Several participants found that with EyeSQUAD, they could free their hands rather than always holding a controller and positioning it with certain gestures which causing hand and arm fatigue, and hence they preferred EyeSQUAD. Although controlling with eye movements was unfamiliar, many participants said that they were eager to explore eye tracking techniques for not only selection but also more interaction techniques.

6 DISCUSSION

6.1 Error Rates

The results verified H3, with EyeSQUAD outperforming ray-casting in all conditions with respect to error. However, the results contradict H4 since SQUAD had higher accuracy and faster selection speed than EyeSQUAD on overall.

The error rate of ray-casting greatly increased with decreasing the target size and SQUAD almost yields no errors (0.9% error rate), which is consistentwith H5. However, the error rate of EyeSQUAD was not negligible. We can list several reasons why the error rate of EyeSQUAD was unexpectedly high.

First, unlike holding a controller with the hand like with SQUAD, EyeSQUAD requires users to fixate at certain parts of the screen when pressing the button on the controller during the quad-menu refinement process, otherwise a wrong part can be easily selected if the user blinks or looks away during pressing the button. Participants subconsciously looked away from the fixation point when they were aware of incoming visual changes. Results indicate that the performance of EyeSQUAD was weakened with increasing the refinement steps on the quad-menu. Many involuntary eye movements usually can be involved especially when the user loses attention. This problem could be avoided if certain number of frames (e.g., 100 frames) were used ahead of getting selection commands and choosing the quadrant part with highest score rather than the part where the point-of-regard locates immediately when the user is pressing the button. This way, the sensitivity of quad-menu selection could be greatly lowered, providing a more user-friendly experience.

Secondly, apart from the design aspect, human aspects such as amblyopia (lazy eyes) and dominant eyes (Porac and Coren, 1976) also need to be taken into consideration when implementing eye tracking techniques. EyeSQUAD currently takes equal weights for data of left eye and right eye to calculate the point-of-regard that mapped in three-dimensional space. Redistributing the weights for left eye and right eye according to user's dominance of eyes (amblyopia would be the extreme case) could be a good choice for enhancing stability and accuracy of EyeSQUAD interactions.

Thirdly, the eye tracking stability and accuracy from the hardware could be potential factors that limited the performance of EyeSQUAD. Since we noticed several participants encountered different extents of inaccuracy of eye tracking especially after running the device for a period of time even without any movements of the headset. This could be caused by the weight of the FOVE headset mounted with a Vive tracker with total weight of 820 g. These aspects should be further checked before we are able to make definitive conclusions.

6.2 Selection Time

Our first hypothesis was verified, as the selection time of both SQUAD and EyeSQUAD was found not affected by the target size while ray-casting was slower with small targets and faster with large targets. The study results also verified H2, with the selection time of ray-casting being independent of the distractor density while the time of target selection with SQUAD or EyeSQUAD was proportional to the number of distractors.

With respect to selection time, our fourth hypothesis (H4) was not verified. Contrary to our prediction, SQUAD was verified to be faster than EyeSquad. Our original hypothesis is that the eye-tracking based technique would be faster, as gaze moves faster than the hand. However, we believe technical limitations of the hardware we were using hindered the speed of selection.

We also posit that human human and design factors may have played an unforseen role in the selection time with EyeSQUAD. Participants may need to be more careful when using EyeSQUAD, which could cause longer selection time than we expected. We believe the selection time in EyeSQUAD can also be improved if more attention is paid to these aspects especially the error-tolerant rate of the technique encapsulated in the design which can narrow down the barriers of using the technique while at the same time improving performance.

6.3 General Discussion

Although not a focus of the study, the Design of EyeSQUAD allows for the accurate eye-based selection with low required precision of single, sparse targets. In that case, the selection becomes immediate and the only object inside the sphere gets selected, with no need for refinement phases (similar to what cone-casting would achieve). In situations where only a few targets occur inside the selection sphere, in its current design, EyeSQUAD would still require a single refinement phase, and improvements to its design could be sought to allow for a more fluid selection in those situations.

Our study offered a high level of experimental control, which enabled the objective comparison of the techniques with respect to selection time and error rate. We made a conscious decision of control over ecological validity. We acknowledge that there will be other factors at play in realistic situations, where targets may not be as salient as they were in our study. In these situations, there will necessarily be tasks that will be influenced by variables outside the ones that we studies. For example, cognitive overload can certainly play a role depending on the types of candidate targets in realistic scenarios. We leave these explorations for future research.

All in all, we believe EyeSQUAD to be a marked improvement for usable selection with eye tracking in VR. Although EyeSQUAD could not outperform SQUAD on accuracy or speed, the results were very positive. We have demonstrated that eye tracking with progressive refinement is a viable choice for usable interaction in virtual environments. By carefully paying attention to several aspects such as system, design, human factors and eye tracking technology itself, we believe the performance of EyeSQUAD could be further improved greatly. For example, just by rearranging the display of objects in the quad-menu refinement phases, we were able to reduce the error rates from 17.4% in the pilot study to 6.2% in the formal experiment.

7 LIMITATIONS AND FUTURE WORK

The work presented in this paper is the first attempt at coupling eye tracking with progressive refinement interaction techniques in virtual reality. There are many limitations that should be noted and addressed in future research.

An early decision in the design of EyeSQUAD was to use a hand-controlled button to indicate selection. We understand that this is a major issue, as it ultimately renders the technique as not hands-free. However, the main focus of our research was to investigate eye-based control with progressive refinement, without potential issues with hands-free triggering mechanisms. Future work should certainly investigate usable hands-free means to trigger actions with EyeSQUAD. Probably, eye fixation to trigger selection may not be a good choice due to high latency and the "Midas touch" problem. We have started considering a few options that may prove more usable. For example, a tongue click sound or head nod may be good choices since the movements of eyes and head are decoupled.

As with SQUAD, the out-of-context selection design makes it difficult to select identical objects in different configurations since the spatial information of all candidate objects is ignored once the objects are arranged on the quad-menu. Moreover, scale and other object attributes may be lost once they are on the quad-menu. Candidate targets displayed in refinement phases also results in visual search within each quadrant at every refinement phase, as the new candidate targets distribution is random. This can take a performance toll itself, as with many objects in the initial phase, the search time could be significant. In order to leverage spatial memory of the target location, a future iteration of EyeSQUAD could maintain the spatial relationship among objects in the quad-menu across refinement phases. Although spheres with a distinct target were in the experiment for visual consistency from any viewpoints, these limitations should be considered when implementing EyeSQUAD in realistic situations. Bacim et al. (Bacim et al., 2013) have implemented a number of in-context hand-based progressive refinement techniques. Exploring them with eye-tracking based interaction could lead to even more benefits.

Another limitation deals with system aspects in our experimental environment. The FOVE HMD has reasonably basic eye tracking cameras, which may have led to more difficult calibration and overall lower accuracy. Future work should consider using newer eye-tracking hardware, such as the HTC Vive Pro Eye (2021), which incorporates higher-quality eye tracking using Tobii (2021) technology. EyeSQUAD ran at a relatively low framerate (around 60–80 and 44 Hz in the worst case) as compared to SQUAD and ray-casting (around 80–100 Hz). However unfortunate as an experimental confound, this limitation further proves the case for EyeSQUAD, as even running at a lower frame rate, it had better performance than ray-casting and relatively low error rates. Besides faster computers, running the target approximation in a thread less often than every frame may improve frame rates while not affecting performance, as saccadic movements happen at a much lower rate.

Finally, mismatches between the position of the convergence point from the recorded eye rays and the physical position of point-of-regard would appear after running the system for sometime or immediately after moving the HMD in relation to the head. However, the calculated point-of-regard was accurate and stable when mapped in a 2D plane which was orthogonal to the user (e.g., the quad-menu). Hence, one possible method to optimize the performance of the technique in 3D could be separating the depth control from the technique like the depth ray idea by Vanacken et al. (Vanacken et al., 2007) and determining the depth by other user data such as pupil size and extent of squinting. By this way, the point-of-regard can be cast onto an invisible plane which is orthogonal to the user and is changed with the extent of squinting or pupil size.

8 CONCLUSION

We designed a novel eye tracking selection technique with progressive refinement–Eye-controlled Sphere-casting refined by QUAD-menu (EyeSQUAD). An approximation method was designed to stabilize the calculated point-of-regard and a space partitioning method was used to improve computation. A user study was performed to examine the performance of the EyeSQUAD selection technique in comparison to two previous selection techniques under different target size and distractor density conditions.

EyeSQUAD achieved similar selection speed as ray-casting and SQUAD, and was more accurate than ray-casting when selecting small targets. Even though EyeSQUAD was less accurate than SQUAD, it still showed acceptable time and accuracy performance. Further, more careful design could improve its usability, as evidenced by one iteration of pilot testing, where errors were reduced from a rate of 17.4% to a rate of 6.2%. Additionally, this error rate is lower than previous eye tracking-based selection techniques (Miniotas et al., 2004; Ashmore et al., 2005; Kumar et al., 2007; Khamis et al., 2018). In fact, this is the lowest error rate for eye tracking selection we could find.

In summary, we provided a new selection technique using eye tracking. With selection by progressive refinement, this new technique could obtain better accuracy and precision than standard ray-casting technique when targets were small. Additionally, transferring some workload from hands to eyes, we surprisingly found that similar performance could be achieved via selection with eye movements as manual control even with consumer eye tracking devices. This indicates that implementing eye tracking into human-computer interaction techniques is available and possible to achieve similar performance as typical manual controls.

Finally, there is an incredible potential to eye tracking-based progressive refinement interaction techniques for assistive technologies. In this study we have shown that it is possible to design usable techniques with affordable eye trackers, and the application of these techniques for users with mobility impairments can make VR more inclusive and universal.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the Duke University Campus Institutional Review Board. The participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

YW contributed to the design of EyeSQUAD, implemented the software, conducted the user study, performed data analysis and wrote the manuscript. RK conceptualized the research, contributed to the design of EyeSQUAD, designed the user study, assisted with data analysis and revised the manuscript.

ACKNOWLEDGMENTS

The authors wish to thank all participants in the experiment for their voluntary help for the study. We also want to thank Missy Cummings and her PhD student Haibei Zhu for their kind help of statistical analysis.

REFERENCES

Ashmore, M., Duchowski, A. T., and Shoemaker, G. (2005). "Efficient Eye Pointing with a Fisheye Lens," in Proceedings of Graphics interface 2005 (Victoria, Canada: Citeseer), 203–210.

Bacim, F., Kopper, R., and Bowman, D. A. (2013). Design and Evaluation of 3D Selection Techniques Based on Progressive Refinement. *Int. J. Human-Computer Stud.* 71, 785–802. doi:10.1016/j.ijhcs.2013.03.003

Birch, J. (1997). Efficiency of the Ishihara Test for Identifying Red-green Colour Deficiency. *Oph Phys. Opt.* 17, 403–408. doi:10.1111/j.1475-1313.1997.tb00072.x

Cournia, N., Smith, J. D., and Duchowski, A. T. (2003). "Gaze- vs. Hand-Based Pointing in Virtual Environments," in CHI'03 extended abstracts on Human factors in computing systems (Clemson, SC: ACM)), 772–773.

Creed, C. (2016). "Eye Gaze Interaction for Supporting Creative Work with Disabled Artists," in Proceedings of the 30th International BCS Human Computer Interaction Conference: Companion Volume, Bournemouth, UK (Birmingham, United Kingdom: BCS Learning & Development Ltd.), 38.

Dalmaijer, E. S., Mathôt, S., and Van der Stigchel, S. (2014). PyGaze: An Open-Source, Cross-Platform Toolbox for Minimal-Effort Programming of Eyetracking Experiments. *Behav. Res.* 46, 913–921. doi:10.3758/s13428-013-0422-2

De Haan, G., Koutek, M., and Post, F. H. (2005). "Intenselect: Using Dynamic Object Rating for Assisting 3D Object Selection," in IPT/EGVE, (Delft, Netherlands: Citeseer), 201–209.

Deubel, H., and Schneider, W. X. (1996). Saccade Target Selection and Object Recognition: Evidence for a Common Attentional Mechanism. *Vis. Res.* 36, 1827–1837. doi:10.1016/0042-6989(95)00294-4

Dishart, D. C., and Land, M. F. (1998). "The Development of the Eye Movement Strategies of Learner Drivers," in *Eye Guidance in reading and Scene Perception* (Nottingham, UK: Elsevier), 419–429. doi:10.1016/b978-008043361-5/50020-1

Duchowski, A. T. (2007). "Eye Tracking Methodology," Clemson, SC, in *Theory and Practice*, 328.

Dv, J., Saluja, K. S., and Biswas, P. (2018). "Gaze Controlled Interface for Limited Mobility Environment," in Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, Baltimore, MD, 319–322.

Field, A. (2013). *Discovering Statistics Using IBM SPSS Statistics*. Brighton, England: sage.

Findlay, J. M. (1982). Global Visual Processing for Saccadic Eye Movements. *Vis. Res.* 22, 1033–1045. doi:10.1016/0042-6989(82)90040-2

Fitts, P. M. (1954). The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *J. Exp. Psychol.* 47, 381–391. doi:10.1037/h0055392

FOVE Inc. (2018). Eye Tracking Virtual Reality Headset. Available at: https://www.getfove.com/ Accessed August 30, 2018.

Frees, S., Kessler, G. D., and Kay, E. (2007). PRISM Interaction for Enhancing Control in Immersive Virtual Environments. *ACM Trans. Comput.-Hum. Interact.* 14, 2. doi:10.1145/1229855.1229857

Fuhl, W., Tonsen, M., Bulling, A., and Kasneci, E. (2016). Pupil Detection for Head-Mounted Eye Tracking in the Wild: an Evaluation of the State of the Art. *Machine Vis. Appl.* 27, 1275–1288. doi:10.1007/s00138-016-0776-4

Goldberg, J. H., and Wichansky, A. M. (2003). "Eye Tracking in Usability Evaluation," in *The Mind's Eye* (Redwood Shores, CA: Elsevier), 493–516. doi:10.1016/b978-044451020-4/50027-x

Guenter, B., Finch, M., Drucker, S., Tan, D., and Snyder, J. (2012). Foveated 3D Graphics. *ACM Trans. Graphics (Tog)* 31, 164. doi:10.1145/2366145.2366183

HTC Vive Pro Eye (2021). Available at: https://enterprise.vive.com/us/product/vive-pro-eye-office/ Accessed April 1, 2021).

Iacono, W. G., Peloquin, L. J., Lumry, A. E., Valentine, R. H., and Tuason, V. B. (1982). Eye Tracking in Patients with Unipolar and Bipolar Affective Disorders in Remission. *J. Abnormal Psychol.* 91, 35–44. doi:10.1037/0021-843x.91.1.35

Jacob, R. J. K., and Karn, K. S. (2009). "Eye Tracking in Human-Computer Interaction and Usability Research," in *The Mind's Eye* (Berlin, Heidelberg: Springer), 573–605. doi:10.1016/b978-044451020-4/50031-1

Khamis, M., Oechsner, C., Alt, F., and Bulling, A. (2018). "VRpursuits: Interaction in Virtual Reality Using Smooth Pursuit Eye Movements," in Proceedings of the 2018 International Conference on Advanced Visual Interfaces, Castiglione della Pescaia Grosseto, Italy, 1–8.

Khushaba, R. N., Wise, C., Kodagoda, S., Louviere, J., Kahn, B. E., and Townsend, C. (2013). Consumer Neuroscience: Assessing the Brain Response to Marketing Stimuli Using Electroencephalogram (Eeg) and Eye Tracking. *Expert Syst. Appl.* 40, 3803–3812. doi:10.1016/j.eswa.2012.12.095

Kopper, R., Bacim, F., and Bowman, D. A. (2011). "Rapid and Accurate 3D Selection by Progressive Refinement," in 3D User Interfaces (3DUI), 2011 IEEE Symposium on, Singapore, (IEEE), 67–74.

Kopper, R., Bowman, D. A., Silva, M. G., and McMahan, R. P. (2010). A Human Motor Behavior Model for Distal Pointing Tasks. *Int. J. human-computer Stud.* 68, 603–615. doi:10.1016/j.ijhcs.2010.05.001

Kumar, M., Paepcke, A., and Winograd, T. (2007). "EyePoint: Practical Pointing and Selection Using Gaze and Keyboard," in Proceedings of the SIGCHI conference on Human factors in computing systems, Montréal, Canada, 421–430.

LaViola, J. J., Jr, Kruijff, E., McMahan, R. P., Bowman, D., and Poupyrev, I. P. (2017). *3D User Interfaces: Theory and Practice*. New York, NY: Addison-Wesley Professional.

Majaranta, P., Ahola, U.-K., and Špakov, O. (2009). "Fast Gaze Typing with an Adjustable Dwell Time," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, 357–360.

Mardanbegi, D., Mayer, B., Pfeuffer, K., Jalaliniya, S., Gellersen, H., and Perzl, A. (2019). "Eyeseethrough: Unifying Tool Selection and Application in Virtual Environments," in 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (Osaka, Japan: IEEE), 474–483.

Meagher, D. J. (1980). *Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*. Troy, NY: Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory).

Meena, Y. K., Cecotti, H., Wong-Lin, K., and Prasad, G. (2017). "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, Jeju Island, Korea (Ireland, UK: EMBC)), 905–908. doi:10.1109/EMBC.2017.8036971

Meena, Y. K., Cecotti, H., Wong-Lin, K., and Prasad, G. (2016). "A Novel Multimodal Gaze-Controlled Hindi Virtual Keyboard for Disabled Users," in 2016 IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary (Ireland, UK: SMC)), 003688–003693. doi:10.1109/SMC.2016.7844807

Mine, M. R. (1995). *Virtual Environment Interaction Techniques*. Chapel Hill, NC: UNC Chapel Hill CS Dept.

Miniotas, D., and Špakov, O. (2004). An Algorithm to Counteract Eye Jitter in Gaze-Controlled Interfaces. *Aalborg, Denmark, Inf. Techn. Control.* 30.

Miniotas, D., Špakov, O., and MacKenzie, I. S. (2004). "Eye Gaze Interaction with Expanding Targets," in CHI'04 extended abstracts on Human factors in computing systems 1255–1258.

Patney, A., Salvi, M., Kim, J., Kaplanyan, A., Wyman, C., Benty, N., et al. (2016). Towards Foveated Rendering for Gaze-Tracked Virtual Reality. *ACM Trans. Graphics (Tog)* 35, 179. doi:10.1145/2980179.2980246

Pfeiffer, T. (2008). "Towards Gaze Interaction in Immersive Virtual Reality: Evaluation of a Monocular Eye Tracking Set-Up," in Virtuelle und Erweiterte Realität-Fünfter Workshop der GI-Fachgruppe VR/AR.

Pfeuffer, K., Geiger, M. J., Prange, S., Mecke, L., Buschek, D., and Alt, F. (2019). "Behavioural Biometrics in Vr: Identifying People from Body Motion and Relations in Virtual Reality," in Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 1–12.

Pfeuffer, K., Mayer, B., Mardanbegi, D., and Gellersen, H. (2017). "Gaze+ Pinch Interaction in Virtual Reality," in Proceedings of the 5th Symposium on Spatial User Interaction, Brighton United Kingdom, 99–108.

Pfeuffer, K., Mecke, L., Delgado Rodriguez, S., Hassib, M., Maier, H., and Alt, F. (2020). "Empirical Evaluation of Gaze-Enhanced Menus in Virtual Reality," in 26th ACM Symposium on Virtual Reality Software and Technology, 1–11.

Piumsomboon, T., Lee, G., Lindeman, R. W., and Billinghurst, M. (2017). "Exploring Natural Eye-Gaze-Based Interaction for Immersive Virtual Reality," in 3D User Interfaces (3DUI) 2017 IEEE Symposium on (San Jose, CA: IEEE), 36–39.

Poole, A., and Ball, L. J. (2006). Eye Tracking in HCI and Usability Research. *Encyclopedia Hum. Comput. interaction* 1, 211–219. doi:10.4018/978-1-59140-562-7.ch034

Porac, C., and Coren, S. (1976). The Dominant Eye. *Psychol. Bull.* 83, 880–897. doi:10.1037/0033-2909.83.5.880

Poupyrev, I., Ichikawa, T., Weghorst, S., and Billinghurst, M. (1998). "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques," in *Computer Graphics Forum* (Seattle, WA: Wiley Online Library)), 17, 41–52. doi:10.1111/1467-8659.00252

Rivu, R., Abdrabou, Y., Pfeuffer, K., Esteves, A., Meitner, S., and Alt, F. (2020). "Stare: Gaze-Assisted Face-To-Face Communication in Augmented Reality," in ACM Symposium on Eye Tracking Research and Applications, Stuttgart, Germany, 1–5.

Robinson, D. A. (1964). The Mechanics of Human Saccadic Eye Movement. *J. Physiol.* 174, 245–264. doi:10.1113/jphysiol.1964.sp007485

Sidenmark, L., and Gellersen, H. (2019). "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, New Orleans, LO, 1161–1174.

Sidenmark, L., Gellersen, H., Zhang, X., Phu, J., and Gellersen, H. (2020). "Eye, Head and Torso Coordination during Gaze Shifts in Virtual Reality," in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu HI, 1–40. doi:10.1145/3361218

Smith, J. D., and Graham, T. (2006). "Use of Eye Movements for Video Game Control," in Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology, Hollywood, CA (Kingston, Canada: ACM)), 20.

Stellmach, S., and Dachselt, R. (2012). "Look & Touch: Gaze-Supported Target Acquisition," in Proceedings of the SIGCHI conference on human factors in computing systems, Austin Texas, 2981–2990.

Tanriverdi, V., and Jacob, R. J. (2000). "Interacting with Eye Movements in Virtual Environments," in Proceedings of the SIGCHI conference on Human Factors in Computing Systems (Medford, MA: ACM), 265–272.

Tobii (2021). Available at: https://tobii.com/ (Accessed April 1, 2021).

Vanacken, L., Grossman, T., and Coninx, K. (2007). "Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments," in 3D User Interfaces 2007 3DUI'07 IEEE Symposium on (Charlotte, NC: IEEE).

Wedel, M., and Pieters, R. (2008). "A Review of Eye-Tracking Research in Marketing," in *Review of Marketing Research* (College Park, MD: Emerald Group Publishing Limited), 123–147. doi:10.1108/s1548-6435(2008)0000004009

Zhai, S., Morimoto, C., and Ihde, S. (1999). "Manual and Gaze Input Cascaded (MAGIC) Pointing," in Proceedings of the SIGCHI conference on Human Factors in Computing Systems, The Hague, Netherlands (San Jose, CA: ACM)), 246–253.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.