GazeRayCursor: Facilitating Virtual Reality Target Selection by Blending Gaze and Controller Raycasting

Di Laura Chen
Reality Labs Research, Meta Inc.
Toronto, ON, Canada
chendi@dgp.toronto.edu

Marcello Giordano
Reality Labs Research, Meta Inc.
Toronto, ON, Canada
marcello@marcellogiordano.ca

Hrvoje Benko
Reality Labs Research, Meta Inc.
Redmond, WA, USA
benko@meta.com

Tovi Grossman
University of Toronto
Toronto, ON, Canada
tovi@dgp.toronto.edu

Stephanie Santosa
Facebook Reality Labs
Toronto, ON, Canada
ssantosa@meta.com

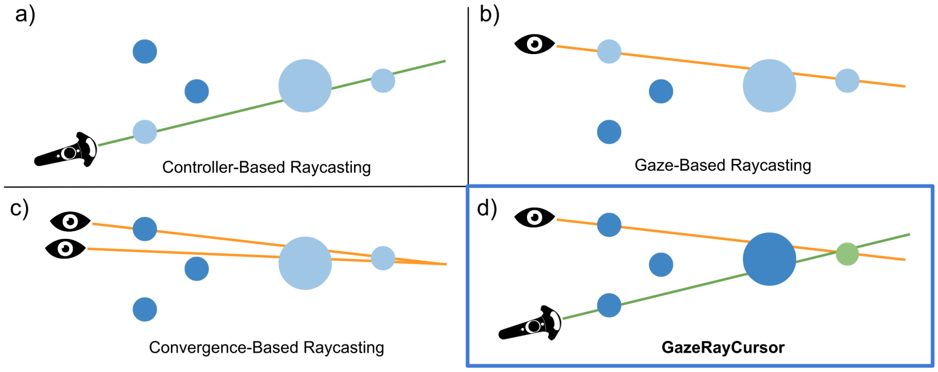

Figure 1: An illustration of various controller and gaze ray selection methods. Dark blue is the object's default colour, light blue indicates a candidate object is highlighted, and green indicates the object is selected. a) using only a controller ray results in intersection ambiguity with multiple objects; b) similarly, using only a gaze ray can cause object selection ambiguity at various depths; c) the convergence of the gaze rays from each eye can help resolve ambiguity, but can be inaccurate due to the angle between the eyes being small because the eyes are close together; d) we propose the GazeRayCursor, which uses the intersection of the gaze and controller rays to disambiguate selection, since the angle between the rays is sufficiently large.

ABSTRACT

Raycasting is a common method for target selection in virtual reality (VR). However, it results in selection ambiguity whenever a ray intersects multiple targets that are located at different depths. To resolve these ambiguities, we estimate object depth by projecting the closest intersection between the gaze and controller rays onto the controller ray. An evaluation of this method found that it significantly outperformed a previous eye convergence depth estimation technique. Based on these results, we developed GazeRayCursor, a novel selection technique that enhances Raycasting, by leveraging gaze for object depth estimation. In a second study, we compared two variations of GazeRayCursor with RayCursor, a recent technique developed for a similar purpose, in a dense target environment. The results indicated that GazeRayCursor decreased selection time by 45.0% and reduced manual depth adjustments by a factor of 10 in a dense target environment. Our findings showed that GazeRayCursor is an effective method for target disambiguation in VR selection without incurring extra effort.

CCS CONCEPTS

• **Human-centered computing** → **Pointing**; **Virtual reality**; *User studies.*

KEYWORDS

object selection; disambiguation; raycasting; gaze; controller; VR

ACM Reference Format:
Di Laura Chen, Marcello Giordano, Hrvoje Benko, Tovi Grossman, and Stephanie Santosa. 2023. GazeRayCursor: Facilitating Virtual Reality Target Selection by Blending Gaze and Controller Raycasting. In *29th ACM Symposium on Virtual Reality Software and Technology (VRST 2023), October 9–11, 2023, Christchurch, New Zealand.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3611659.3615693

VRST 2023, October 9–11, 2023, Christchurch, New Zealand

ACM ISBN 979-8-4007-0328-7/23/10…$15.00
https://doi.org/10.1145/3611659.3615693

1 INTRODUCTION

Object selection is one of the most essential interactions in 3D virtual reality (VR) environments. Raycasting has been shown to be a commonplace technique for pointing and selection tasks in such environments [18, 28]. However, the selection of objects via Raycasting can be ambiguous if the target object is in an environment that is densely packed with other objects. In this case, these objects, which are often placed at varying depths, overlap each other and cause the ray to intersect with several objects at once. Various disambiguation techniques have been proposed to enhance Raycasting-based selection, but these techniques often require the use of a manual mode to adjust the depth of the ray [3, 35] or require one to first select a general area and then identify the target of interest within this area through additional input mechanisms [8, 19]. Although effective, these approaches introduce extra refinement steps to the disambiguation process.

When selecting an object in VR, it is often inevitable for one's eye gaze to fall upon or near an object of interest [33], thus making gaze a natural complement to the capabilities of other input modalities. Previous research has leveraged eye gaze to aid in selection tasks, for example, by analyzing gaze and hand motions to predict a user's intention [5], or by combining eye pointing and head pointing to achieve faster and more accurate target selection [15].

In this paper, we investigate the fusion of eye gaze and handheld controller Raycasting to estimate the depth of objects in VR to resolve ambiguity during target selection. When only using a controller ray (Fig. 1a) or a gaze ray (Fig. 1b) ambiguity arises whenever objects appear at different depths. One method to estimate the depth of the intended object would be to utilize the convergence point between the gaze rays from each eye [13], but this may be inaccurate because the eyes are so close together (Fig. 1c). We propose the GazeRayCursor, which combines the controller ray and gaze ray, and projects the closest intersection point between these rays onto the controller ray to infer the depth of an object (Fig. 1d). As the controller ray and gaze ray are much farther apart, there is a larger angle from which to calculate the intersection. Motivated by this notion, this research sought to answer two research questions:

(1) How accurately can gaze be used to estimate object depth in VR, by intersecting the gaze and controller rays?
(2) How well can this form of depth estimation be used to facilitate target selection in VR?

To answer these research questions, we conducted two studies. The first study compared the use of different subsets of the controller ray, left eye gaze, and right eye gaze, to estimate the depth of VR objects. We found that using the gaze ray produced from the average of the left and right gaze rays together with the controller ray achieved significantly less depth estimation error than using the convergence of the left and right gaze rays. These results motivated the development of GazeRayCursor, a novel technique for 3D object selection based on Raycasting. We evaluated our technique against RayCursor [3], a recent technique effective at disambiguating target depth, in a dense target environment where ambiguity resolution would be needed. We tested two variations of GazeRayCursor, one automatic and one semi-automatic, that has an additional mode to allow users to manually adjust the depth of the intended selection [3]. The results showed that both variations of GazeRayCursor significantly reduced selection time, with the semi-automatic variation also reducing the number of manual adjustments that were needed compared to RayCursor. The findings suggest that GazeRayCursor offers a viable method for target disambiguation during VR selection, without introducing new explicit input modalities or additional steps.

2 RELATED WORK

2.1 Gaze-Assisted Selection

Research in gaze-based interactions has long shown that gaze input can be utilised in various ways to achieve effective target selection and navigation (e.g., [6, 23, 30, 34, 37, 46]. Apart from gaze-only interactions, gaze input has also been widely explored as a natural modality to supplement other modalities in selection tasks. With MAGIC pointing, eye gaze was used with manual computer input by warping the cursor to the eye gaze area where a target resided [48]. Gaze has also been used together with hand and pinch gestures in VR and AR for target selection and object manipulation [31, 33, 47]. Lystbæk et al. explored aligning both gaze and hand rays to activate selection of an object, which was useful in both text entry [24] and AR menu selection [25]. A Fitt's Law study on Gaze-Hand Alignment further confirmed that gaze-assisted selection outperformed hands-only methods [42]. Note that our proposed GazeRayCursor differentiates from the Gaze-Hand Alignment techniques [25, 42] in that the gaze cursor is not explicitly shown to the user, and gaze is not used for target pre-selection. Rather, gaze acts as an implicit modality that complements controller ray selection and assists in target disambiguation. With Pinpointing, eye gaze movements were used together with head movements to achieve precise target selection in AR [17]. Other research has focused on combining gaze with touch input from touch surfaces or handheld devices [32, 41]. This prior research has shown that gaze is a suitable modality to be used in combination with other input to assist in object selection and manipulation and has the potential to reduce physical effort and achieve greater accuracy, naturalness, and speed compared to only using one modality.

Apart from being a valuable input modality for object acquisition, eye gaze has also been used for depth estimation in 3D. One method proposed by Mlot et al. used the vergence, or the simultaneous turning motions of the eyeballs towards or away from each other, to calculate depth [13]. In vergence movements, the eyes converge to point to the same object such that the closer the object is, the more the eyes rotate towards each other. Thus, the idea behind using vergence for depth estimation was to cross the gaze rays from both eyes and use the intersection to determine the 3D gaze position. Sidenmark et al. developed a Vergence Matching technique that correlated relative eye vergence movements with object depth changes to infer a user's attention towards very small targets [38]. Other works have trained models using vergence and depth measures to improve depth estimates [20, 43]. Kwon et al. used the Pupil Center Distance between eyes to find an object's depth because this distance increases when an object is far from the eye and decreases when an object is near the eye [16]. Another method used the vestibulo-ocular reflex (VOR), a reflex that stabilizes gaze during head movement, to estimate object depth [26, 27].

While the above techniques are promising, they also demonstrate that substantial effort is needed for the convergence point to be used to estimate depth, since the left and right gaze vectors can be close to parallel. We were inspired by this prior research on using convergence to estimate object depth, but set out to investigate whether adding the gaze modality to the traditional controller input would improve depth estimations and object selection.

2.2 Target Facilitation Techniques

Target facilitation is a fundamental task for object selection and manipulation, thus various approaches have been developed to improve facilitation. One of the earliest examples of pointing and target facilitation was Raycasting, which resembled pointing with a "laser gun" in real life [21]. Since then, Raycasting has evolved into a comprehensive family of interaction techniques [1]. Some notable examples of Raycasting-based pointing and targeting techniques include the Depth Ray and Lock Ray, both of which used an adjustable depth marker to select the closest intersected object [12]. The iSith technique used two selection rays that originated from the left and right hands to calculate the shortest line connecting the two rays when they were within a close range of each other [45]. The midpoint generated from the shortest line then enabled users to grab an intersected object. Another ray-based technique, Flexible Pointing, bent the selection ray to enable a user to reach obstructed objects [9]. Target facilitation has also been supported by predicting the endpoint of the ray. For example, Henrikson et al. proposed a head-coupled kinematic template matching technique to predict the endpoint of ray pointing in VR [14].

Other techniques employed different shapes as selection tools instead of using a traditional ray. For instance, the BubbleCursor automatically expanded a circular shape to enclose the closest object to its center [11, 22]. On the other hand, Aperture Selection employed a cone with an adjustable apex angle as its method of selection [10]. Among many of the target facilitation techniques, the selection of the object nearest the cursor was the linchpin that made the techniques successful. The idea was thus used within GazeRayCursor to highlight the object closest to the calculated intersection.

2.3 3D Target Disambiguation

Target disambiguation in 3D is often needed for target facilitation. Various disambiguation techniques have been proposed for Raycasting, since it is one of the most popular techniques for 3D object selection [1]. Often, these techniques use a manual mode where a user manipulates a slider on a mobile phone, or a touchpad on a VR controller to adjust the depth of the ray to select a target [3, 35]. Previous research has identified the eye-hand visibility mismatch [2], where objects that are visible from the user's eye position may be occluded from the user's hand position, and objects that are selectable from the hand position may appear occluded from the eye position. In our GazeRayCursor technique, the gaze and controller rays complement each other to overcome this mismatch.

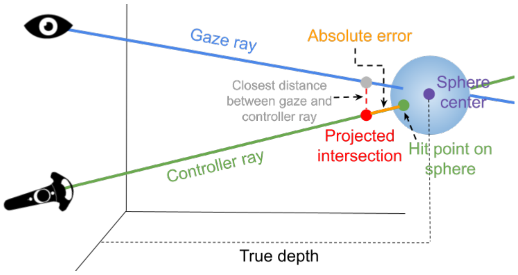

Figure 2: An illustration of the target sphere, the gaze and controller rays in the VR environment, and factors related to the estimated depth calculation.

Coarse-to-fine grained disambiguation techniques have also been proposed in the literature. Usually, this type of approach involves first selecting a general area with multiple objects, and then decluttering that area to identify the object of interest. Decluttering spreads out the dense area of objects so that the target object is no longer occluded and is thus easier to select. Deng et al. introduced a gaze probe for this decluttering process [8], while Grossman and Balakrishnan used a Flower Ray to spread out objects in a cluttered area in a similar way [12]. Lee et al. developed a way to acquire a subspace of interest, instead of an object of interest [19].

Using eye gaze for disambiguation has also been explored, since eye gaze imposes little additional physical and mental effort on the user. For example, Outline Pursuits identified potential targets by head gaze pointing or controller pointing, then disambiguated them using eye gaze to follow the motion around the outline of the intended target [39]. In DualGaze, the user selected an object using a two-step gaze gesture [29]. When the user's gaze fell upon the intended target, a confirmation flag popped up and required the user to gaze back at it to confirm the selection. Target ambiguity has also been resolved through depth estimations during gaze interactions [27]. Although prior disambiguation techniques have proven to be effective, they usually introduce extra refinement steps to the process. In this present research, we reduce the need for a refinement step by combining the advantages of eye gaze tracking and controller Raycasting.

3 STUDY 1 - GAZE FOR DEPTH ESTIMATION

The purpose of this first study was to explore different fusions of gaze and controller rays that could be used to infer object depth in VR. As it is unlikely for two rays to intersect at exactly the same point in 3D space, we developed the idea of the *projected intersection*, i.e., the point on a controller ray that is closest to a gaze ray (Fig. 2). The projection is done on the controller ray rather than the gaze ray because the user can manipulate the controller ray with greater precision. The projected intersection can be found by calculating the shortest line that connects the two rays. This intersection point could then be used to measure the accuracy of an object's estimated depth by comparing the distance between the projected intersection and the object (absolute error). Herein we describe the evaluation of four methods for object depth estimation.

3.1 Participants

We recruited 12 participants (10 male, 2 female; μ = 33 years, range = 18 to 54 years,) with normal or corrected-to-normal vision. All participants were right-handed. Participants were asked to perform a dominant eye test before the study, and 11 participants had the right eye as their dominant eye. When asked how they would describe themselves as a VR user, 7 participants indicated they were beginners, 3 were intermediate, and 2 were experts. Participants were compensated $50 CAD for their time.

3.2 Apparatus

The study was conducted using an HTC Vive Pro Eye head-mounted display (HMD), with a 110 degree field of view (FOV), a refresh rate of 90 Hz, and a resolution of 1440 × 1600 pixels per eye. The gaze data was output from the Vive's eye tracking at 120 GHz. An HTC Vive controller was used for input. The experiment was developed in Unity3D and ran on a laptop with a 5.1 GHz Intel Core i7-10875H processor and a NVIDIA GeForce RTX 2080 Super graphics card.

3.3 Task

The experiment consisted of a pointing task in VR. During each trial, a white target sphere was rendered in the VR environment. The sphere was rendered at a different target depth, size, and angle during each trial. The participant's task was to select the target by pointing the controller ray at the target while focusing their gaze directly on or near the sphere. The target became highlighted in yellow and selectable only when the controller ray passed through it and the eye gaze ray fell within a gaze focus boundary around the sphere. The boundary was 1.8 times the radius of the sphere to allow for some inaccuracy in the gaze direction. The controller ray and gaze ray had to be directed at the target for at least 0.3 seconds before the target became selectable to prevent the participant clicking at random. When the target became highlighted, the participant selected it by clicking on the controller's trigger button. A 0.5 second delay was enforced between each trial to allow for some recovery time before showing the next target. It is important to note that gaze was not used to facilitate selection in this study, it was only incorporated to allow us to test the accuracy of the depth estimation techniques described in section 3.7.

3.4 Procedure

Participants were first instructed to stand in an area that enabled stable tracking by the Vive base stations. The system recorded the participant's initial position and created a virtual foot stand for positional reference. A training module was then shown to explain the task and participants completed 10 training trials to become familiar with the task. Throughout the study, text instructions were displayed in VR to help the participant complete the task.

The study then consisted of 6 blocks, with 225 trials per block. Participants could take a break in between each block if needed. Eye tracking was calibrated at the start of each block. The study lasted approximately 60 minutes.

3.5 Experiment Design

We used a repeated-measures, within-subject design. The position of the target sphere in each trial varied based on 3 independent variables - depth, size, and angle. Depth determined the distance between the origin to the sphere center along the z-axis. We used 25 depths, ranging from 0.5 meters to 50 meters (i.e., 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 15, 20, 25, 30, 50). We refer to this depth as the true depth for the rest of the paper, to distinguish it from the estimated depth. Size was determined by the angular diameter of the sphere, which described how large the sphere appeared from the HMD's point of view (Fig. 3). For a single angular diameter, the farther the object, the larger its size. The object would appear in the HMD as the same size even at different depths. Three angular diameters were evaluated (i.e., 4, 8, and 12 degrees). Angle was defined as the angle between the HMD's forward vector and the sphere's center. The sphere was placed at 3 angles (i.e., -20, 0, 20 degrees) in each of the x- and y-directions, for a total of 9 positions (Fig. 3).

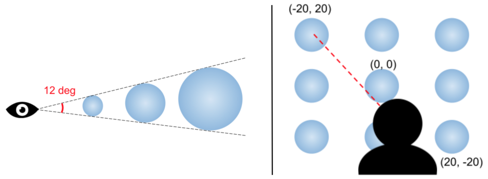

Figure 3: Left: The angular diameter determined how large a sphere appeared from the HMD's point of view. Right: The x- and y-direction angles where the target object was positioned.

The presentation order of the depths, sizes, and angles were randomized, however we ensured that no two consecutive trials would have the sphere appearing at the same angle. Each target depth, size, and angle appeared in the study two times, resulting in a total of 1350 trials per participant (i.e., 25 x 3 x 9 x 2 = 1350 trials).

3.6 Data Pre-Processing

Before performing the analysis, we calculated the task completion time (TCT) for each trial, and found that the mean TCT across all participants was 1.24 seconds (SD = 0.20). Based on this, we considered trials with a TCT of more than 10 seconds as outliers, and removed them from the analysis. We also removed trials with more than 1 error click. In total, 1.41% of trials were removed. The analysis was performed using the remaining data.

3.7 Depth Estimation Techniques

We evaluated four depth estimation techniques that combined subsets of the left gaze ray, the right gaze ray, and the controller ray. Note that these estimations were only calculated in the background for post-experiment analysis, and did not directly affect the experimental task itself.

LGC (Left Gaze and Controller) calculated the projected intersection of the left eye gaze ray and the controller ray (Fig. 4a). Similarly, *RGC* (Right Gaze and Controller) calculated the projected intersection of the right eye gaze ray and the controller ray (Fig. 4b). These two methods were evaluated to see how well a single gaze ray could calculate depth, and if using the left versus right gaze made any difference.

CGC (Combined Gaze and Controller) calculated the average of the left and right gaze rays, combined them into a single ray originating from between the two eyes, and found the projected intersection of the combined gaze ray and controller ray (Fig. 4c).

ConvGC (Convergence Gaze and Controller) is similar to existing vergence techniques that estimate depth through eye gaze convergence (e.g., [13]). We implemented the most basic form of vergence, without additional processes like feature extractions or regression model training. We considered ConvGC as the baseline to compare

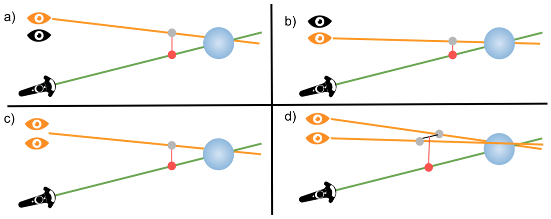

Figure 4: Study 1 depth estimation techniques: a) LGC - Left Gaze and Controller; b) RGC - Right Gaze and Controller; c) CGC - Combined Gaze and Controller; d) ConvGC - Convergence Gaze and Controller. The red dot represents the projected intersection on the controller ray.

our other methods to. The convergence point was first found by calculating the midpoint of the shortest line between the left and right gaze rays. This point was then projected onto the controller ray (Fig. 4d). The additional step to project the convergence point was added to harmonize the controller-gaze-based methods in terms of projected intersections to enable a fair comparison and consistent error metric.

3.8 Results: Technique Comparison

In this section we first report the results on absolute error and percentage error for all techniques across different depths.

3.8.1 Depth Absolute Error. As described above, the depth estimation techniques used the projected intersection on the controller ray. The absolute error was calculated as the distance (in meters) between the projected intersection and the controller ray hit point on the sphere (Fig. 2). The overall mean absolute errors for the LGC, RGC, CGC, and ConvGC techniques were 5.07 meters, 5.34 meters, 5.04 meters, and 7.50 meters respectively. This meant that the technique with the lowest mean absolute error (CGC) saw a 32.8% improvement over the baseline (ConvGC).

The RM-ANOVA found that there was a significant effect of technique on depth absolute error ($F_{4,44} = 75.888$, $p < .0001$) across aggregations of depth, size, and angle factors. Post-hoc Fisher LSD tests reported that ConvGC had significantly larger absolute errors than the other techniques. No significant differences were found between other techniques.

3.8.2 Percentage Error. We also computed the percentage error of the estimated depth from the true depth to compare trends across different depths. Percentage error was calculated by dividing the depth absolute error by the true depth. The percentage error grew rapidly as the true depth increased, and true depths beyond 4 meters had more than 20% percentage error for all techniques (Fig. 5). However, the percentage errors for close objects were much lower. For example, the percentage error for CGC at 0.5 meters of true depth was 5.58%. Fig. 5 Right shows a zoomed in view of percentage errors for true depths closer to the user.

3.8.3 Summary. Overall, CGC achieved the lowest error among all techniques, consistently across each of the experiment independent variables. Although CGC, LGC, and RGC revealed no significant differences, CGC was deemed to be a more balanced choice among the three, as it leveraged both the left and right gaze rays, consistent with how gaze was used in prior works on gaze-assisted selection. Thus, we elected to use CGC as our technique for depth estimation, and explore detailed performance considerations further in the next section.

Table 1: The effect of angle in the x- and y-direction on mean absolute error using CGC, ordered from smallest to largest absolute error.

x-angle (deg)	y-angle (deg)	Absolute Error (m)
0	0	4.471
-20	0	4.593
20	0	4.756
20	20	4.788
-20	20	4.934
0	20	5.181
-20	-20	5.488
0	-20	5.498
20	-20	5.639

3.9 Results: Additional Analysis of CGC

In this section we provide some additional insights on the performance of the CGC depth estimation technique.

3.9.1 Effect of Size and Angle. The effect of size on absolute error was significant ($F_{2,22} = 6.602$, $p < .01$). Post-hoc tests showed that the absolute error for a sphere with angular diameter of 4° ($\mu = 4.51$ meters) was significantly lower than that of 8° ($\mu = 5.17$ meters) and 12° ($\mu = 5.43$ meters). A possible explanation for this is that a smaller sphere forced the eye gaze to focus on a more restricted area on the sphere, which improved the projected intersection used to estimate depth.

A main effect was also found for angle ($F_{8,88} = 2.311$, $p < .05$; Table 1). Post-hoc tests showed that (0, 0) and (-20, 0) were significantly different than (-20, -20), (0, -20), and (20, -20). The center position from the participant's point of view, (0, 0), achieved the lowest error. The y-direction angles seemed to form a pattern, where the lowest three errors occurred when the y-angle was 0° (i.e., the target was in the middle row), and the highest three errors occurred when the y-angle was $-20°$ (i.e., the target was at the bottom row). This may be due to eye tracking accuracy deteriorating when targets are further into the periphery [4, 7, 36].

3.9.2 Effect of True Depth. As the true depth increased, the estimated depth error also increased significantly (Fig. 5). This can be explained by looking at the gaze-controller angle, i.e. the angle between the combined gaze ray and the controller ray's forward vectors. This angle was large when the object was close, but eventually reached a plateau when the object was placed farther away (Fig. 6 Left). There was a significant effect of true depth on gaze-controller angle ($F_{24,264} = 376.685$, $p < .0001$). As objects increase in depth, the angle becomes very small. This would make depth estimations using projected intersections difficult, as a small angle would indicate that the gaze and controller rays would be almost parallel to each other.

We also examined if the error in depth estimation was due to overshooting or undershooting. Overshooting meant that the projected intersection was behind the target sphere, so the estimated

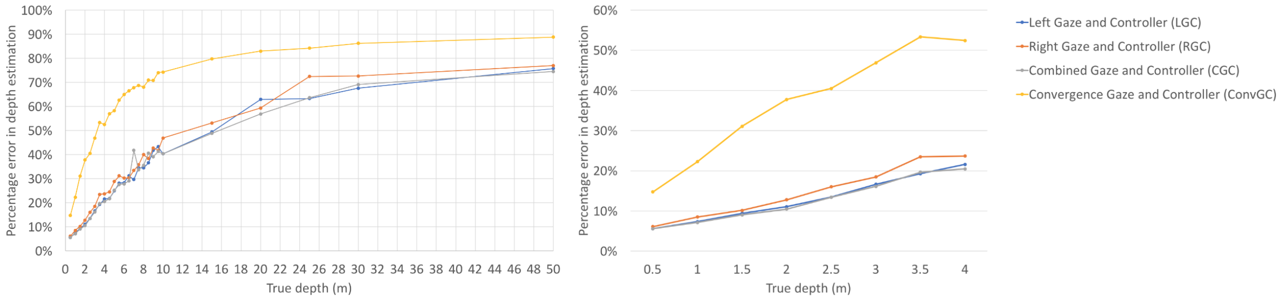

Figure 5: Left: The mean depth estimation percentage error for each technique with respect to all true depths (m). Right: The mean depth estimation percentage error for true depths less than or equal to 4 meters.

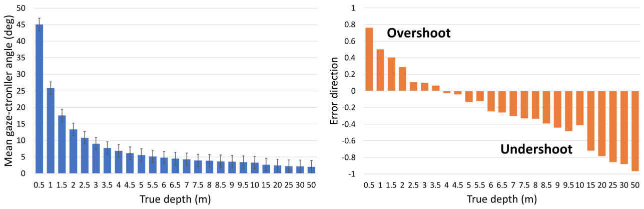

Figure 6: Left: The mean angles between the combined gaze ray and controller ray with respect to the true depth (m). Right: The mean error directions with respect to the true depth (m). A positive sign indicated that more overshooting occurred at that particular depth, whereas a negative sign indicated that more undershooting occurred.

depth would be larger than the actual depth. On the other hand, undershooting meant that the project intersection was in front of the target sphere, so the estimated depth would be smaller than the actual depth. We define an Error Direction metric, with values of 1 for an overshoot and -1 for an undershoot (Fig. 6 Right). For true depths smaller than 4 meters, overshooting occurred more often than undershooting. For true depths larger than 4 meters, the opposite occurred. This implies that for objects farther away, one's eye gaze perceived the object to be closer than it actually was.

3.9.3 Summary. Overall, the projected intersection from the combined gaze ray and controller ray (CGC) shows promise for inferring target depth, especially at target depths less than 4 meters. However, our analysis does indicate that this technique is still error prone, with percentage error rates ranging from 6%-20% depending on the object depth. Furthermore, the size and the location of the object may further impact the accuracy. As such, it is important to consider how object facilitation techniques can be designed to best utilize the CGC technique. This is explored in our next study with the design and evaluation of the GazeRayCursor.

4 STUDY 2 - SELECTION FACILITATION

When multiple objects are in a 3D space, a controller ray could pass through more than one object at once, creating ambiguity during object selection. From the first study, we found that the combined eye gaze with controller ray can be used to infer the depth of the object the user intends to select, and found that this depth estimation technique performed better than using eye convergence. However, questions remain as to how to best integrate this form of depth estimation into a target selection technique. We developed a selection technique using this concept called GazeRayCursor, and evaluated two variations of it, GazeRayCursorAuto and GazeRayCursorSemi, compared to the RayCursor technique [3], one of the most recent and best performing techniques for resolving ambiguity during Raycasting target selection. As all three techniques behave identically when only a single target is intersected, the study was performed in a dense target environment, to specifically test the techniques in situations when ambiguity resolution would be needed.

4.1 Participants

Study 2 was conducted with 12 participants (10 male, 2 female; μ = 26 years, range = 18 to 34 years; 11 right-handed, 11 right-eye dominant) with normal or corrected-to-normal vision. None of the participants from Study 1 took part in Study 2. Nine participants described themselves as beginner VR users, 2 as intermediate, and 1 as an expert. The experiment took approximately 90 minutes and each participant was provided with $150 USD for their time.

4.2 Apparatus

The same apparatus from Study 1 was used in Study 2.

4.3 Selection Techniques

Three selection techniques were evaluated during our study. The baseline technique, RayCursor, is an effective pointing facilitation technique that was recently introduced by Baloup et al. [3]. RayCursor moves a cursor along a controller ray via thumb swipes on a controller's touchpad, highlighting the closest object to the cursor. Baloup et al. also presented a variation of this technique, called the semi-auto RayCursor (Fig. 7c,d), that continually highlighted an object until the cursor was closer to another intersected object, at which point it would snap to that object. When the ray intersected more than one object, the object closest to the user would be highlighted. Semi-auto RayCursor also supported RayCursor's manual adjustment feature, where a user could manipulate the cursor position using the controller touchpad if desired. During manual adjustments, the cursor turned red, and if the participant's thumb left the touchpad for more than one second, the cursor returned to its default behaviour. To control the sensitivity of the swipes in manual mode, we implemented the VitLerp transfer function as described in RayCursor [3], with parameters k1 = 1, k2 = 5, v1 = 1.5,

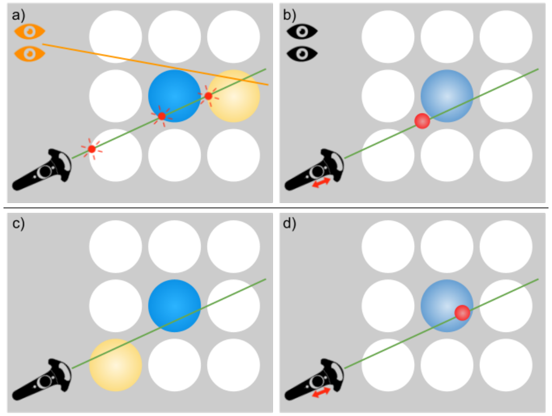

Figure 7: Study 2 selection techniques: (a) GazeRayCursorAuto calculated the projected intersection and highlighted the closest object to it. (a-b) GazeRayCursorSemi was identical to GazeRayCursorAuto but with an additional manual mode. (c) With RayCursor, when the ray intersected multiple objects, the closest target was highlighted and (d) participants could manually adjust the cursor to select a desired target.

and v2 = 10. These parameters were tweaked to suit our experimental world's scale. As semi-auto RayCursor significantly improved error rates [3], we decided to implement the semi-auto RayCursor as the baseline for our experiment and refer to it as *RayCursor*.

The second technique, *GazeRayCursorAuto*, was directly derived from the CGC depth estimation method used in Study 1, as it achieved the best results. In this technique, when the controller ray intersects multiple objects in the environment, the object whose hit point is closest to the projected intersection is highlighted. During Study 1, we saw that larger objects had lower depth estimation accuracy, as there was a less focused region to gaze upon. Thus, for each hit point on an intersected object, a red 3 centimeter circle that flashed at a frequency of 16 Hz was shown at the hit points to help focus gaze on the object and generate more accurate projected intersections (Fig. 7a).

Because the baseline RayCursor technique supports manual adjustments of the depth cursor, we evaluated a third technique, *GazeRayCursorSemi*, which allowed for similar adjustments (Fig. 7a,b). By default, it behaved exactly like GazeRayCursorAuto. However, when the user's thumb swiped on the touchpad, manual adjustments along the controller ray could be made. All parameters for the manual mode in GazeRayCursorSemi were the same as for RayCursor. We evaluated GazeRayCursorSemi to determine how often participants would use the manual mode for target dismabiguation.

4.4 Task

In Study 1, we identified CGC as a promising method to estimate target depth. This first study was conducted in a single target environment, to identify the depth estimation accuracy levels when pointing at known targets. However, since the goal of Study 2 was to examine how well our techniques could resolve ambiguity in target selection, we created a highly dense environment to maximize the likelihood of disambiguation being needed during each trial. It is important to note that all three techniques behave identically when only a single target is intersected. The target environment consisted of 4 layers of uniformly distributed grids of spheres in the VR scene, each at different depths. One solid dark blue sphere served as the target, while the rest were translucent white distractor objects. During each trial, a different sphere was assigned to be the target and the participant's task was to correctly select the target.

When the current selection technique determined that a distractor was the candidate target, it was highlighted translucent yellow to indicate that it was selectable. When the target sphere was the candidate target, it was highlighted solid light blue (Fig. 7). To select the highlighted target, the participant clicked the controller's trigger. If the participant had correctly selected the target sphere, the controller vibrated and the next trial started. If a distractor was selected, it would turn translucent red for 0.5 seconds before changing back to its default colour, and the participant could try again. If the participant did not successfully select the target within 20 seconds, the controller emitted a long vibration and began the next trial. A 0.5 second delay occurred between each trial.

4.5 Procedure

Similar to Study 1, participants wore the HMD and stood in an area that enabled stable tracking. The participant's initial position was recorded and a virtual foot stand was created. A training block was shown before the start of each technique, wherein participants were asked to familiarize themselves with the technique, for a maximum of 30 training trials. The study was divided into 9 blocks, with 3 blocks used for each technique. Participants could take a break between each block if needed. Eye tracking calibration was performed before each block started. After the study, we asked participants to complete a questionnaire to collect subjective feedback.

4.6 Experiment Design

A repeated-measures, within-subject design was employed. The independent variables were the 3 selection techniques and the position of the target within the grid of spheres. Four layers of grids were rendered in the z-direction in uniform intervals. The depth of the first layer was 0.5 meters from the participant's position, and the depth of the last layer was 4 meters. For each grid of spheres in one layer, the grid's length and width were both 1.85 meters. The grid contained 100 uniformly distributed spheres, with 10 spheres in each row and each column of the grid. The diameter of each sphere was 20 centimeters. As all spheres had the same fixed diameter, the angular diameters of the spheres were different at each layer, with the numbers being 22.62°, 6.87°, 4.04°, and 2.86° respectively, from layers 1 to 4. Since the first layer was very close to the participant (i.e., 0.5 meters), the participant could not see all objects in the first grid at once in their FOV. The grids were set up such that all objects in the second layer were within the FOV. The size and density chosen for the spheres ensured that no spheres collided with each other, and that any target behind the first layer would be partially occluded.

For layers 2 to 4, every sphere appeared once as the target, for a total of 300 trials. For layer 1, only the 16 spheres (4 x 4) from the center-most positions of the grid were used once as the target, as these spheres were in the FOV when seen from the initial head position. Thus, 316 trials were used to evaluate each technique,

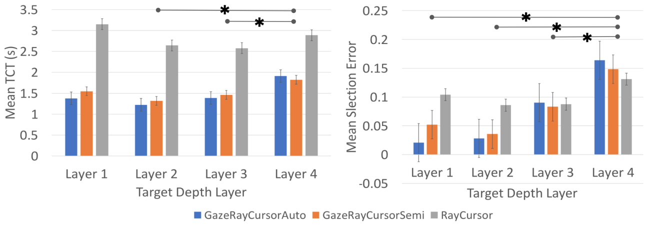

Figure 8: Left: The mean TCT for each technique at each target depth layer. Right: The mean selection error for each technique at each target depth layer. Asterisks indicate significantly different pairs of depth layers.

resulting in 948 trials per participant. The target position during each trial was randomized. The order of the selection techniques was counterbalanced across participants.

4.7 Results and Discussion

In total, 11 trials were skipped (5 GazeRayCursorAuto, 6 RayCursor; 0.09% of all trials) due to the 20 second timeout threshold elapsing. The remaining data was used in the analysis.

4.7.1 Task Completion Time. We were interested in the task completion time (TCT) for each selection technique, and the effect of target depth on the TCT. The TCT is defined as the time it took the participant to select the target from when the target was first shown. The target depth was defined as the layer that the target was positioned within amongst the 4 layers of grids.

The RM-ANOVA revealed that TCT differed significantly based on selection technique ($F_{2,22} = 61.246$, $p < .0001$). Post-hoc tests revealed that RayCursor ($\mu = 2.73$ seconds) was significantly slower than GazeRayCursorAuto ($\mu = 1.50$ seconds) and GazeRayCursorSemi ($\mu = 1.53$ seconds; Fig. 8 Left). GazeRayCursorAuto decreased selection time by 45.0% compared to RayCursor. There was no significant difference between GazeRayCursorAuto and GazeRayCursorSemi. This is a promising result, showing that the manual adjustment option may not be needed for the technique to perform well.

Target depth was also found to have a significant effect on TCT ($F_{3,33} = 5.678$, $p < .005$). Post-hoc tests showed significant differences between layers 2 (1.73 seconds) and 4 (2.21 seconds), and layers 3 (1.81 seconds) and 4 (2.21 seconds). Targets located on layers 2, 3, and 4 were rendered behind distractor objects, and the post-hoc results showed that targets on these layers took more time to select. The interaction effect between selection technique and target depth was also found to be significant ($F_{6,66} = 2.531$, $p < .05$). RayCursor had the highest TCT on depth layer 1 (3.15 seconds), and showed significant differences from layers 2 (2.64 seconds) and 3 (2.58 seconds). On the other hand, GazeRayCursorAuto and GazeRayCursorSemi had no significant differences between layers 1, 2, and 3. For GazeRayCursorAuto, TCT was significant lower on layers 1, 2, and 3 (1.38, 1.22, 1.39 seconds, respectively), compared to layer 4 (1.91 seconds). For GazeRayCursorSemi, only layer 2 (1.32 seconds) and layer 4 (1.82 seconds) were significantly different. It was surprising that targets on layer 1, which were not occluded, had higher TCTs than targets on some farther depth layers. A possible explanation could be that the 0.5 meter depth was much closer to the user compared to other depths, thus requiring participants to take more time and make larger head movements to find and focus on the target. In any case, GazeRayCursorAuto and GazeRayCursorSemi achieved significantly better TCT compared to RayCursor, regardless of depth.

4.7.2 Selection Error Rate. A selection error occurred whenever a participant selected an incorrect distractor object at least once before the target object was selected. The RM-ANOVA revealed no significant main effect of technique on selection error rate (p = 0.2), however target depth did have a significant main effect on selection error ($F_{3,33} = 19.550$, $p < .0001$) (Fig. 8 Right). Post-hoc analysis found that the mean selection error for depth layer 4 ($\mu = 0.148$) was significantly greater than layers 1, 2, and 3 ($\mu = 0.059$, 0.050, 0.087, respectively). No other significant differences were found between layers. The interaction effect between technique and target depth was found to be significant ($F_{6,66} = 2.931$, $p < .05$). Fig. 8 Right shows the mean selection error for each technique at every target depth layer. At layers 1 and 2, GazeRayCursorAuto and GazeRayCursorSemi had significantly lower selection errors than RayCursor. The selection error for all 3 techniques were similar at layer 3. However, at layer 4, the selection error for the two GazeRayCursor techniques overtook RayCursor. Thus, although GazeRayCursorAuto and GazeRayCursorSemi had faster selection times across all depth layers, it was more difficult to select the correct target when the target was farther away. This is consistent with the findings from Study 1, since targets farther away were harder for one's eye gaze to pinpoint.

4.7.3 Manual Adjustments. The manual mode was available with RayCursor and GazeRayCursorSemi to evaluate how often participants would use it to disambiguate between intersected objects. On average, participants used manual mode 96.4% of the time while using RayCursor, and 9.4% of the time while using GazeRayCursorSemi. All participants triggered manual mode using RayCursor, while 4 participants did not trigger manual mode at all using GazeRayCursorSemi. A high trigger rate was expected with RayCursor, since the target was heavily obstructed by a dense number of distractor objects and RayCursor's default Raycasting highlighted the closest object to the participant. In these cases, to select the target, the participant would need to manually adjust the cursor to move behind the distractors blocking the target. The trigger rate for GazeRayCursorSemi was much lower, which again indicates that the GazeRayCursor technique by itself was often sufficient for selecting the desired target without manual intervention.

We also computed the distance between the target and the manual cursor when manual mode was first entered, as well as the time spent in manual mode, to gauge the effort needed to select a target using these techniques. The mean manual distance and time for participants who used manual mode at least once was 2.36 meters (SD = 1.26) and 2.06 seconds (SD = 0.56) for GazeRayCursorSemi, while for RayCursor it was 4.22 meters (SD = 0.92) and 2.00 seconds (SD = 0.40). Although the time spent in manual mode was similar for the two techniques, GazeRayCursorSemi required a smaller distance for the manual cursor to reach the target sphere. This matched our expectations, because the majority of the times the goal target was obstructed by distractor objects in front of it. This meant that using RayCursor, when manual mode was entered, the starting position of the manual cursor would be near objects in layer 1. Using GazeRayCursorSemi, the initial manual cursor position was dependent on the last position of the projected intersection, which would be closer to the goal target since participants would try to focus their gaze near the target before entering manual mode. The smaller manual adjustment distance traveled in GazeRayCursorSemi potentially leads to less physical effort and fatigue.

4.7.4 Subjective Feedback. During the post-study questionnaire, we collected participants' preferences through several 5-point Likert scale questions and short answer comments. Participants rated each technique on how easy it was to select the goal target when it was directly in front of them, and when it was behind other distractor objects (1 - hard, 5 - easy). We conducted Friedman tests on the ratings with corrections for ties, and found a significant effect of technique on ease of selection when the target was at the front ($\chi^2(2) = 15.077, p < .001$), and also when the target was behind other distractors ($\chi^2(2) = 6.541, p < .05$). Post-hoc tests reported that when selecting targets in front, GazeRayCursorAuto (median = 5) and GazeRayCursorSemi (median = 5) showed significant differences compared to RayCursor (median = 4). When selecting targets behind, GazeRayCursorSemi (median = 5) was found to be significantly easier to use compared to RayCursor (median = 3). No significant differences were found between GazeRayCursorAuto (median = 4) and other techniques.

We also asked participants to rate their preference on using each technique to select objects in the experiment. Overall, technique had a significant effect over preference ($\chi^2(2) = 8.773, p < .05$). GazeRayCursorAuto (median = 5) and GazeRayCursorSemi (median = 4) were significantly preferred over RayCursor (median = 2). Preferences between GazeRayCursorAuto and GazeRayCursorSemi were not significant.

Participants who preferred GazeRayCursorAuto explained that the gaze selection was accurate and fast enough to select the target and not having a manual option forced them to commit more to gaze without adding additional thumb strain. Participants who preferred GazeRayCursorSemi mentioned that for closer targets, it was easier to select the target automatically and for targets that were farther away, manual selection allowed for more precision. Participants who had less preference for RayCursor explained that it required the most effort to achieve the desired selection, and most targets demanded manual adjustment. One participant who identified GazeRayCursorAuto as their least preferred technique did not like that they did not have control over the technique when their gaze could not accurately detect distant objects. Having the manual mode helped them compensate for gaze focus estimation issues.

5 LIMITATIONS AND FUTURE WORK

Our results show substantial promise for gaze-based depth estimation, and for two variations of the GazeRayCursor, our proposed target facilitation technique. However, there are several limitations to the results from this research which warrant further discussion.

First, some participants reported difficulty with manipulating the controller ray when selecting distant targets, which was likely due to distant selections requiring more precise and subtle manipulations of the ray. This behavior led to false selections and accidentally moving away from the correct target even after the controller ray was positioned correctly. Sometimes the trigger button also caused shaking due to the force required to press the trigger. This phenomenon, called the Heisenberg Effect of Spatial Interaction, is a well-known side effect when using spatially tracked input devices for pointing and selection tasks in AR and VR [44]. Future iterations could leverage conic rays (e.g., [21, 40]) or alternative button locations.

Secondly, our studies were conducted in abstract environments, and for the second study, the target and distractor objects were set at a fixed size, which was pre-determined to be easy to see within all layers of the grid. Although participants found gaze-based selection easy to use, further work is needed to explore the technique within real VR application scenarios. Potential application scenarios involving dense environments or selection ambiguity may include placing multiple objects in a room for interior design, or selecting buildings on a map during urban planning.

Finally, we tested GazeRayCursor against the baseline in relatively close range in a dense environment because the goal was to see how well GazeRayCursor could resolve target selection ambiguity. Although GazeRayCursor was shown to be especially effective in a highly dense environment, we speculate that it would achieve similar performance in a less dense environment, since the mechanics of the technique would not change. It is worth mentioning that at further depths or lower density, the baseline RayCursor may also reach better performance than how it performed in our Study 2 setup. In fact, all three techniques should behave equivalently when only a single target is intersected. As such, the results from our study should be interpreted with care, and not be overgeneralized to sparse target environments. Future works could be conducted to validate further increasing depths and environments with less ambiguity. Nevertheless, the results demonstrated that GazeRayCursor is a promising selection technique that does not require separate input devices or extra steps during the selection process, with substantial benefits over state-of-the-art techniques when target ambiguity is present.

6 CONCLUSION

We proposed GazeRayCursor, a novel VR selection technique. We conducted two studies and saw the benefits of using combined gaze ray and controller ray intersections for target selection in dense environments. The CGC technique achieved a significantly lower error rate compared to the baseline ConvGC that relied solely on gaze convergence, and the resulting GazeRayCursor developed from CGC was an improvement over Raycasting or gaze convergence techniques. Both the automatic and semi-automatic variations of GazeRayCursor demonstrated faster selection times compared to RayCursor, and selection time was not affected by the availability of the manual mode. We conclude that the semi-automatic variation may be the best technique going forward, as the manual option would give users more control when gaze cannot precisely pinpoint objects farther away. GazeRayCursor has demonstrated high effectiveness for target disambiguation in dense target environments, and its use of natural and implicit modalities does not incur extra effort during selection, making it a promising approach to enhance interaction experiences in VR.

REFERENCES

[1] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

[2] Ferran Argelaguet, Carlos Andujar, and Ramon Trueba. 2008. Overcoming Eye-Hand Visibility Mismatch in 3D Pointing Selection. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology* (Bordeaux, France) *(VRST '08)*. Association for Computing Machinery, New York, NY, USA, 43–46. https://doi.org/10.1145/1450579.1450588

[3] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. RayCursor: A 3D Pointing Facilitation Technique Based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300331

[4] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of Eye-Gaze over Head-Gaze-Based Selection in Virtual and Augmented Reality under Varying Field of Views. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 9 pages. https://doi.org/10.1145/3206343.3206349

[5] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. 2017. Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3718–3728. https://doi.org/10.1145/3025453.3025753

[6] Myungguen Choi, Daisuke Sakamoto, and Tetsuo Ono. 2020. Bubble Gaze Cursor + Bubble Gaze Lens: Applying Area Cursor Technique to Eye-Gaze Interface. In *ACM Symposium on Eye Tracking Research and Applications* (Stuttgart, Germany) *(ETRA '20 Full Papers)*. Association for Computing Machinery, New York, NY, USA, Article 11, 10 pages. https://doi.org/10.1145/3379155.3391322

[7] Viviane Clay, Peter König, and Sabine U. König. 2019. Eye tracking in virtual reality. *Journal of Eye Movement Research* 12, 1 (Apr. 2019). https://doi.org/10.16910/jemr.12.1.3

[8] Shujie Deng, Jian Chang, Shi-Min Hu, and Jian Jun Zhang. 2017. Gaze Modulated Disambiguation Technique for Gesture Control in 3D Virtual Objects Selection. In *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*. 1–8. https://doi.org/10.1109/CYBConf.2017.7985779

[9] Alex Olwal Steven Feiner. 2003. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *Proc. UIST*, Vol. 3. 81–82.

[10] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. 1996. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*. 95–96.

[11] Tovi Grossman and Ravin Balakrishnan. 2005. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 281–290. https://doi.org/10.1145/1054972.1055012

[12] Tovi Grossman and Ravin Balakrishnan. 2006. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 3–12. https://doi.org/10.1145/1166253.1166257

[13] Esteban Gutierrez Mlot, Hamed Bahmani, Siegfried Wahl, and Enkelejda Kasneci. 2016. 3D Gaze Estimation Using Eye Vergence. In *Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies* (Rome, Italy) *(BIOSTEC 2016)*. SCITEPRESS - Science and Technology Publications, Lda, Setubal, PRT, 125–131. https://doi.org/10.5220/0005821201250131

[14] Rorik Henrikson, Tovi Grossman, Sean Trowbridge, Daniel Wigdor, and Hrvoje Benko. 2020. Head-Coupled Kinematic Template Matching: A Prediction Model for Ray Pointing in VR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376489

[15] Shahram Jalaliniya, Diako Mardanbegi, and Thomas Pederson. 2015. MAGIC Pointing for Eyewear Computers. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers* (Osaka, Japan) *(ISWC '15)*. Association for Computing Machinery, New York, NY, USA, 155–158. https://doi.org/10.1145/2802083.2802094

[16] Yong-Moo Kwon, Kyeong-Won Jeon, Jeongseok Ki, Qonita Shahab, Sangwoo Jo, and Sung-Kyu Kim. 2006. 3D Gaze Estimation and Interaction to Stereo Display. *IJVR* 5 (01 2006), 41–45.

[17] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173655

[18] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: theory and practice*. Addison-Wesley Professional.

[19] Hyeongmook Lee, Seung-Tak Noh, and Woontack Woo. 2017. TunnelSlice: Freehand Subspace Acquisition Using an Egocentric Tunnel for Wearable Augmented Reality. *IEEE Transactions on Human-Machine Systems* 47, 1 (2017), 128–139. https://doi.org/10.1109/THMS.2016.2611821

[20] Youngho Lee, Choonsung Shin, Alexander Plopski, Yuta Itoh, Thammathip Piumsomboon, Arindam Dey, Gun Lee, Seungwon Kim, and Mark Billinghurst. 2017. Estimating Gaze Depth Using Multi-Layer Perceptron. In *2017 International Symposium on Ubiquitous Virtual Reality (ISUVR)*. 26–29. https://doi.org/10.1109/ISUVR.2017.13

[21] Jiandong Liang and Mark Green. 1994. JDCAD: A highly interactive 3D modeling system. *Computers & Graphics* 18, 4 (1994), 499–506. https://doi.org/10.1016/0097-8493(94)90062-0

[22] Yiqin Lu, Chun Yu, and Yuanchun Shi. 2020. Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 35–43. https://doi.org/10.1109/VR46266.2020.00021

[23] Christof Lutteroth, Moiz Penkar, and Gerald Weber. 2015. Gaze vs. Mouse: A Fast and Accurate Gaze-Only Click Alternative. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 385–394. https://doi.org/10.1145/2807442.2807461

[24] Mathias N. Lystbæk, Ken Pfeuffer, Jens Emil Sloth Grønbæk, and Hans Gellersen. 2022. Exploring Gaze for Assisting Freehand Selection-Based Text Entry in AR. *Proc. ACM Hum.-Comput. Interact.* 6, ETRA, Article 141 (may 2022), 16 pages. https://doi.org/10.1145/3530882

[25] Mathias N. Lystbæk, Peter Rosenberg, Ken Pfeuffer, Jens Emil Grønbæk, and Hans Gellersen. 2022. Gaze-Hand Alignment: Combining Eye Gaze and Mid-Air Pointing for Interacting with Menus in Augmented Reality. *Proc. ACM Hum.-Comput. Interact.* 6, ETRA, Article 145 (may 2022), 18 pages. https://doi.org/10.1145/3530886

[26] Diako Mardanbegi, Christopher Clarke, and Hans Gellersen. 2019. Monocular gaze depth estimation using the vestibulo-ocular reflex. 1–9. https://doi.org/10.1145/3314111.3319822

[27] Diako Mardanbegi, Tobias Langlotz, and Hans Gellersen. 2019. Resolving Target Ambiguity in 3D Gaze Interaction through VOR Depth Estimation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300842

[28] Mark R Mine. 1995. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept* (1995).

[29] Pallavi Mohan, Wooi Boon Goh, Chi-Wing Fu, and Sai-Kit Yeung. 2018. DualGaze: Addressing the Midas Touch Problem in Gaze Mediated VR Interaction. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 79–84. https://doi.org/10.1109/ISMAR-Adjunct.2018.00039

[30] Aunnoy Mutasim, Anil Ufuk Batmaz, Moaaz Hudhud Mughrabi, and Wolfgang Stuerzlinger. 2022. Performance Analysis of Saccades for Primary and Confirmatory Target Selection. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology* (Tsukuba, Japan) *(VRST '22)*. Association for Computing Machinery, New York, NY, USA, Article 18, 12 pages. https://doi.org/10.1145/3562939.3565619

[31] Aunnoy K Mutasim, Anil Ufuk Batmaz, and Wolfgang Stuerzlinger. 2021. 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* (Virtual Event, Germany) *(ETRA '21 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 15, 7 pages. https://doi.org/10.1145/3448018.3457998

[32] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-Touch: Combining Gaze with Multi-Touch for Interaction on the Same Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 509–518. https://doi.org/10.1145/2642918.2647397

[33] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + Pinch Interaction in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/3131277.3132180

[34] Thammathip Piumsomboon, Gun Lee, Robert Lindeman, and Mark Billinghurst. 2017. Exploring natural eye-gaze-based interaction for immersive virtual reality. 36–39. https://doi.org/10.1109/3DUI.2017.7893315

[35] Hyocheol Ro, Seungho Chae, Inhwan Kim, Junghyun Byun, Yoonsik Yang, Yoonjung Park, and Tackdon Han. 2017. A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2873–2878. https://doi.org/10.1109/SMC.2017.8123063

[36] Immo Schuetz, T. Scott Murdison, and Marina Zannoli. 2020. A Psychophysics-Inspired Model of Gaze Selection Performance. In *ACM Symposium on Eye Tracking Research and Applications* (Stuttgart, Germany) *(ETRA '20 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 25, 5 pages. https://doi.org/10.1145/3379156.3391336

[37] Asma Shakil, Christof Lutteroth, and Gerald Weber. 2019. CodeGazer: Making Code Navigation Easy and Natural With Gaze Input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300306

[38] Ludwig Sidenmark, Christopher Clarke, Joshua Newn, Mathias Lystbæk, Ken Pfeuffer, and Hans Gellersen. 2023. Vergence Matching: Inferring Attention to Objects in 3D Environments for Gaze-Assisted Selection. In *2023 ACM CHI Conference on Human Factors in Computing Systems*.

[39] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. 2020. Outline Pursuits: Gaze-Assisted Selection of Occluded Objects in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376438

[40] Anthony Steed and Chris Parker. 2004. 3D selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In *8th international immersive projection technology workshop*, Vol. 2.

[41] Sophie Stellmach and Raimund Dachselt. 2012. Look &amp; Touch: Gaze-Supported Target Acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2981–2990. https://doi.org/10.1145/2207676.2208709

[42] Uta Wagner, Mathias N. Lystbæk, Pavel Manakhov, Jens Emil Sloth Grønbæk, Ken Pfeuffer, and Hans Gellersen. 2023. A Fitts' Law Study of Gaze-Hand Alignment for Selection in 3D User Interfaces. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 252, 15 pages. https://doi.org/10.1145/3544548.3581423

[43] Martin Weier, Thorsten Roth, André Hinkenjann, and Philipp Slusallek. 2018. Predicting the Gaze Depth in Head-Mounted Displays Using Multiple Feature Regression. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research &amp; Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 19, 9 pages. https://doi.org/10.1145/3204493.3204547

[44] Dennis Wolf, Jan Gugenheimer, Marco Combosch, and Enrico Rukzio. 2020. Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376876

[45] H.P. Wyss, R. Blach, and M. Bues. 2006. iSith - Intersection-based Spatial Interaction for Two Hands. In *3D User Interfaces (3DUI'06)*. 59–61. https://doi.org/10.1109/VR.2006.93

[46] Xin Yi, Leping Qiu, Wenjing Tang, Yehan Fan, Hewu Li, and Yuanchun Shi. 2022. DEEP: 3D Gaze Pointing in Virtual Reality Leveraging Eyelid Movement. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 3, 14 pages. https://doi.org/10.1145/3526113.3545673

[47] Difeng Yu, Xueshi Lu, Rongkai Shi, Hai-Ning Liang, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2021. Gaze-Supported 3D Object Manipulation in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 734, 13 pages. https://doi.org/10.1145/3411764.3445343

[48] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 246–253. https://doi.org/10.1145/302979.303053