EZCursorVR: 2D Selection with Virtual Reality Head-Mounted Displays

Adrian Ramcharitar*
Carleton University
Ottawa, Canada

Robert J. Teather†
Carleton University
Ottawa, Canada

ABSTRACT

We present an evaluation of a new selection technique for virtual reality (VR) systems presented on head-mounted displays. The technique, dubbed EZCursorVR, presents a 2D cursor that moves in a head-fixed plane, simulating 2D desktop-like cursor control for VR. The cursor can be controlled by any 2DOF input device, but also works with 3/6DOF devices using appropriate mappings. We conducted an experiment based on ISO 9241-9, comparing the effectiveness of EZCursorVR using a mouse, a joystick in both velocity-control and position-control mappings, a 2D-constrained ray-based technique, a standard 3D ray, and finally selection via head motion. Results indicate that the mouse offered the highest performance in terms of throughput, movement time, and error rate, while the position-control joystick was worst. The 2D-constrained ray-casting technique proved an effective alternative to the mouse when performing selections using EZCursorVR, offering better performance than standard ray-based selection.

Keywords: Virtual Reality, selection, Fitts' law, ISO 9241-9.

Index Terms: Human-centered computing → Virtual Reality • Human centered computing → Pointing

1 INTRODUCTION

Selection is a key element of virtual reality (VR) user interaction. Consider, for example, shooting an enemy in a VR first-person shooter game, or grasping a virtual object presented in a museum exhibit; both tasks involve selection. Selection in VR has traditionally been divided into two (rough) classes of virtual hands (requiring depth precision to grasp an object) and ray-based techniques (requiring remote pointing at a target) [1]. There are numerous selection techniques that have been previously developed for use in VR (see e.g., [2], [3], [11], [7]).

One common selection technique used with devices such as Microsoft's Hololens and various "cardboard" VR[1] displays is to use a ray cast from the head (controlled by head rotation) in lieu of a 3D wand, presenting a cursor fixed in the centre of the screen. However, excessive head motion can yield neck fatigue and can be disorienting to users (as the viewpoint is coupled to the selection ray). In contrast, most modern head-mounted displays (e.g., the Oculus Rift, and HTC Vive) employ tracked wand input devices. While immersive, 6 degree of freedom (6DOF) devices employing typical virtual hand or ray-based selection techniques can be problematic. Depth perception is imprecise leading to inaccuracy with selection methods that require accuracy in depth [6], [30], and latency and jitter remain problems, especially with ray-based techniques [29]. Bowman et al. recommend minimizing the number of DOFs when considering the design of a section device or technique [13]. Furthermore, previous work has shown that 2DOF selection can offer superior performance, even in stereo 3D virtual environments [4], [31].

We note that selection in VR typically involves both the interaction technique itself (i.e., the software part), and the input device (i.e., the hardware part). Example *interaction techniques* include ray-casting, Poupyrev's go-go technique [22], and direct touch with the hand. Common VR *input devices* include wands, such as those provided with the HTC Vive and Oculus Rift, but joysticks (e.g., on game controllers) and even the mouse can be used. Laviola et al. [13] point out that interaction technique and input device are separable – an input device can support multiple different interaction techniques, and vice versa. Consider, for example, that ray-casting (an interaction technique) is supported by both 3D trackers and the mouse (input devices). Likewise, 3D trackers support both ray-casting and direct touch metaphor interaction techniques. Both components are important considerations when performing selections in VR, and it is desirable when designing new interaction techniques that they can work with multiple different input devices. After all, not all users have access to the same equipment.

Based on these observations and our past research [24], we proposed a novel selection technique we call EZCursorVR. EZCursorVR is a 2D head-coupled cursor fixed in the screen plane of the head mounted display (HMD). Unlike stationary cursors in the center of the field of view (as used with Hololens, for example), EZCursorVR can move independently using 2DOF input from any peripheral input device, employing position or rate-control mappings. Several non-VR games such as ArmA[2] use this method of aiming. Unlike most first-person shooter (FPS) games, where the mouse simultaneously controls the cursor and rotates the viewpoint, ArmA decouples these: moving the mouse controls the cursor, and viewpoint rotation begins when the cursor reaches the screen edge. Some Nintendo Wii games (e.g., GoldenEye) use a similar technique, with the remote pointing controller, effectively allowing the player to decouple view direction and selection. This effective style of interaction was our inspiration for EZCursorVR. In addition to supporting any source of 2DOF input, EZCursorVR also allows users to use their head rotation to perform selections, or a combination of both head rotation and 2DOF input.

Following a description of the design of EZCursorVR, we present a user study investigating the effectiveness of the technique with multiple input devices, in comparison to a standard 6DOF ray-based and head-based selection techniques. A secondary objective was to determine which existing 2DOF devices work best with EZCursorVR. To this end, the study included several 2DOF input devices: a mouse, joystick, and a ray-based technique limited to 2DOF control, similar to the Wii's motion controller. The experiment conformed to a previously validated 3D extension [31]

*e-mail: adrian.ramcharitar@carleton.ca
†e-mail: rob.teather@carleton.ca

[1] Including devices that use a smartphone as the display such as Google Cardboard (https://vr.google.com/cardboard/) and Samsung's Gear VR (http://www.samsung.com/global/galaxy/gear-vr/)

[2] https://arma3.com/

of ISO 9421-9 [26] which uses Fitts' law to compare pointing devices [8]. As is typical in 3D Fitts' law evaluations, we compared these selection techniques across several target sizes, distances and depths while measuring movement time, error rate and throughput.

The main hypotheses of our work are:

H1: Performance with 2D techniques will be higher than 3D techniques, as found in prior research [34]

H2: The mouse will perform best, followed by Ray2D, Velocity-Joystick, Head-only, and finally Position-Joystick. This ranking is based on our own pilot testing and intuition, as well as previous studies that used similar input methods [17], [19], [23]

H3**:** Throughput will be consistent across target depth using EZCursorVR , but will vary with depth using the standard ray, as found in previous research [32]

We note that EZCursorVR supports combinations of head and controller movement for selection. We speculate that participants might, for example, use the head to get the cursor in the general vicinity of a target, and the mouse (or other input device) to perform fine-grained positioning. We have included a head-only selection technique (as used with the Hololens, or smartphone-based VR HMDs) to determine if this combination is beneficial.

2 Related Work

2.1 3D Selection Techniques

There is an extensive body of literature on 3D selection techniques, dating back to the 90s. For the sake of brevity, we discuss only key studies here, and refer the reader to Argelaguet and Andujar's comprehensive 3D selection survey [1] and/or Laviola et al. [13, Chapter 7] for a more thorough overview.

Past studies have compared variations of direct touch [15] with ray-based techniques. Traditional ray-based techniques, although the most commonly used technique in commercial VR systems, are susceptible to hand tremor which at far distances and when selecting smaller targets yield high error rates [28]. Several methods to addressed these issues have been proposed such as the bubble cursor[9], [33]and go-go [22], which were designed to support easier selection of remote or small targets by changing the style of the selection cursor. Non-traditional 3D selection techniques such as starfish (which uses a cursor with four branches that lands on nearby targets) are useful for selection in dense environments [35]. However, non-standard techniques may necessitate additional learning. In contrast, EZCursorVR should be easy to understand due to its similarity to desktop interaction – users already have extensive experience with two-dimensional cursors, and can leverage their familiarity.

Previous research has also looked at progressive refinement selection interfaces. Kopper et al proposed a two tier selection process where to user first selects a group of objects, then in multiple steps, refines the selection using a quad divided menu for increased object selection accuracy [12]. They report that this was more accurate at selecting remote objects compared to ray-casting. Similarly, our proposed technique allows combinations of 2DOF input for cursor movement with refinement via head movement (or vice versa). Unlike progressive refinement techniques, this can be done simultaneously rather than dividing the selection process into multiple steps.

Young et al developed an IMU-based input device mounted on the users' arm to enabled 6DOF target selection via virtual hand techniques [36]. Such a device is an attractive option for use with EZCursorVR, since it does not require tethering as it is largely self-contained and does not require an external tracker. Although the results show a lower error rate than optical trackers, throughput was lower and arm fatigue was very high. Fatigue is a major on-going problem with VR controllers[5], [10]. Our goal with EZCursorVR was to design a control scheme that supports the kind of "lazy" interactions envisioned by Mine et al [18], using an approach to minimize physical movements (and hence fatigue) while increasing target selection throughput.

2.2 2D vs 3D Selection

Image-plane interaction is an early example of leveraging the benefits of 2D interaction in 3D spaces [20]. Like our technique, it requires only 2DOF input to select objects, but does so by lining up the hand with objects rather than explicit use of a cursor. We provide a detailed comparison between our technique and image-plane interaction in Section 3.3.

Like previous work [29], [31], [15] our selection task presents targets in a plane. When viewed from the starting position, this essentially "collapses" the 3D selection task into a 2D task [14]. Our selection technique is similar to that of Qian and Teather, who used a 2D eye-controlled cursor that moved within the reference-frame established by head orientation [23]. Eye-based selection was shown to offer worse error rates and throughput than head-based selection. This is likely due to the imprecise and jittery nature of eye saccades. We expect different results, as our implementation used lower jitter controller inputs such as a joystick and mouse. Hence, we expect our results to be more in line with previous comparisons of 2D and 3D selection [34], [32] which revealed 2D techniques outperformed 3D techniques [20], [19].

One issue with using 2D selection cursors in stereo 3D environments is having two cursor images, due to lining up the cursor at one depth with a remote feature at a different depth. This diplopia occurs since the eyes cannot converge to the depth of the cursor and target simultaneously. The result is a "doubling" of either the target or cursor, and has been shown to influence 3D selection, more so when the depth difference between the cursor and target is large [31]. One possible solution is to render the cursor to one eye only, but this may cause eye fatigue [25]. We instead address this by dynamically scaling and resizing the cursor according the target depths such that it always remains the same size and is rendered close to the target to avoid diplopia while also being rendered to both eyes. This approach is recommended by Unity3D tutorials on interaction in VR[3].

2.3 Fitts' Law

Since our study employs Fitts' law, we briefly describe it here. Fitts' law is a predictive model that characterizes performance of selection techniques and pointing devices, revealing the highly linear relationship between task difficulty (*ID* – index of difficulty) and selection time (*MT*). The model is given as:

$$MT = a + b \times ID \quad (1)$$

where $$ID = log_2\left(\frac{D}{W} + 1\right) \quad (2)$$

D is the distance to the target and *W* is the target's size (width), while *a* and *b* are derived via linear regression. This has been formalized as a tool for testing input devices [8, 15] via ISO 9241-9 [26]. Many studies have used the ISO 9241-9 standard for comparing 2D input devices [19], [6]. The standard has also been adapted for use in 3D selection tasks[27], [31]. The standard prescribes the use of throughput (*TP*) as a dependent variable. Throughput is calculated as

$$TP = \frac{ID_e}{MT} \quad (3)$$

[3] https://unity3d.com/learn/tutorials/topics/virtual-reality/interaction-vr

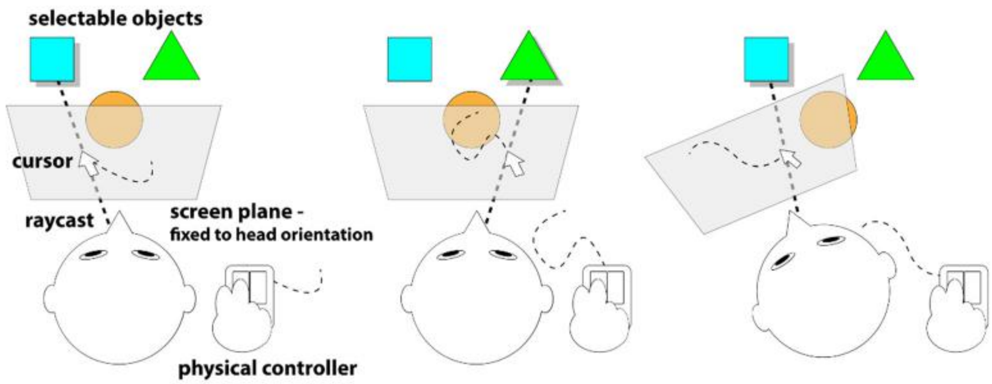

Figure 1: Movement of EZCursorVR. Head-movement and rotation influences the position of the plane-fixed cursor. The cursor can be independently controlled by an external input device (e.g., a mouse, in this example, although other sources of 2DOF or even 6DOF are supported).

As per the ISO 9241-9 standard, effective *ID* (ID_e) is used to calculate throughput as:

$$ID_e = log_2\left(\frac{D_e}{W_e} + 1\right) \quad (4)$$

where $W_e = 4.133 \times SD_x$

D_e is the effective amplitude and W_e is the effective target width. Effective *ID* enables direct comparison between studies with varying error rates, as it adjusts experimental error rate to 4%. The accuracy adjustment is done by calculating SD_x – the standard deviation of over/under-shoot lengths relative to the target centre, projected onto the task axis (the line between subsequent targets). It is multiplied by 4.133, which corresponds to a z-score of ±2.066 in a normal distribution, or 96% of the selection coordinates hitting the target (i.e., a 96% hit rate, or 4% error rate). It also better accounts for the task participants performed, than that which they were presented with.

3 EZCursorVR

Like screen-based techniques [32] EZCursorVR uses ray-casting and relies on the concept of image plane selection [20]. From the user's perspective, they appear to select targets using a 2D cursor to overlap the 2D "screen-space" projection of targets. The plane the cursor resides in appears to be fixed to the head. Rotating or moving the head also results in cursor movement, although the cursor itself appears fixed in this plane. See Figure 1. Unlike classical image-plane interaction [20], where the user can line up their hand with virtual objects for selection, our technique instead does this indirectly via an external controller that controls the cursor position, similar to desktop environments.

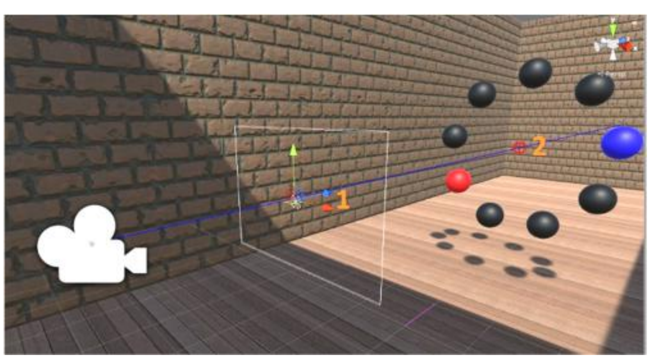

Figure 2: The invisible *control* cursor (#1) that moves in the head-coupled plane, and the visible *rendered* cursor (#2).

In actuality, the *rendered* cursor is displayed in world-space at the intersection point of a ray originating at the head (the camera in Figure 2) and directed towards an invisible *control* cursor that moves in a head-coupled plane (#1 in Figure 2). The control cursor is constrained to move from one extent of the user's field of view to the other. The ray from the head to the control cursor is used to determine which object is selected, and where to position the rendered cursor (#2 in Figure 2).

3.1 Cursor Rendering

Although our intent is to support 2D selection in 3D spaces, simply rendering the control cursor fixed in a head-coupled plane would introduce the double-vision problem detailed earlier [31]. We address this problem by instead displaying the *rendered* cursor (#2 in Figure 2) as an object in the scene. The control cursor is not displayed at all. The rendered cursor is displayed at the correct depth, as determined by ray-casting, using the ray depicted in Figure 2, originating at the eye/head position, and directed through the control cursor. The rendered cursor is drawn at the intersection point with the scene. We then scale the rendered cursor to cancel out the scaling effect of perspective. As a result, the rendered cursor appears consistent in size regardless of its depth. We also render it as a billboard, so it is always oriented towards the viewer. The end result is that the rendered cursor appears to operate in 2D, but its stereo depth is correct for any point in the scene, eliminating double-vision effects [31].

3.2 Input Sources

Since the *control* cursor resides in a plane, 2DOF input sources can readily control its movement through simple mappings. For example, from the default screen-centre position, mouse displacement can map to control cursor displacement (subject to a gain function). Similarly, joysticks can be used in both velocity- and position-control mappings. Changes in the position of the control cursor are reflected in changes to that of the rendered cursor, via ray-casting as described above. Due to cancelling out perspective, the rendered cursor appears to move in 2D, but with correct stereo depth.

For our study, we have also implemented a technique that uses a 6DOF input source to control the cursor. In our case, the user points a tracked wand at the head-coupled plane. The wand-ray/plane intersection point is used for the position of the control cursor. This is similar to the ray-screen technique demonstrated in previous work [32], which in turn, is similar to how remote pointing works with the Nintendo Wii remote.

3.3 Comparison with Image-Plane Selection

Our technique is similar to image-plane selection introduced by Pierce and Forsberg [20]. The 2D plane for our technique is a head-coupled plane that moves along with the user's head rotation to remain parallel to the user's FOV. Our technique most closely resembles the 'Sticky Finger' technique where a user can select objects with an outstretched finger. In contrast, we replace direct interaction with a 2D controlled cursor.

Our technique is different from image-plane selection in two key ways. First, with image-plane selection, the user must outstretch their arms to point at or frame targets. This in-air interaction causes extreme fatigue after extended use, leading to the well-known "gorilla-arm syndrome" [10]. Our technique avoids this by using 2D selection devices, which necessitate less effort and thus reduce fatigue. Second, with image-plane selection, movement is mapped 1:1. In contrast, EZCursorVR offers the ability to apply control-display (CD) gain to cursor movement. While this tends not to be available with 1:1 VR selection techniques (e.g., ray-casting), we argue that gain could help with 2DOF control. Consider, for example, that remote targets perspective scale to be smaller – and in accordance with Fitts' law, harder to select. Remote targets are difficult to select with rays [21], but with EZCursorVR, slow 2D movement (e.g., with a mouse) could be further decelerated by lowering CD gain, enabling precise selection of small targets. Similarly, gain could be increased for long-range ballistic movements, enabling fast crossing of the screen for far away targets. While gain is not explored in our current study, it is a topic for future work.

4 METHODOLOGY

4.1 Participants

Our study included 18 participants (15 male, 3 female, aged 18-44 years) recruited from the local community. We gave participants a pre-test questionnaire asking about their familiarity with VR. Only 6 participants had never had any previous VR exposure.

4.2 Apparatus

4.2.1 Hardware

The experiment was conducted on a VR-ready laptop with an Intel core i7-7700HQ quad core processor, a Nvidia Geforce 1070 GPU, and 16GB of RAM, running Microsoft Windows 10. We used an Oculus Rift CV1 head-mounted display, connected to the computer via HDMI. The CV1 features a resolution of 1080 x 1200 per eye, a 90Hz refresh rate and a 110° field of view. See Figure 3.

Figure 3: Participant wearing the Oculus Rift using the touch controllers. Inset: close-up of Oculus Touch controllers.

Participants were seated far enough away from obstacles to ensure there was no chance of hitting anything. Depending on the experimental condition, participants either used a mouse, an Oculus Touch controller, or the HMD itself as an input device. The Oculus Touch controller (Figure 3) features real-time motion tracking, a thumb joystick, two trigger buttons, and vibrotactile feedback and was used for several different input methods in our experiment.

4.3 Software

Our test environment was created in Unity with external libraries for the Oculus Rift hardware[4]. The test environment was based on ISO 9241-9 reciprocal selection task (Figure 4). Each round consisted of 9 spherical targets, presented in one of three different sizes, at one of three different distances from each other. Each ring of targets was presented at one of three different depths from the user. Within a round, target size, distance, and depth were held constant. Targets were presented in four different colours: blue for the intended target (i.e., the target to select), green for targets that were previously hit, red for targets that were previously missed, and black for targets that were not yet active. The software automatically logged performance data, such as selection times, error rates, and calculated throughput as describe in Equation (4).

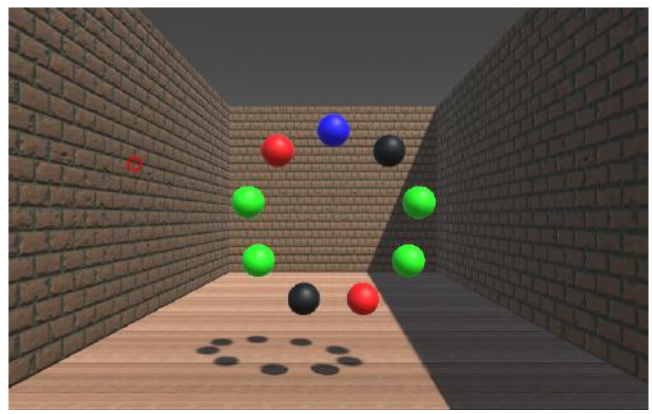

Figure 4: Fitts' law test environment in Unity. The red cursor depicts the position of the rendered cursor, as described in Section 3.1.

4.3.1 Controllers

Our study included 6 input-device/interaction-technique combinations, which we refer to as "controllers". We describe these, and their effect on the control cursor (noting that the effect on the rendered cursor is implied) as follows:

Mouse: The control cursor is controlled by the mouse using a direct mapping of the mouse's *x* and *y* movement.

Head: The control cursor was fixed in the center of the field of view, and thus was only controlled by the user's head gaze. This was intended as a baseline condition (i.e., EZCursorVR was disabled) to assess the added value of independent cursor control.

Velocity-Joystick: The control cursor is controlled by the joystick on the Oculus Touch controller and moves at a constant velocity in the direction the user pushes on the joystick.

Position-Joystick: The control cursor is controlled by the joystick on the Oculus Touch controller but uses a position-control mapping. It thus moves depending on the location the joystick is pushed to; pushing the joystick moves the cursor to the corresponding *position* on the field of view. When the user is not pushing the joystick, the control cursor returns to the center.

Ray2D: The control cursor position is determined by the intersection of the head-coupled plane and the 6DOF ray from the

[4] https://github.com/adrianramcharitar/UnityFittsLawVR

Oculus Touch controller. In other words, the user points the controller *at* the plane to control the cursor position, rather than at objects themselves.

Ray3D: The user controls a standard 6DOF ray using the Oculus Touch controller, necessitating selection by pointing at the target volumes (rather than their projection). This was intended as another baseline condition, as the most typical interaction technique used with 6DOF-tracked wands in modern VR games.

4.4 Procedure

Upon arrival, we asked participants to answer a pre-experimental questionnaire about their familiarity with VR input devices and any previous VR experiences. They were then shown how to use each of the controllers and how the target selection task worked. They were given a practice round to familiarize themselves with the hardware and software. Data gathered from these practice trials were excluded from our analysis. After the participants were comfortable using the hardware and software, they were then asked to perform the actual experiment. Their instructions were to select the highlighted target as quickly as possible and as close as possible to the centre. Upon pressing the selection button, the trial advanced to the next target (which turned blue, indicating it was the "active" target) regardless if the selection hit or missed. Upon finishing a round (9 targets) a new combination of target width, distance, and depth was randomly picked (without replacement). The experiment ended after the participant completed all combinations of distance, width, and depth, with each controller. Following each controller condition, participants were asked to fill out a questionnaire so we could gather qualitative data about such features as using a combination of head and cursor movement to select targets, as well as controller preference. Their responses were written down and then analysed

After completing the experiment, we gave participants another questionnaire that asked them to evaluate their preference toward each controller. We also asked them to rank their preferred controller from best to worst. Finally, they were debriefed and were given $10 compensation. The experiment took roughly 1 hour.

4.5 Design

Our experiment used a within-subjects design with the following independent variables and levels:

Controller: Mouse, Ray2D, Ray3D, Head, Velocity-Joystick, Position-Joystick.
Width: 0.75, 0.5, 0.25 m
Distance: 1, 2, 3 m
Depth: 10, 20, 30 m

Each participant completed 9 trials per round × 6 controllers × 3 distances × 3 widths × 3 depths = 1458 trials, or 26244 trials over all 18 participants. The combinations of distance and width produced 9 indices of difficulty, ranging from 1.2 bits to 3.7 bits. Width, distance, and the resulting *ID* combinations were not analyzed, but used to produce a realistic range of task difficulty.

Our experiment included 3 dependent variables: Throughput (bits/sec, calculated as described earlier), error rate (percentage of missed targets), and movement time (in milliseconds). Movement time was calculated as the difference in time from selection of target n to target n+1.

5 Results and Discussion

5.1 Throughput

Results for throughput are shown in Figure 5. Repeated measures ANOVA revealed that the main effect of controller on throughput was statistically significant ($F_{5,85} = 68.74$, $p < 0.0001$), as was the main effect for depth ($F_{2,34} = 48.09$, $p < 0.0001$). The controller × depth interaction effect was also statistically significant ($F_{10,170} = 6.87$, $p < 0.0001$). The Scheffe posthoc test indicated that most pairs of controllers were significantly different ($p < .05$). These pairwise differences are also seen in Figure 5. Average throughput with the mouse was somewhat lower at around 2.66 bps than those of the other 3D studies that have reported mouse throughput of around 3.7 bits/sec [19]. This may be because the cursor was controlled by both the head and the mouse, and head movements may have adversely affected the throughput. Previous studies did not use head-coupled cursor planes.

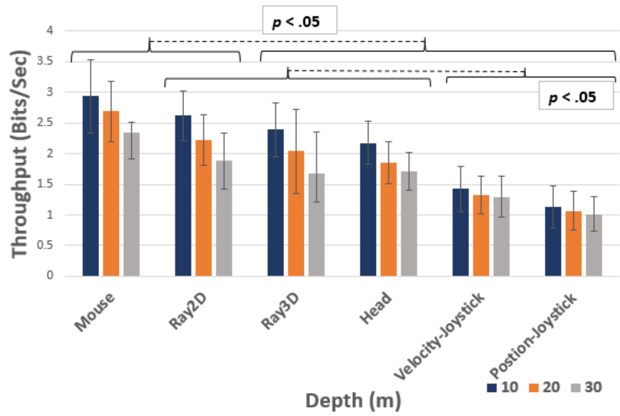

Figure 5: Throughput by Depth. Error bars show ±1 *SD*. Statistical groups (i.e., controllers that are not significantly different) are indicated with curly braces, with dashed lines showing significant differences to other groups via the Scheffe test.

5.2 Movement Time

Results for movement time are shown in Figure 6. Repeated-measures ANOVA revealed that the main effect of controller on movement time was statistically significant ($F_{5,85} = 36.63$, $p < 0.0001$) as was the main effect on depth ($F_{2,34} = 8.48$, $p < 0.005$). The controller × depth interaction effect was not statistically significant ($F_{10,170} = 1.21$, $p > 0.5$). The Scheffe posthoc test revealed many pairwise differences between the controller types ($p < .05$) – all of the Mouse, Ray2D, Ray3D, and Head controllers had significantly faster movement times than the two joystick-based controllers. These are seen in Figure 6.

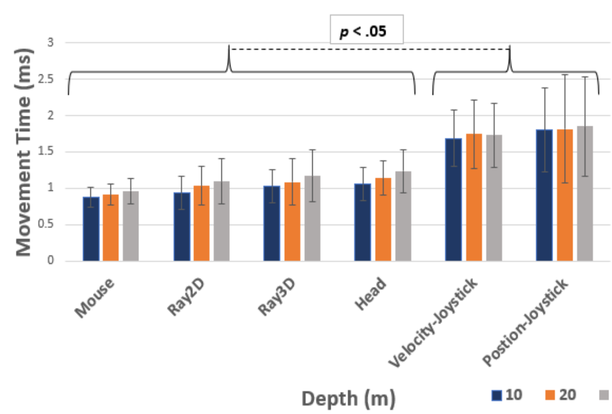

Figure 6: Movement time by controller and depth. Error bars show ±1 SD.

5.3 Error Rate

Results for error rate are seen in Figure 7. Repeated-measures ANOVA revealed that the main effect of controller on error rate was statistically significant ($F_{5,85} = 20.43$, $p < 0.0001$) as was the main effect on depth ($F_{2,34} = 224.62$, $p < 0.001$). The controller × depth interaction effect was statistically significant ($F_{10,170} = 7.43$, $p < 0.001$). The Scheffe post hoc test revealed four pair-wise significant differences ($p < .05$), seen in Figure 7.

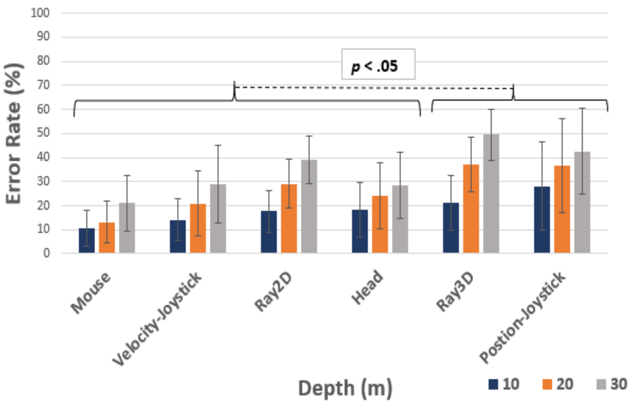

Figure 7: Error rate by controller and depth. Error bars show ±1 *SD*.

5.4 Subjective

Participants ranked the control schemes on a 5-point Likert scale for perceived accuracy, fatigue and speed. Results are shown in Figure 8. The non-parametric Friedman test revealed a significant difference for accuracy, fatigue and speed ($\chi^2 = 59.6$, $p < 0.0005$, $df = 5$), ($\chi^2 = 16.9$, $p < 0.005$, $df = 5$) and ($\chi^2 = 42.1$, $p < 0.0005$, $df = 5$) respectively. Vertical bars (●━●) show pairwise significant differences.

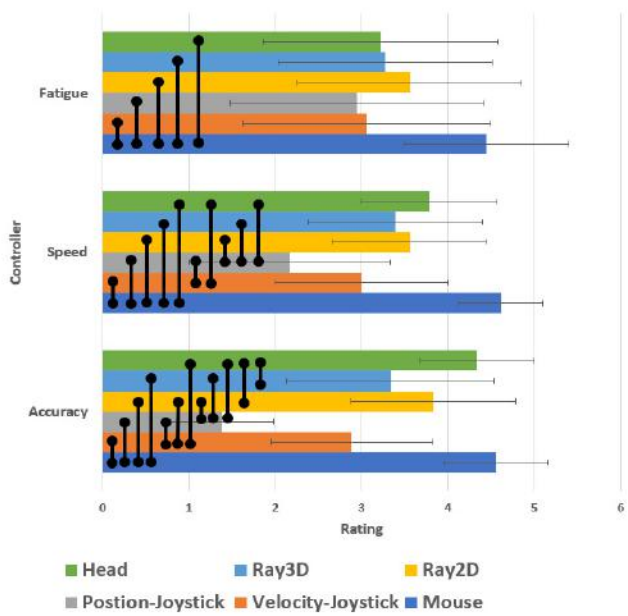

Figure 8: Qualitative Results for controller Fatigue, Speed and Accuracy. Error bars show ±1 SD.

6 Overall Discussion

Overall, the mouse outperformed the other controllers. This was expected based on previous work, and as the mouse was most familiar controller. However, using the mouse with EZCursorVR yielded worse performance than in previous work in non-head-tracked stereo 3D environments. Although we anticipated a larger difference, this result still validates the basic concept of EZCursorVR – the technique offered better performance than other common VR selection techniques, notably rays controlled by either a wand or the head.

Hypothesis H1, that 2DOF devices would perform better than 3/6DOF devices, was partly confirmed. The mouse Ray2D were the two top performers. EZCursorVR worked well with some of the controller input devices. On the other hand, both joystick-based controllers performed very poorly. This suggests that the performance of EZCursorVR is highly dependent on the actual input device it is used with. Future work will investigate this further.

Similarly, hypothesis H2 was partially confirmed as well. While the mouse and Ray2D did outperform the other controller schemes, velocity-joystick didn't perform as well as expected. The poor performance of the velocity-joystick may be attributable to the constant cursor speed. This restricted participant control over cursor acceleration, resulting in frequent overshooting of targets. This may highlight an opportunity to use CD gain, or a more complex transfer function to potentially improve joystick performance. Position-joystick also offered very low performance. This can likely be attributed to the high sensitivity of the cursor, and the fact that participants were unfamiliar with position-controlled cursors in general.

As expected the mouse had the lowest error rate. Both ray controllers as well as the head only had lower error rates compared to both joystick controller schemes. We attribute this to the abstract and unnatural pointing nature of the joysticks as opposed to a more natural feeling, 'look to select' or 'point to select' methods of the ray and head only controllers, especially for far or small targets where a controller that can fine tune the movement of the cursor would result in lower error rates.

Participants experienced some difficulty selecting remote targets with the ray-based techniques, as we had anticipated. With Ray3D, selecting remote targets was difficult, due to their smaller angular size. As a result, most participants preferred Ray2D over Ray3D, especially when selecting far targets. Additionally, participant hand tremor, while small, propagated up the visible ray in Ray3D causing it to sway substantially, also making it difficult for selecting far targets. Although Ray2D also used a ray, this swaying was reduced due to the comparatively short distance to the head-coupled plane. This likely explains the easier time participants had with Ray2D.

Surprisingly, H3 was not found to be true, despite previous evidence [32] that suggests throughput calculated in the plane should be constant over depth. There are two possible reasons for this. First, we used more extreme depth differences than in previous work, which was constrained to a depth range of about 28 cm. In contrast, our depth range was 30 m. Another factor is that head motion influenced our techniques, unlike in previous work. In our study, the head was a constant source of potential input noise, as it was the origin of all rays. These two factors, taken together, may have yielded this result, and may speak to a limitation of our technique and/or a need to reinvestigate projected throughput.

The coupling of head with the cursor movement proved valuable for both joystick controllers as participants. Participants were observed using a combination of head and joystick movement and confirmed this in post-experiment debriefing. Several participants noted they used the joystick for coarse motions, and then "fine tuned" their selection via head movement. This made it easier to select small or high distance targets (although opposite to how we initially expected). Similarly, some participants expressed interest in being able to switch between the head only and EZCursorVR while performing selections (i.e., toggling independent cursor movement on and off). This is a topic for a future study, and will allow us to definitively determine if participants actually use the two control styles (head + controller) together or independently.

7 Limitations and Future Research

In the current experiment, both joystick controllers moved the control cursor linearly, adjusted by a scale factor. Adding a dynamic and potentially non-linear gain function to provide cursor acceleration could improve joystick performance, and perhaps even the mouse condition. Consider, for instance, the difficulty participants had in selecting remote targets. Remote targets perspective scale to be smaller and hence harder to select targets. A gain function that reduced gain with slow movements might make these easier to select. As argued earlier, we see this is a principle advantage of EZCursorVR over classical image-plane selection techniques that rely exclusively on 1:1 selection.

The visual design of EZCursorVR itself could also be explored, for instance, using different crosshair styles, sizes, and transparency levels. Participants noted that, especially for small targets, the cursor could sometimes occlude targets, making it more difficult to select them. An improved visualization might eliminate such problems.

We also note that our study only included selection of non-occluded objects. A follow up study could explore how users can select objects behind other objects. This might be accomplished, for example, by changing the roll of the controller (an unused DOF) for depth selection.

We also note that a number of control variables were chosen based on pilot testing and could be further explored for "fine-tuning". Other factors such as coupled/decoupled head schemes, linear/non-linear cursor movements, non-gamers/experienced gamers would further add to and reinforce our initial study. Exploring a longitudinal study would also be beneficial for learning the effects of performance of the control schemes over longer periods of time.

8 Conclusion

EZCursorVR offers a potentially effective alternative to 3D selection methods for use with head-mounted displays. Using the Oculus Rift, we implemented EZCursorVR controlled with a mouse, the head, as well as two joystick and two ray-based input methods. We tested performance of these 6 controller schemes using a Fitts' law selection task built in Unity.

Results were favourable for EZCursorVR when using the mouse across all dependent variables, as expected. The 2DRay also performed better (although generally not significantly so) than the 3DRay, which worked better than the joystick-based controllers.

Overall, our results are encouraging, but speak to a need for further investigation with different controllers used with EZCursorVR. The technique's performance is strongly dependent on the controller device that it is being used with. It can either perform well or poorly with certain controllers as our study has shown. Future studies will explore devices specifically built for using in conjunction with EZCursorVR and well as analysing several improvements to the movement and design of the cursor itself.

REFERENCES

[1] Ferran Argelaguet and Carlos Andujar, A survey of 3D object selection techniques for virtual environments, Computers and Graphics (Pergamon), vol. 37, no. 3, pp. 121–136, 2013.

[2] Ferran Argelaguet and Carlos Andujar, Improving 3D selection in VEs through expanding targets and forced disocclusion, in Proceedings of the 9th international symposium on Smart Graphics (SG '08), vol. 5166 LNCS, pp. 45–57, 2008.

[3] Ferran Argelaguet and Carlos Andujar, Efficient 3D pointing selection in cluttered virtual environments, IEEE Computer Graphics and Applications, vol. 29, no. 6, pp. 34–43, 2009.

[4] François Bérard, Jessica Ip, Mitchel Benovoy, Dalia El-Shimy, Jeffrey R. Blum, and Jeremy R. Cooperstock, Did "minority report" get it wrong? Superiority of the mouse over 3D input devices in a 3D placement task, vol. 5727 LNCS, no. PART 2. 2009.

[5] Frederick P. Brooks, What's real about virtual reality?, IEEE Computer Graphics and Applications, vol. 19, no. 6, pp. 16–27, 1999.

[6] Gerd Bruder, Frank Steinicke, and Wolfgang Stuerzlinger, Touching the void revisited: Analyses of touch behavior on and above tabletop surfaces, in Lecture Notes in Computer Science, vol. 8117 LNCS, no. PART 1, pp. 278–296, 2013.

[7] Andéol Évain, Ferran Argelaguet, Géry Casiez, Nicolas Roussel, and Anatole Lécuyer, Design and evaluation of fusion approach for combining brain and gaze inputs for target selection, Frontiers in Neuroscience, vol. 10, 2016.

[8] Paul M. Fitts, The information capacity of the human motor system in controlling the amplitude of movement., Journal of Experimental Psychology, vol. 47, no. 6, pp. 381–391, 1954.

[9] Tovi Grossman and Ravin Balakrishnan, The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area, Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '05), pp. 281–290, 2005.

[10] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani, Modeling Cumulative Arm Fatigue in Mid-Air Interaction based on Perceived Exertion and Kinetics of Arm Motion, in Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '17), pp. 3328–3339, 2017.

[11] David Antonio Gómez Jáuregui, Ferran Argelaguet, and Anatole Lecuyer, Design and evaluation of 3D cursors and motion parallax for the exploration of desktop virtual environments, in Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI '12)., pp. 69–76, 2012.

[12] Regis Kopper, Felipe Bacim, and Doug A. Bowman, Rapid and accurate 3D selection by progressive refinement, in Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI '11)., pp. 67–74, 2011.

[13] Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug Bowman, and Ivan P. Poupyrev, 3D user interfaces : Theory and Practice, 2nd ed. Redwood City, CA, USA.: Addison-Wesley Professional, 2017.

[14] Sangyoon Lee, Jinseok Seo, Gerard Jounghyun Kim, and Chan-mo Park, Evaluation of pointing techniques for ray casting selection in virtual environments, In Third International Conference on Virtual Reality and Its Application in Industry, pp. 38–44, 2003.

[15] Paul Lubos, Gerd Bruder, and Frank Steinicke, Analysis of direct selection in head-mounted display environments, in Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI '14)., pp. 11–18, 2014.

[16] I.Scott MacKenzie, Fitts' Law as a Research and Design Tool in Human-Computer Interaction, Human–Computer Interaction, vol. 7, no. 1, pp. 91–139, 1992.

[17] Victoria McArthur, Steven J. Castellucci, and I.Scott MacKenzie, An empirical comparison of "wiimote" gun attachments for pointing tasks, Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '09), pp. 203–208, 2009.

[18] Mark R. Mine, Frederick P. Brooks, Jr., and Carlo H. Sequin, Moving Objects in Space: Exploiting Proprioception In Virtual-

Environment Interaction, Proceedings of ACM SIGGRAPH ’97, pp. 19–26, 1997.
[19] Daniel Natapov and I.Scott MacKenzie, The trackball controller: improving the analog stick, in Proceedings of the International Academic Conference on the Future of Game Design and Technology, pp. 175–182, 2010.
[20] J. S. Pierce, A. Forsberg, M. J. Conway, S. Hong, R. Zeleznik, and M. R. Mine, Image Plane Interaction Techniques in 3D Immersive Environments, in ACM SIGGRAPH Symposium on Interactive 3D Graphics, pp. 39–44, 1997.
[21] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst, Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques, Computer Graphics Forum, vol. 17, no. 3, pp. 41–52, 1998.
[22] Ivan Poupyrev and Mark Billinghurst, The go-go interaction technique: non-linear mapping for direct manipulation in VR, Proceedings of the 9th annual ACM Symposium on User Interface Software and Technology (UIST ’96), pp. 79–80, 1996.
[23] Yuanyuan Qian and Robert J. Teather, The eyes don’t have it : An empirical comparison of head based and eye-based selection in virtual reality, in Proceedings of the 5th ACM Symposium on Spatial User Interaction (SUI ’17), pp. 91–98, 2017.
[24] Adrian. Ramcharitar and Robert. J. Teather, A head coupled cursor for 2D selection in virtual reality, in Proceedings of the 5th ACM Symposium on Spatial User Interaction (SUI ’17), pp. 162–162, 2017.
[25] Leila Schemali and Elmar Eisemann, Design and evaluation of mouse cursors in a stereoscopic desktop environment, in In Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI ’14)., pp. 67–70, 2014.
[26] International Standard, Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non -keyboard input devices, Iso 2000, vol. 2000, p. 54, 2000.
[27] Anthony Steed, Towards a general model for selection in virtual environments, in Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI ’06), pp. 103–110, 2006.
[28] Anthony Steed and Chris Parker, 3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays, 8th International Immersive Projection Technology Workshop., pp. 1–8, 2004.
[29] Robert J. Teather, Andriy Pavlovych, and Wolfgang Stuerzlinger, Effects of latency and spatial jitter on 2D and 3D pointing, in Proceedings IEEE Virtual Reality, pp. 229–230, 2009.
[30] Robert J. Teather and Wolfgang Stuerzlinger, Visual aids in 3D point selection experiments, Proceedings of the 2nd ACM Symposium on Spatial User Interaction (SUI ’14), pp. 127–136, 2014.
[31] Robert J. Teather and Wolfgang Stuerzlinger, Pointing at 3D targets in a stereo head-tracked virtual environment, in Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI ’11), pp. 87–94, 2011.
[32] Robert J. Teather and Wolfgang Stuerzlinger, Pointing at 3D target projections with one-eyed and stereo cursors, Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ’13), pp. 159–168, 2013.
[33] Lode Vanacken, Tovi Grossman, and Karin Coninx, Exploring the effects of environment density and target visibility on object selection in 3D virtual environments, in Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI ’07)., pp. 115–122, 2007.
[34] Colin Ware and Kathy Lowther, Selection using a one-eyed cursor in a fish tank VR environment, ACM Transactions on Computer-Human Interaction, vol. 4, no. 4, pp. 309–322, 1997.
[35] Jonathan Wonner, Antonio Capobianco, and Dominique Bechmann, Starfish : a Selection Technique for Dense Virtual Environments, Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology, pp. 101–104, 2012.
[36] Thomas S. Young, Robert J. Teather, and I. Scott Mackenzie, An arm-mounted inertial controller for 6DOF input: Design and evaluation, in Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI ’17)., pp. 26–35, 2017.