Is the Pen Mightier than the Controller? A Comparison of Input Devices for Selection in Virtual and Augmented Reality

Duc-Minh Pham
Simon Fraser University
Surrey, Canada, British Columbia
ducminhp@sfu.ca

Wolfgang Stuerzlinger
Simon Fraser University
Surrey, British Columbia, Canada
w.s@sfu.ca

Figure 1: Illustration of the input devices we compared in Virtual and Augmented Reality systems for pointing tasks.

ABSTRACT

Controllers are currently the typical input device for commercial Virtual Reality (VR) systems. Yet, such controllers are not as efficient as other devices, including the mouse. This motivates us to investigate devices that substantially exceed the controller's performance, for both VR and Augmented Reality (AR) systems. We performed a user study to compare several input devices, including a mouse, controller, and a 3D pen-like device on a VR and AR pointing task. Our results show that the 3D pen significantly outperforms modern VR controllers in all evaluated measures and that it is comparable to the mouse. Participants also liked the 3D pen more than the controller. Finally, we show how 3D pen devices could be integrated into today's VR and AR systems.

CCS CONCEPTS

• **Human-centered computing** → **Pointing devices**; **Pointing**; **Virtual Reality**; **Mixed / augmented Reality**;

KEYWORDS

3D pointing, input devices, Virtual and Augmented Reality

ACM Reference Format:

Duc-Minh Pham and Wolfgang Stuerzlinger. 2019. Is the Pen Mightier than the Controller? A Comparison of Input Devices for Selection in Virtual and Augmented Reality. In *Proceedings of The ACM Symposium on Virtual Reality Software and Technology (VRST).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.475/123_4

VRST, November 2019, Sydney, Australia

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

1 INTRODUCTION

Based on recent technological advances, Virtual and Augmented Reality (VR/AR) applications have become more popular, in education, research, training, and entertainment. Powerful computing hardware, high-resolution head-mounted displays (HMDs), and sub-millimeter tracking systems not only deliver a comfortable experience but also pave the way for professional applications. VR/AR users typically manipulate virtual content through handheld devices, (e.g., HTC Vive controllers), their gaze, or their fingers (e.g., HoloLens). This primary interaction method plays a key role in different scenarios, including selecting, manipulating, and creating virtual content. Also, different applications may require different levels of selection accuracy. For example, selecting a small datapoint in a 3D scatterplot may need higher precision than choosing a large menu button. This motivates research on VR/AR selection methods.

Virtual environments can have any size and virtual objects can thus be far away from the user. Similarly, augmented content attached to real-world targets in AR can be distant from the user's hands. Therefore, we focus on selection methods suitable for both near and distant objects in this work. Virtual hand techniques with a one-to-one mapping are limited to the reach of the human arm, which makes them a sub-optimal choice for some scenarios. Nonlinear mappings [21, 26] allow users to interact with distant objects, but potentially prevent them from accurately selecting small parts of 3D mechanical models or a single data point. Exocentric metaphors [2], such as automatic scaling or the world-in-miniature technique, can address the issue, but require transformations of virtual or augmented objects, which potentially break the immersive experience, increase interaction effort, and/or introduce motion sickness. 3D navigation methods apply only in VR scenarios and also increase interaction effort. Thus, we chose to evaluate user performance with ray-casting, the most popular egocentric method for distant object selection [5]. Most VR users are familiar with ray-casting as many applications in commercial systems use this technique to enable interaction with distant virtual items or the menu buttons.

Most current VR controllers are designed to be held in a power grip (and not in a precision grip). Thus, one might wonder if they are an optimal choice for selection tasks in term of accuracy and error rates. With ray-casting the ray typically starts from one end of the controller and users need to either rotate their wrist or even move their arm to control that ray. An unanswered question is if this the most efficient option for interaction. Different grips involve different muscle groups, which potentially results in different pointing performances. Due to the tripodal finger configuration, a precision grip affords more accuracy than a power grip, which is best illustrated by the progression of pen grips observed in children who are learning to write [27]. In essence, a pen is a cylindrical object small enough for users to manipulate with multiple finger tips, when held in the typical pen grip. Previous studies has evaluated devices held in either a power grip, e.g., [15], or a precision grip, e.g., [17, 18, 23]. However, no one has yet done a formal comparison between devices held in these two grips. Also, several VR/AR pens or stylii, such as Massless (massless.io) and HoloStylus (hololight.com), have been recently introduced to provide more options for content creation and interaction. Thus, we evaluate a pen-like device held in a precision grip and compare its performance with a traditional VR controller, to provide timely design guidance for AR/VR developers.

For 2D content, a mouse or a touch screen are among the best options for interaction. Many users are already very experienced with these devices and can point at small targets with high accuracy. The technical properties of current mice in terms of latency, sampling rate, and sensor resolution support this, too. Given its ubiquity, this makes the mouse an ideal baseline condition, even though it is not a 3D input device. Also, previous work has identified that a controller is generally inferior relative to the mouse in terms of pointing performance [32]. Therefore, in this study, we investigate also if a pen could reach performance comparable to a mouse.

Hand-held devices like a controller or pen can be used to perform 3D pointing with ray-casting. This method does not work with the mouse. Users typically manipulate the mouse on a 2D surface to perform 2D pointing on a computer monitor. However, other work has shown that a mouse cursor can be used for 3D pointing with ray-casting [10, 33]. Thus, we chose to perform our evaluation with a selection task where targets are located on a virtual plane that faces the user. Then, users can move a cursor on that plane to select the targets with any of our chosen input devices. Hence, they perform 2D interaction with the mouse but 3D interaction with the controller and the pen (albeit both the controller and pen are practically only moved through 2D rotations in our task). This choice also enables us to evaluate the pointing performance with the very well-defined and reliable methodology associated with Fitts' law [12].

Other studies have evaluated different methods that use a human finger or the arm as a pointing device. Brown et al. identified that a bare finger tracked by the LeapMotion is not an efficient pointing method [7]. Other work used a motion capture system to track the user's finger or arm in 3D space [20, 25, 35]. However, these techniques require retro-reflective markers attached to the user's finger or a custom glove, which is not practical in many scenarios. As designated input devices can be shared between users with little concern for setup time and tracking quality, we decided to limit our work to device-based pointing.

Past work has evaluated the performance of distant pointing with wands and lasers in front of large displays, e.g., [15, 22, 23]. Yet, modern VR/AR systems involve HMDs and afford an experience that is different from large displays. Moreover, HMDs allow the user to be "inside" the virtual environment, which affords a higher degree of immersion.

We present the following contributions:

- A comparison of the mouse, a controller, and a pen as input devices for distant pointing for users wearing VR and AR headsets.
- We identify likely reasons for why the pen is a significantly better device for pointing in VR and AR compared to the controller.
- We also discuss how pens could be integrated as input devices into today's AR and VR systems.

2 RELATED WORK

Here we first review evaluation methods for pointing devices, then discuss human factors related to pointing, and finally present the results of previous work for various input devices.

2.1 Fitts' Law and Throughput

Fitts' law [12] is well known in 2D user interfaces for its ability to accurately predict how long a pointing task to a target with a given size at a given distance will take:

$$MT = a + b \log_2 \left(\frac{A}{W} + 1 \right) \tag{1}$$

where MT is the time to point to a target, A its distance, W its size, and a and b are empirically determined constants. The logarithmic term is also called the index of difficulty of pointing (ID).

Extending Fitts' law, the *throughput* measure better captures the natural speed-accuracy tradeoff [1, 19] of pointing. Throughput is defined through the use of effective measures that take the task that the users actually performed into account [1, 19]. Most work in 2D user interfaces has adopted this methodology to characterize pointing performance. Followup work generalized this approach for measuring pointing performance to 3D [30, 32]. One important insight from this line of work is that latency can have significant impact on pointing performance [24, 30].

2.2 Human Factors for Pointing

Previous research investigated the influence of muscle groups on pointing performance. For 6DOF devices, Zhai [38] identified that holding an ball-shaped input device with the fingers can afford faster 6DOF manipulation relative to a palm-held device. Balakrishnan and MacKenzie compared the performance of 2D input controlled with a single finger, the wrist, and the forearm and found that a single finger does not perform better than other options [3].

2.3 Input Devices for 3D pointing

Many input devices exist for 2D desktop applications. With proper mappings, such devices can also work well for 3D tasks [5]. Argelaguet et al. surveyed 3D object selection techniques [2], while Hoppe et al. surveyed the input and output devices and associated interaction techniques for 3D interaction [13]. Here we limit our discussion to those directly relevant for our work. Specifically, we do not discuss glove- or hand-based interfaces, also because virtual hand selection is outside of our scope. We focus mostly on comparison based on throughput, as this measure is more robust to the speed-accuracy tradeoff. This is important for comparisons of different input devices that are subject to different amounts of latency and other limitations of tracking technologies, such as tracking noise.

2.3.1 Mouse. The mouse is an ideal input device for 2D interfaces and some 3D interfaces [37]. Previous comparisons have identified that the mouse works well even for 3D manipulation [4]. Krichenbauer et al. [16] compared the mouse and a 3D input device for 3D manipulation and found no significant difference. For 3D pointing, Teather and Stuerzlinger identified that a mouse controlling a cursor performs better than other alternatives [32] and that displaying the cursor only to a single eye can mitigate potential depth conflicts for the mouse cursor in a 3D virtual environment [33].

2.3.2 Controller. Butterworth et al. [9] used a handheld 6D mouse as input device, which functioned as a 3D controller. Teather and Stuerzlinger [32] found that throughput of a controller-like device, was less than for the mouse, see their video, which was confirmed in recent work [29].

2.3.3 Pen. Early work with the throughput measure evaluated a pen-like device for 3D pointing [32] in a fish-tank VR system, where all targets were within arm's reach. Also, the authors stated that their implementation might have suffered from rotational jitter. Building on a study of un-instrumented finger pointing [8], Brown et al. investigated a chopstick (a pen-like device) tracked by the Leap Motion [7] and found the throughput to be (almost) as good as the mouse. This promising result further motivated us to measure the throughput of a pen-like device in an immersive environment.

3 PILOT STUDIES

We performed several pilot studies to identify a pen-like device that works well. For each we used the same ISO 9241-411 methodology [1] as in our main user study, see below.

We first tried to emulate a chopstick device [7] by attaching a long thin rod to a HTC Vive standalone tracker, see Figure 2. Yet, we found that the uneven weight distribution made the device uncomfortable to use, and we observed pointing performance substantially below a controller.

Subsequently, we tried to use the Vive Controller as a pen, by balancing the center of the device on the hand between the thumb and the side of the index finger and using its base as the pointing end, see figure 2. Yet, the controller was too heavy to be moved with the fingers and most participants naturally reverted to using their wrists to control the device. We thus were unable to identify a significant difference between the controller held normally and like a pen.

We also experimented with a ballpoint-pen-like device, as illustrated in Figure 2. This version was too thin and the markers too close together, which affected performance negatively. This motivated us to look at bigger pens.

Finally, we attached a Vive tracker to an optical mouse and compared its native performance, the Vive-tracked version [30], and the Vive controller. This pilot only identified a significant difference for throughput (but not for time or errors) for the mice, with the normal mouse being better.

4 MAIN USER STUDY

The main goal of our study was to compare the pointing performance of various input and output devices. As previous work, we use the ISO 9241-411 methodology [1, 19] to compare not only pointing time and errors but also throughput.

4.1 Subjects

Twelve people (3 female), with ages ranging from 23 to 33 (M = 27.92, SD = 2.84), participated in this study. Only one of them was left-handed. Based on the Porta eye dominance test, 41.67% were left eye dominant. More than half, 58.33% played 3D games more than 5 hours a week. Participants were paid a small compensation for their participation.

4.2 Apparatus

To guarantee that machine performance was not a limiting factor, we used a PC with an Intel® Core™ i7-4790, 16GB RAM, and a nVidia GTX 1080, running Windows 10. The components of this PC far exceed the requirements for our VR and AR headsets. We chose VR and AR headsets with (roughly) similar specifications.

4.2.1 VR Headset. The HTC Vive Pro features a total display resolution is 2880×1600, with 90 Hz expected refresh rate, see Figure 3 (left). The horizontal field of view is approximately 100°. The Vive Pro weighs about 550 g.

4.2.2 AR Headset. The Meta 2 has a total resolution is 2560×1440, with an expected refresh rate of 60 Hz, see Figure 3 (right). We chose this tethered AR headset over un-tethered alternatives, as it has a much larger horizontal field of view, approximately 90°. The Meta 2 weighs about 500 g.

4.2.3 Optical Tracking System. Both the HTC Vive Pro and the Meta 2 have their own tracking systems, with different tracking technologies (Vive Pro outside-in with base stations emitters and Meta 2 inside-out with RGB and depth cameras). These two tracking systems use radically different technologies, which might result in different latency and/or accuracy. To keep the tracking quality consistent between our VR and AR conditions, we decided not to use either of these two tracking systems.

Instead, we used an external outside-in optical tracking system from OptiTrack. In our setup, there were eight OptiTrack S250e, 250 Hz IR cameras, which were hung above the experimental area. The OptiTrack system was calibrated to sub-millimeter accuracy, which corresponds for the pen also to well below a degree of rotation error. We attached optical markers to each headset in different configurations, see Figure 3, to ensure they could be tracked reliably. We placed the markers at the top of the headsets to avoid potential

Figure 2: Prototypes explored during the design process. *Left*: Pen tracked by Vive Tracker. *Middle*: HTC Vive controller used as a pen. *Right*: Ballpoint pen device. All these did not perform well in our pilot studies.

Figure 3: *Left*: HTC Vive Pro with 4 reflective markers. *Right*: Meta 2 with 4 reflective markers.

Figure 4: From left to right: *(a)* Mouse, *(b)* Standard whiteboard pen, *(c)* Wrapped whiteboard pen with 4 reflective markers, and *(d)* Vive controller with 4 reflective markers.

occlusions, even by the user's hands. To avoid interference, we blocked the IR illuminator of the Meta 2, see Figure 3 (right). To achieve the best possible tracking performance, participants were asked to sit at the center of the experimental space in a swivel chair, with a Mobo keyboard and mouse tray, see Figure 5, onto which we placed a wireless keyboard and mouse.

The OptiTrack cameras were connected over a Gigabit Ethernet switch to a secondary tracking PC, which sent the tracking results to the main PC over the network. While this increased the latency slightly, it ensured that the CPU of the main experimental PC was not fully loaded, reducing the potential for dropped frames. We also tracked several input devices through the OptiTrack system in our work, as described in the following.

4.2.4 Mouse. For the *mouse* conditions, we chose a wireless Logitech M215 mouse as input device, see Figure 4 (a). This average mouse is easy to operate and fits most people's hands well. To ensure good tracking performance, we regularly checked the batteries and replaced them to avoid potential tracking issues during the experiment.

We chose not to show a virtual mouse model in the VR condition. The mouse was located on the Mobo's mouse pad attached to the chair, well below the user's line of sight when looking straight ahead at the targets. This is also similar to how most people use a mouse in their daily computer work. We did not track the mouse via the OptiTrack system, see also below. To keep internal validity high, we required that participants indicated selection through the space bar of the keyboard on the Mobo tray, with the their non-dominant hand, see Figure 5. The mouse buttons had no function in the trials. Participants used this mouse naturally through a combination of elbow, wrist and (finer) finger movements.

4.2.5 Controller. We used an HTC Vive controller, which weights 470*g*. Instead of tracking it via the Vive base stations, we attached a set of four reflective markers on its upper surface, as shown in Figure 4 (d), and tracked the device with the OptiTrack system. The positions of the markers were chosen carefully so that participants could still hold the controller comfortably without hiding the markers. Because the *controller* was potentially visible in the user's field of view, we showed its 3D model in VR to improve the consistency between VR and AR conditions. As the controller is held in the palm, participants used this device with a combination of wrist and elbow movements. As in the *mouse* condition, participants indicated selection through the space bar of the keyboard. Our choice was motivated by the fact that activating the controller buttons can cause "dips", which can affect selection performance negatively [6].

4.2.6 Pen. As mentioned above, we iterated through several designs before ending up with the final version shown in Figure 4 (c).

We took a normal whiteboard pen, Figure 4 (b), and attached four reflective markers so that the OptiTrack cameras could determine its 6DOF pose. To eliminate tracking interference we wrapped the pen in black tape. The weight of this current *pen* was about 60 *g*, with an average diameter of approximately 1 *cm*. The balance and weight of the device enables users to hold it like a real pen, which means that they were able to use multiple fingers in coordination as well as the wrist to move the device. Similar to how participants held the *controller*, the pen was often held so that it was visible in the field of view of the VR headset. Hence, we displayed a virtual 3D model of the *pen* in VR so that they could be aware of its position and direction. As with the other input devices, participants also indicated selection via the space bar of the keyboard.

4.3 Pointing Task

In this study, participants were asked to perform pointing tasks in different combinations of Immersive Environments and Input Devices. Each pointing trial required participants to use one of the input devices to point to 11 spherical targets, while wearing either the VR or AR headset. Corresponding to a 3D version of the ISO 9241-411 task [1], targets were shown in a circular arrangement in a plane parallel to the view plane, with the center aligned to the eye level of participants at the beginning of the trial, as shown in Figure 5. We also displayed the transparent plane containing the targets, so that participants could easily recognize the planar arrangement of those targets in 3D space. The diameter of the circle (and thus the distance between targets) and the size of the targets varied across trials. We chose a pointing task on a 2D plane in 3D, as ray-casting involves mostly rotating the wrist in 2DOF, making it comparable to 2D manipulation. Moreover, Teather et al. [33] have shown that pointing to 3D targets at different distances is equivalent to projecting all targets to the same plane. Although this study was conducted in front of normal display, it is reasonable to expect the same result when evaluating pointing at a virtual plane in VR/AR.

Participants manipulated a green spherical cursor on that target plane with the input devices: *mouse*, *controller* or *pen*. For the *mouse*, we simply mapped motions to cursor movements on the target plane. With the *controller* and *pen*, we used ray-casting based on the current 6DOF position of the device and displayed the cursor at the ray-plane intersection. The goal of the task was to move the cursor to hit each target to select them in the order specified by the ISO 9241-411 task [1]. Although the mouse was controlled horizontally on a surface to manipulate the cursor on the vertical virtual plane, previous work has shown that this mapping should not result in a difference in pointing performance, e.g., [31]. Also, users are very adept at using a mouse on a horizontal desk to operate a cursor on a vertical screen.

Initially, all targets were grey except a random yellow one indicating the current *objective*. Participants then control the cursor to hit the target by moving it inside the objective. When the cursor center is inside the target, it was highlighted dark yellow, as highlighting makes it easier to see which object will be selected [34]. Once participants hit the space bar in this state, we counted a "hit" and the target changed to black. Otherwise, if the cursor was outside of the target, the objective became grey and an error sound was triggered to indicate a failed selection. After each selection, the next objective became highlighted. In the prescribed sequence, consecutive targets are (approximately) opposite in the target circle. Participants continued to select objectives until all 11 targets were selected. Figure 5 shows different target states during a pointing trial.

4.4 Procedure

After greeting the participants, we asked them to fill the consent form and then informed them about the general study procedure. Next, they were asked to fill a pre-study questionnaire to record their age and experience with 3D virtual environments. Then, we introduced the pointing task and gave them a demonstration. Then they experienced a practice section where they could try both headsets and all input devices. These initial steps took about 15 minutes.

When participants indicated that they were ready for the main experiment, we positioned the swivel chair at the center of our tracking area. To maintain optimal tracking performance participants were asked not to move or rotate the chair (much) during the study. Then, they performed all pointing tasks with one of the two headsets and the three input devices before taking a 3-minute break and proceeding to the other headset. This sequence helped maintains immersion with each headset and avoided potential negative effects due to repeatedly switching headsets. We counterbalanced the order of headsets across participants. For each headset condition, they used all three input devices (*mouse*, *controller*, and *pen*) in counterbalanced order. This order was the same for both VR and AR conditions of a participant.

For each of the $2 \times 3 = 6$ system configurations, participants were asked to complete a series of 18 pointing tasks. Different pointing tasks had varying parameters of *target size*, *target circle size*, and *target plane distance*. Target size is the diameter of the 11 spherical targets in the circle. This sub-condition had three unique values: 1.5, 2.5, or 3.5 *cm*. The diameter of the target circle sub-condition was either 15, 20, or 30 *cm*. The target plane distance, between the participant and the whole target circle, was either 75 or 150 *cm*. Thus, the total number of target circles for each device was $3 \times 3 \times 2 = 18$. The order of these target circles was also counterbalanced via a Latin square for each participant. In summary, there were 2 headsets $\times$ 3 devices $\times$ 18 trials = 108 target circles for each participant, corresponding to 1188 individual target selections. When the experimental section finished after about an hour, participants were asked to answer a post-study questionnaire to elicit their preferences for the Immersive Environments and Input Devices.

5 RESULTS

5.1 Objective Measures

We first present the results for the objective measures followed by the subjective ones. As all pre-conditions for ANOVA were met by the data, including normality of the data, we applied repeated measures ANOVA unless noted otherwise. If sphericity did not hold for any factor, we used either Greenhouse-Geisser or Huynd-Feldt correction depending on their estimated ϵ [11].

Figure 5: *Left*: All six combinations of Immersive Environments and Input Devices investigated in our user study. *Right*: Screen shots of the pointing tasks in VR (top) and AR (bottom).

To gain deeper insights into the data, we also performed ANOVA on VR and AR separately to investigate the effect of Input Device, for every measure. In addition, we used Holm-Bonferroni correction to conduct post-hoc tests. The effects of sub-factors including *target size*, *target circle size*, and *target plane depth* were insignificant in all conditions, and we thus do not report their results.

5.1.1 Movement Time. The movement time is the average time measured in seconds between the selection of two consecutive targets, excluding the time for the first target. We observed significant effects of both Immersive Environment, $F(1, 11) = 10.554$, $p = .008$, $\eta^2 = .490$, and Input Device, $F(2, 22) = 10.869$, $p = .001$, $\eta^2 = .497$, on movement time. Their interaction, however, was not significant, $F(1.370, 15.065) = 3.775$, $p = .060$, $\eta^2 = .256$.

Investigating each Immersive Environment separately, we observed the significant effect of Input Device on VR, $F(2, 22) = 8.838$, $p = .002$, $\eta^2 = .446$, and AR, $F(2, 22) = 10.850$, $p = .001$, $\eta^2 = .497$. Post-hoc tests indicated that the movement time in VR of *controller* was significantly larger than *mouse* and *pen*. In AR, *pen* required significantly smaller movement time when compared to *mouse* and *controller*, see Figure 6.

5.1.2 Error Rate. We computed the error rate as the ratio of missed selections over the number of targets in a circle. There was no significant difference between VR and AR conditions with regard to the error rate, $F(1, 11) = .020$, $p = .889$, $\eta^2 = .002$. However, the Input Device had a significant effect, $F(2, 22) = 5.114$, $p = .015$, $\eta^2 = .317$. The interaction Immersive Environment × Input Device was significant, $F(2, 22) = 3.450$, $p = .050$, $\eta^2 = .239$.

Investigating each Immersive Environment separately, we observed the significant effect of Input Device on VR, $F(2, 22) = 11.125$, $p < .001$, $\eta^2 = .503$, but not on AR, $F(2, 22) = .684$, $p = .515$, $\eta^2 = .059$. Post-hoc tests showed that participants produced significantly higher error rate with *controller* when compared to *mouse* and *pen* in VR, see Figure 6.

5.1.3 Throughput. The ANOVA indicated that different Immersive Enviroments yielded significantly different throughput, $F(1, 11) = 5.652$, $p = .037$, $\eta^2 = .339$. The main effect of Input Device on throughput was also significant, $F(2, 22) = 14.034$, $p < .001$, $\eta^2 = .561$. Their interaction was not significant, $F(2, 22) = .727$, $p = .495$, $\eta^2 = .062$.

Investigating each Immersive Environment separately, we observed the significant effect of Input Device on VR, $F(2, 22) = 9.454$, $p = .001$, $\eta^2 = .462$, and AR, $F(2, 22) = 13.880$, $p < .001$, $\eta^2 = .558$. Post-hoc tests revealed the same pattern of throughput in both Immersive Environments, which indicated that *controller* generated significantly lower throughput than *mouse* and *pen*.

5.1.4 Cursor Speed. Due to the different control mappings, it is challenging to characterize how efficiently participants used each device. As the targets are all in a plane, cursor speed in that plane is a more objective measure. It captures how fast the cursor on said plane is moving. and is computed as $CursorSpeed = \frac{S}{T}$ where T is the completion time and S the total cursor travel distance on the target plane between the first and the last selection. It is measured in cm/s. The ANOVA revealed that the main effect of Immersive Environment was significant, $F(1, 11) = 6.676$, $p = .025$, $\eta^2 = .378$. The main effect of Input Device was also significant, $F(2, 22) = 10.791$, $p = .001$, $\eta^2 = .495$. Their interaction was not significant, $F(2, 22) = 1.127$, $p = .342$, $\eta^2 = .093$.

Investigating each Immersive Environment separately, we observed the significant effect of Input Device on VR, $F(2, 22) = 8.723$, $p = .002$, $\eta^2 = .442$, and AR, $F(2, 22) = 7.308$, $p = .004$, $\eta^2 = .399$. In VR, post-hoc tests indicated that *pen* manipulated the cursor significantly faster than *controller*. In AR, *pen* resulted in larger cursor speed than *mouse* and *controller*, see Figure 6.

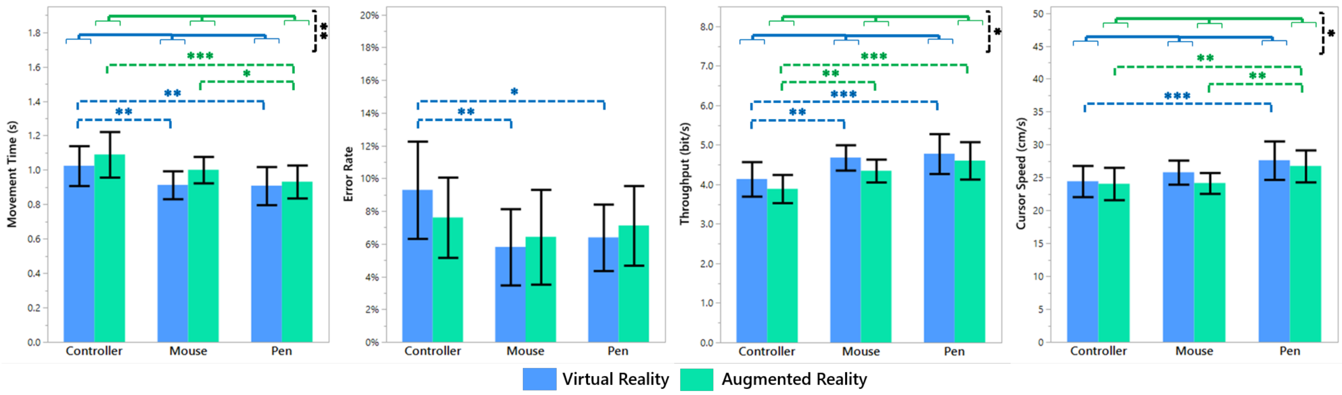

Figure 6: Objective measures. From left to right: movement time, error rate, throughput, and cursor speed of input devices in VR and AR. Error bars represent 95% confidence intervals. ($^{*}p \leq .05$, $^{**}p < .01$, $^{***}p < .001$)

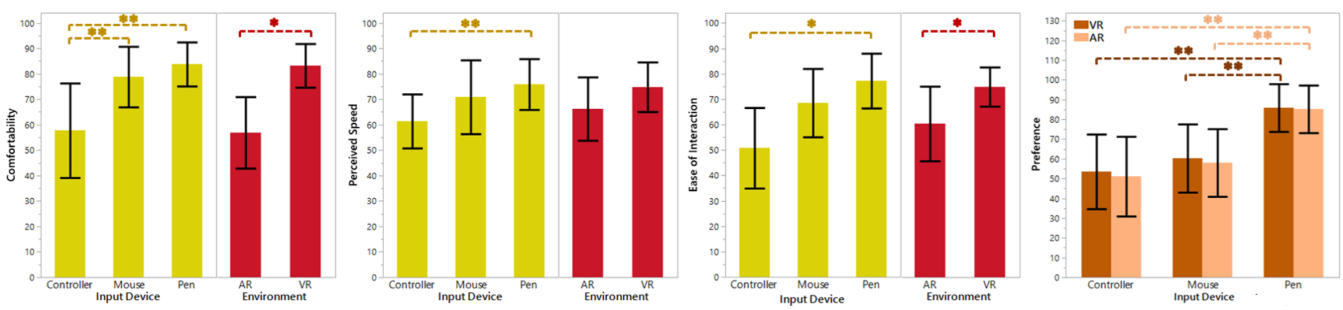

Figure 7: Subjective measures. From left to right: comfortability, perceived speed, ease of interaction, and user preference for Input Devices and Immersive Environments. Error bars represent 95% confidence intervals. ($^{*}p \leq .05$, $^{**}p < .01$, $^{***}p < .001$)

5.2 Subjective Measures

Here we present the results of the post-study questionnaires, see Figure 7. Our subjective data used a 0-100 Likert-scale. We applied Aligned Rank Transform [36] on the data before performing ANOVA.

5.2.1 Comfortability. A good Immersive Environment or Input Device for a pointing task should not only allow users to achieve high performance but also feel comfortable to use. Thus, participants were ask to rate the comfortability on a 0-100 scale for each Immersive Environment and Input Device. 0 indicates *very uncomfortable* and 100 indicates *very comfortable*. For Immersive Environment, a t-test showed that participants felt significantly more comfortable in VR than with the AR headset, $t(1) = 9.305$, $p = .011$, $\eta^2 = .458$. The Input Devices received significantly different comfortability scores, $F(2, 22) = 6.426$, $p = .006$, $\eta^2 = .369$. Post-hoc tests showed that *pen* and *mouse* were rated significantly more comfortable than *controller*, see Figure 7.

5.2.2 Perceived Movement. To validate the objective measure of cursor speed, we asked participants to rate their perception of how fast they manipulated the cursor. The rating ranges from 0-*very slow* to 100-*very fast*. A t-test indicated that there was no significant difference in perceived cursor movement speed between VR and AR conditions, $t(1) = 2.067$, $p = .178$, $\eta^2 = .158$. An ANOVA revealed a significant effect of Input Device, $F(2, 22) = 3.530$, $p = .047$, $\eta^2 = .243$. Post-hoc tests showed that participants perceived they could move the cursor significantly faster with the *pen* compared to the *controller*, see Figure 7.

5.2.3 Ease of interaction. We also recorded ease of pointing, by asking participants to rate how easy they felt that the pointing task was for each Immersive Environment and Input Device. The range is from 0-*very difficult* to 100-*very easy*. Participants found the task significantly easier in VR than in AR, $t(1) = 5.007$, $p = .047$, $\eta^2 = .313$. To compare different Input Devices, participants were asked to rate how easy they found each device for pointing. An ANOVA identified a significant main effect of Input Device, $F(2, 22) = 4.216$, $p = .028$, $\eta^2 = .277$. Post-hoc tests showed that *controller* was perceived to be significantly more difficult to use than *pen*, see Figure 7.

5.2.4 Preference. We also asked participants to rate their preference for each combination of Input Device and Immersive Environment. The value ranges from 0-*strongly not recommended* to 100-*strongly recommended*. An ANOVA revealed that Immersive Environment did not have significant impact on user preference, $F(1, 11) = .745$, $p = .407$, $\eta^2 = .063$. On the other hand, the main effect of Input Device was significant, $F(2, 22) = 9.683$, $p = .001$, $\eta^2 = .468$. The interaction was not significant, $F(1.164, 12.802) =$

.202, p = .698, η^2 = .018. Post-hoc tests revealed that the *pen* was significantly more preferred than *mouse* or *controller* in either VR or AR, see Figure 7.

6 DISCUSSION

Overall, we identified that the *pen* is an input device that has similar performance to the *mouse* for 3D pointing tasks. This is very promising as, to our knowledge, no other work has identified an input device that is usable in immersive environments and is comparable to (or non-significantly exceeds) the mouse in terms of throughput. Also, participants generally liked the pen as an input device. Our results also confirm the outcomes of previous work [32] in that the *controller* has significantly worse performance in terms of throughput than the *mouse*. In the following we discuss more specific findings and how they relate to potential limitations of our apparatus.

6.1 VR Pointing Performance is Better than AR

We found that the pointing performance was significantly better in VR than in AR in most measures. Before isolating and exploring possible causes, we present how we reduced potential confounds for the different Immersive Environments.

6.1.1 Similarities of Headset Specifications and Tracking System. To mitigate the effect of external factors the Immersive Environment, we choose the HTC Vive Pro and Meta 2 headsets for our comparison of pointing tasks in VR and AR. Both headsets have reasonably comparable fields of view, resolutions, and weight. We used them on the same computer, and we achieved an average refresh frequency for the HTC Vive Pro and Meta 2 of 90 Hz and 80 Hz, respectively. The similarity in specifications for the two headsets leads us to expect (roughly) comparable outcomes.

Early pilots made us only too aware of the differences in the headset tracking systems, with different latencies and different degrees of accuracy. To avoid these potential confounds, we chose to use an external OptiTrack system to track both VR and AR headsets as well as the *pen* and *controller*. Thus, the tracking latency and quality should be comparable for both headsets as well as the two input devices.

6.1.2 Different Display Latencies. The results of the main study revealed that in the AR condition participants took longer to point, manipulated the cursor slower, and achieved less throughput compared to VR. The most likeliest cause for this is different headset display latencies. To determine the display latencies for both headsets, we measured the delay between the movement of a physical input device and that of a virtual object shown on the device, by moving a controller in front of each headset display showing the manipulated sphere and recording both with a 240 Hz camera [24, 30]. By observing the delays in the movement in the display device in several trials, we measured the average latency of the VR condition to be 51 ms and 79 ms for AR. Given these different latencies and that previous work identified a clear effect of latency on pointing performance [24, 30], we believe that latency is a good explanation for the differences observed between the AR and VR conditions.

6.1.3 Other Differences Between Conditions. The VR headset allows users to see only the virtual scene, while the AR headset allows them to also see the real world. To reduce potential issues due to the (virtual or real) environment visible in the headset, we replicated the general appearance of the physical room in VR. Yet, there might be still some differences due to different lighting and material properties. One third of participants identified that they sometimes got distracted by real objects. One participant said "The fact that the room was [free] of objects made it easier to perform the task in VR." Another reported "It was easier to concentrate in VR without [the] real world image." While users might have been able to concentrate less in an AR environment, we point out that the pointing task is fairly repetitive. Thus, it is not surprising that participants pay more attention if they interact only with virtual content.

Participants also found the AR condition to be less comfortable. One explanation for this is that, although lighter than the Vive, the Meta 2 headset has a relatively unbalanced design, with the front being substantially heavier (and a third of participants perceived the Meta 2 as heavier).

6.2 User preferences

Interestingly, participants perceived the *pen* to be more comfortable than the *controller*. One participant even explicitly identified that "[the] pen-like device is smaller and lighter compared to the HTC [Vive] controller." Although they had never tried a *pen* in VR and AR before, they got quickly used to it, as "it felt like using your finger [for pointing]". While it achieved equivalent performance to the *pen*, the *mouse* received poor reviews from the participants. It is more familiar, but "it wasn't convenient since the movement was horizontal [on the mouse pad]" and "the cursor seemed to appear from nowhere which made it difficult [to point]".

6.3 Different Latency for the Mouse

A notable limitation is that we used a different tracking method for the *mouse* than the other two input devices. The *mouse*'s light emitter and detector track its relative movement on a surface, while the position and orientation of *controller* and *pen* are determined by OptiTrack – an external optical tracking system. As latency reduces pointing performance [24, 30], at least above an end-to-end latency of about 50 ms, this can potentially reduce pointing performance. For simplicity, we discuss only the AR condition here, results for VR are analogous.

We measured the end-to-end latency for the mouse and the pen with the AR headset and observed 55 ms respectively 79 ms, i.e., a 24 ms difference. Given that the latency for the pen was higher, a low-latency implementation of a pen should perform better than our apparatus. This means that the throughput measurements in our study form a lower bound for the pen and we expect that future implementations might perform even better.

6.4 Pen is Better than the Controller

Overall, our results indicate that a *pen* is better, i.e., "mightier", than a *controller* in all objective measurements. Although the Vive *controller* is specifically designed for VR interaction, the *pen* helped users to complete pointing tasks significantly quicker, to manipulate the cursor faster, to make fewer errors, and to reach higher throughput. 83.33% of our participants had no or only a little experience with VR and AR at the time of participation. Consequently, most of them were not familiar with a *controller* nor did they expect that one could use a *pen* in such systems.

6.4.1 User behaviors. Analyzing videos captured during the experiment, we saw that participants usually used their wrist and sometimes their arms to control the direction of the *controller*, as shown in Figure 5. On the other hand, they used the movement of at least three fingers and sometimes their wrist to manipulate the *mouse* and the *pen*, as shown in Figure 5. This matches observations from previous work on 2D input [3], but extends their results to 3D pointing. We believe the pen/precision grip using multiple fingers is the likeliest explanation for the better pointing performance of the *pen* compared to the *controller*, which is grasped with the palm. Another indication is that our *pen* is controlled similarly to how people use a *real pen* to write/draw on paper. This similarity is another explanation for higher accuracy of the *pen*, and thus higher throughput, compared to the *controller*.

6.5 Pen is Comparable to the Mouse

Our results identified no significant difference between the *pen* and *mouse* in most measurements, including task completion time, error rate, and throughput. While this lack of a significance does not mean that there is no difference (one “cannot prove the null”), we point out that the average performance of our *pen* exceeds that of the *mouse*. Given that the *mouse* is used daily as an input device, it is likely more familiar than a *controller* or *pen*. However, this potential advantage still does not yield a significantly better performance than the *pen*. Finally, participants liked the pen better than the mouse for interaction in VR and AR.

One could argue that the *mouse* is a lower-cost and more familiar solution compared to the *pen*. However, this argument is only valid if VR/AR users would always sit at a desk or at least on a chair with integrated mouse pad to afford a surface for the *mouse* to move on. Yet, the characteristics of the VE and the application scenario, such as 3D engineering, architecture, or immersive analytics, potentially require the users to stand or move around and interact with objects distributed in a large volume. In such situations, the pen, becomes quickly a more convenient and flexible option.

6.6 Applications of the Pen

The *pen* achieved an average throughput of 4.7 bit/s, compared to 4.0 bit/s of the *controller*. This difference may not have a considerable impact on most current VR and AR games where game objects are usually big enough for players to easily see and interact with. The distances of these objects from the player are also adjusted so that the users can easily point at and select them unambiguously with the *controller*. As pen input devices become more available, we expect that the design of the VEs will change to contain also smaller targets.

However, for many professional applications, that require accuracy, such as engineering, the performance of pointing devices is critical, also in VR and AR settings. Consider a user drawing a part in AR that has to match the dimensions and shape of a real object, e.g., to add a handle or to replace a broken part, which requires pointing to accurate locations and may require a device as good as the *mouse*, which is very familiar to computer users. As indicated by the results of our experiment, the *pen* is at least comparable to and potentially better than the *mouse*, 4.7 bit/s vs. 4.5 bit/s on average. The *pen* is, therefore, a promising alternative for professional applications, such as 3D engineering or immersive analytics. It enables VR and AR users to, e.g., select a specific data point on a complex graph, move a virtual part to match another, or draw a line between two points in a fast and accurate manner. The *pen* also increases user productivity and the quality of the experience relative to the *controller*, while removing the need for a desktop-like surface for operating the *mouse*.

6.7 VR/AR Pen Design Space

We chose the shape of a typical whiteboard pen to create the prototype of our VR *pen*, as this kind of pen is familiar to many people. Such a *pen* is also lighter than the *controller*, which enables people to easily control the device with three fingers [14]. In our pilot studies, we asked participants to hold the *controller* like a pen. The results showed that the *controller* was too heavy to be held comfortably in a precision grip, which meant that users could not actuate it similar to the pen, and thus that pilot did not demonstrate in significantly higher pointing performance than when held in a power grip. Hence, we believe that the weight and how it is distributed on the pointing device matters for performance. However, how exactly the weight and its distribution in a pointing device affects the performance is still unknown and needs further investigation.

The *pen* in our study did not have any button, as we wanted to avoid the Heisenberg effect [6] and to study the device’s pointing performance without this potential confound. Thus, we asked participants to confirm target selection with another device, i.e., by pressing the space bar on a keyboard. However, a more commercially viable design of the *pen* should includes buttons, although the number of such buttons may have to be smaller than the *controller* because of limited area on the *pen*’s surface. To support additional input one could add a touchpad on top of the pen, e.g., [28], see Figure 8. Alternatively, we could add a touchstrip along its length to support at least forward-backward scrolling/movements. Such a pen can then alternatively be used like a controller, simply by holding it in a different grip. This then makes a pen equivalent to a controller as a 6DOF input device. Alternatively, future 6DOF controllers could also be made slimmer, so that they could be used as both controllers and pen-like devices.

Looking at current headsets, we identified that many have surfaces where a pen could be magnetically snapped to, either to the side of the headset (say roughly at the temple) or the top, see Figure 8. This permits the user to store a pen temporarily while they are not using it, but to also quickly and easily retrieve it when they need it. A simple extension of this concept is to use two pens, one on either side of the headset. Given that a pen can also be held like a controller, as discussed above, this two-pen system then creates a system that is very similar to the two-controller setup offered by several current VR systems, e.g., the HTC Vive, but also affords precision input.

Figure 8: Illustration of a pen with hexagonal touchpad/button magnetically snapped to the top of a headset for convenient pickup and storage.

6.8 Limitations

While we used an external tracking system to reduce the effect of different tracking technologies, we recognize that this likely affected display latencies for the VR and AR conditions, as we could not benefit from the built-in motion compensation in the HTC Vive and Meta 2 headsets. Yet, such compensation systems are (mostly) targeted at reducing motion sickness during head movements. As participants sat in a chair and kept their head (relatively) stable during the experiment, we believe that the lack of motion compensation was not a main factor. Moreover, none of our participants reported motion sickness symptoms. Still, better tracking might further reduce input latencies.

In the Windows computer used for our experimental platform, mouse acceleration was turned on by default with the default settings. We did not change this, as we assumed most users are familiar with this setting. However, one may wonder if a linear mapping, i.e., without acceleration, between the *mouse* and the cursor would make a difference. Yet, this is a subject to investigate this issue in future work.

We believe that our results are not affected by the difference of the used ray casting techniques – the *mouse* uses a ray from the eye through the cursor, while the *controller* and *pen* use a ray that originates at the device. The *mouse* cursor on the plane is indistinguishable from the intersection of the eye-cursor ray with the same plane. For the *controller* and the *pen* we did not display the ray from the device in our study, only its intersection with the plane. Thus, there was no visual difference between the techniques and no participant reported that the "felt" a substantial difference between the conditions.

Our study involved only a small number of participants. Yet, we observed not only significant differences but also medium or large effect sizes for our main results. This makes us believe that our results are reasonably robust.

We asked participants to hold the *pen* in a typical pencil grip, more specifically a dynamic tripod grasp, which is only one of many precision grips. We acknowledge that there are other pen grips, e.g., where the *pen* passes underneath or over the thumb. We do not yet know if these other precision grips have different effects on the pointing performance of a *pen*.

7 CONCLUSION AND FUTURE WORK

We presented a comparison of different pointing devices with Virtual and Augmented Reality headsets. Overall, interaction in VR was faster than in AR, potentially due to higher latencies in the AR system. We also identified that a pen devices affords throughput at least as good as the mouse, with both being better than a VR controller. Also, participants liked to use a pen device in VR and AR. Finally, we presented several design options for pens in VR and AR systems.

In general, the results of our work support the introduction of pen-like devices to the VR market. We demonstrated their potential for easy-to-use and accurate distant selection in modern fully- or semi-immersive environments. Compared to the popular controller held in a power grip, the pen held in a precision grip is lighter, more accurate, and easier to control, which helps VR/AR users to achieve higher throughput and potentially smaller error rates. Though it did not outperform the traditional mouse, our study confirmed the pen's potential, which paves the way for the development of improved devices. In the future we plan to evaluate (virtual) hand-based interaction in VR and AR.

REFERENCES

[1] 2012. ISO/TS 9241-411 Evaluation methods for the design of physical input devices. (2012).

[2] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136.

[3] Ravin Balakrishnan and I Scott MacKenzie. 1997. Performance differences in the fingers, wrist, and forearm in computer input control. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, 303–310.

[4] François Bérard, Jessica Ip, Mitchel Benovoy, Dalia El-Shimy, Jeffrey R Blum, and Jeremy R Cooperstock. 2009. Did "Minority Report" get it wrong? Superiority of the mouse over 3D input devices in a 3D placement task. In *IFIP Conference on Human-Computer Interaction*. Springer, 400–414.

[5] Doug Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan P Poupyrev. 2004. *3D User interfaces: theory and practice, CourseSmart eTextbook*. Addison-Wesley.

[6] Doug Bowman, Chadwick Wingrave, Joshua Campbell, and Vinh Q Ly. 2001. Using pinch gloves for both natural and abstract interaction techniques in virtual environments. *Proceedings of HCI International* (01 2001).

[7] Michelle A Brown and Wolfgang Stuerzlinger. 2016. Exploring the Throughput Potential of In-Air Pointing. In *International Conference on Human-Computer Interaction*. Springer, 13–24.

[8] Michelle A. Brown, Wolfgang Stuerzlinger, and E. J. Mendonça Filho. 2014. The Performance of Un-instrumented In-air Pointing. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 59–66. http://dl.acm.org/citation.cfm?id=2619648.2619659

[9] Jeff Butterworth, Andrew Davidson, Stephen Hench, and Marc T Olano. 1992. 3DM: A three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 symposium on Interactive 3D graphics*. ACM, 135–138.

[10] Joanna Camargo Coelho, Niels Bohrweg, and Fons J. Verbeek. 2014. Pointing Task Evaluation of Leap Motion Controller in 3D Virtual Environment.

[11] Girden Ellen R. 1992. *ANOVA : repeated measures*. Sage Publications. https://doi.org/doi.org/10.4135/9781412983419

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

[13] Adrian Heinrich Hoppe, Florian van de Camp, and Rainer Stiefelhagen. 2017. Interaction with Three Dimensional Objects on Diverse Input and Output Devices: A Survey. In *International Conference on Human-Computer Interaction*. Springer, 130–139.

[14] Bret Jackson and Daniel F. Keefe. 2016. Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. *IEEE Transactions on Visualization*

and Computer Graphics 22, 4 (April 2016), 1442–1451. https://doi.org/10.1109/TVCG.2016.2518099

[15] Regis Kopper, Doug A. Bowman, Mara G. Silva, and Ryan P. McMahan. 2010. A Human Motor Behavior Model for Distal Pointing Tasks. *Int. J. Hum.-Comput. Stud.* 68, 10 (Oct. 2010), 603–615. https://doi.org/10.1016/j.ijhcs.2010.05.001

[16] Max Krichenbauer, Goshiro Yamamoto, Takafumi Taketom, Christian Sandor, and Hirokazu Kato. 2018. Augmented Reality versus Virtual Reality for 3D Object Manipulation. *IEEE transactions on visualization and computer graphics* 24, 2 (2018), 1038–1048.

[17] L. Liu, J. Martens, and R. van Liere. 2010. Revisiting path steering for 3D manipulation tasks. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. 39–46. https://doi.org/10.1109/3DUI.2010.5444724

[18] L. Liu, R. van Liere, C. Nieuwenhuizen, and J. Martens. 2009. Comparing Aimed Movements in the Real World and in Virtual Reality. In *2009 IEEE Virtual Reality Conference*. 219–222. https://doi.org/10.1109/VR.2009.4811026

[19] I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139.

[20] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 653, 13 pages. https://doi.org/10.1145/3173574.3174227

[21] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. 2013. 3DTouch and HOMER-S: Intuitive Manipulation Techniques for One-handed Handheld Augmented Reality. In *Proceedings of the Virtual Reality International Conference: Laval Virtual (VRIC '13)*. ACM, New York, NY, USA, Article 12, 10 pages. https://doi.org/10.1145/2466816.2466829

[22] Ji-Young Oh and Wolfgang Stuerzlinger. 2002. Laser Pointers as Collaborative Pointing Devices. *Proceedings - Graphics Interface* 2002 (10 2002).

[23] J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. 2005. TractorBeam: Seamless Integration of Local and Remote Pointing for Tabletop Displays. In *Proceedings of Graphics Interface 2005 (GI '05)*. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 33–40. http://dl.acm.org/citation.cfm?id=1089508.1089515

[24] Andriy Pavlovych and Wolfgang Stuerzlinger. 2009. The tradeoff between spatial jitter and latency in pointing tasks. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. ACM, 187–196.

[25] Katrin Plaumann, Matthias Weing, Christian Winkler, Michael Müller, and Enrico Rukzio. 2018. Towards Accurate Cursorless Pointing: The Effects of Ocular Dominance and Handedness. *Personal Ubiquitous Comput.* 22, 4 (Aug. 2018), 633–646. https://doi.org/10.1007/s00779-017-1100-7

[26] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The Go-go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology (UIST '96)*. ACM, New York, NY, USA, 79–80. https://doi.org/10.1145/237091.237102

[27] Colleen Schneck and Anne Henderson. 1990. Descriptive Analysis of the Developmental Progression of Grip Position for Pencil and Crayon Control in Nondysfunctional Children. *The American journal of occupational therapy : official publication of the American Occupational Therapy Association* 44 (11 1990), 893–900. https://doi.org/10.5014/ajot.44.10.893

[28] Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. Grips and Gestures on a Multi-touch Pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1323–1332. https://doi.org/10.1145/1978942.1979138

[29] Junwei Sun, Wolfgang Stuerzlinger, and Bernhard E Riecke. 2018. Comparing input methods and cursors for 3D positioning with head-mounted displays. In *Proceedings of the 15th ACM Symposium on Applied Perception*. ACM, 8.

[30] Robert J Teather, Andriy Pavlovych, Wolfgang Stuerzlinger, and I Scott MacKenzie. 2009. Effects of tracking technology, latency, and spatial jitter on object movement. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*. IEEE, 43–50.

[31] R. J. Teather and W. Stuerzlinger. 2008. Assessing the Effects of Orientation and Device on (Constrained) 3D Movement Techniques. In *2008 IEEE Symposium on 3D User Interfaces*. 43–50. https://doi.org/10.1109/3DUI.2008.4476590

[32] Robert J Teather and Wolfgang Stuerzlinger. 2011. Pointing at 3D targets in a stereo head-tracked virtual environment. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*. IEEE, 87–94.

[33] Robert J Teather and Wolfgang Stuerzlinger. 2013. Pointing at 3d target projections with one-eyed and stereo cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 159–168.

[34] Robert J Teather and Wolfgang Stuerzlinger. 2014. Visual aids in 3D point selection experiments. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. ACM, 127–136.

[35] Daniel Vogel and Ravin Balakrishnan. 2005. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 33–42. https://doi.org/10.1145/1095034.1095041

[36] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963 event-place: Vancouver, BC, Canada.

[37] Shumin Zhai. 1998. User performance in relation to 3D input device design. *ACM Siggraph Computer Graphics* 32, 4 (1998), 50–54.

[38] Shumin Zhai, Paul Milgram, and William Buxton. 1996. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 308–315.