Comparing Input Methods and Cursors for 3D Positioning with Head-Mounted Displays

Junwei Sun
School of Interactive Arts and Technology
Simon Fraser University
junweis@sfu.ca

Wolfgang Stuerzlinger
School of Interactive Arts and Technology
Simon Fraser University
w.s@sfu.ca

Bernhard E. Riecke
School of Interactive Arts and Technology
Simon Fraser University
ber1@sfu.ca

ABSTRACT

Moving objects is an important task in 3D user interfaces. In this work, we focus on (precise) 3D object positioning in immersive virtual reality systems, especially head-mounted displays (HMDs). To evaluate input method performance for 3D positioning, we focus on an existing sliding algorithm, in which objects slide on any contact surface. Sliding enables rapid positioning of objects in 3D scenes on a desktop system but is yet to be evaluated in an immersive system. We performed a user study that compared the efficiency and accuracy of different input methods (mouse, hand-tracking, and trackpad) and cursor display conditions (stereo cursor and one-eyed cursor) for 3D positioning tasks with the HTC Vive. The results showed that the mouse outperformed hand-tracking and the trackpad, in terms of efficiency and accuracy. Stereo cursor and one-eyed cursor did not demonstrate a significant difference in performance, yet the stereo cursor condition was rated more favourable. For situations where the user is seated in immersive VR, the mouse is thus still the best input device for precise 3D positioning.

CCS CONCEPTS

• **Human-centered computing → Interaction devices**;

KEYWORDS

input devices, cursors, 3D positioning

ACM Reference Format:
Junwei Sun, Wolfgang Stuerzlinger, and Bernhard E. Riecke. 2018. Comparing Input Methods and Cursors for 3D Positioning with Head-Mounted Displays. In *SAP '18: ACM Symposium on Applied Perception 2018, August 10–11, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3225153.3225167

1 INTRODUCTION

Posing a 3D rigid object, i.e., manipulating the position and orientation of an object, is a basic task in 3D user interfaces. This task can be time-consuming as 6 degrees of freedom (DOFs) must be controlled: 3 DOFs for translation along three axes and 3 DOFs for rotation around three axes.

Input devices are an important component in designing, developing, and using 3D user interfaces. There are many different types of input devices to choose from and some of them may be more appropriate for certain tasks than others. Despite the dominance of the mouse in 2D user interfaces [Zhai 1998], no device has been identified to be best suitable for all tasks in 3D user interfaces. For 2D desktop applications, many input devices have been designed and used; yet with proper mappings, they also work well in 3D [Bowman et al. 2004]. The mouse is still one of the most widely used devices in 2D user interfaces and people are quite familiar with the form and function of the mouse.

Typically, the mouse is not designed to be used in immersive 3D environments, since it needs to be placed on a 2D surface to function properly. Naturally, direct manipulation with the hand is a very desirable method of interaction for virtual environments. Interestingly, on (multi-touch) mobile platforms fingers are the most common input devices for 3D manipulation. Consequently, some researchers have presented methods that use touch-based techniques for 3D manipulation in virtual environments [Martinet et al. 2010].

Using an appropriate cursor representation significantly affects users' perception of the scene (and especially the cursor position) during selection. When working with stereoscopic content, the shape and behaviour of a standard 2D cursor is not always suitable, as depth conflicts between the 3D positions of the cursor and the content can lead to confusion [Argelaguet and Andujar 2009]. A one-eyed (mono) cursor, first suggested by Ware and Lowther [Ware and Lowther 1997], eliminates stereo cue conflicts by displaying the cursor only to the dominant eye. However, the lack of a cursor visible in both eyes may cause some discomfort with long-term use [Hill and Johnson 2008]. Teather et al. [Teather and Stuerzlinger 2013] found that one-eyed cursors improve screen-plane pointing/selection techniques. Yet, for positioning techniques based on ray-casting, the one-eyed cursor performed significantly worse.

Virtual reality (VR) headsets have become rapidly prevalent in recent years and are starting to be widely used with computer games. More game developers than ever are helping with the VR revolution in the game industry. The HTC Vive was officially unveiled in 2016 and has become one of the best VR headsets in the industry. The Vive includes two wireless handheld controllers. Figure 1 shows a person using the trackpad on the Vive controller. Hundreds of games and applications are now available for the HTC Vive. For

---

SAP '18, August 10–11, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5894-1/18/08…$15.00
https://doi.org/10.1145/3225153.3225167

Figure 1: The HTC Vive controller. The person's thumb is touching the trackpad.

both games and VR applications, 3D positioning is a fundamental task.

Although we know that the mouse is an ideal input device for 2D interfaces and some 3D interfaces [Zhai 1998], there has been little research that evaluated the suitability of the mouse for 3D positioning tasks in head-mounted displays (HMDs). Moreover, although the one-eyed cursor was shown to be beneficial for 3D selection [Teather and Stuerzlinger 2013], it remains to be seen whether it performs well for 3D positioning.

We are interested in the efficiency and accuracy of input methods and cursors for 3D positioning with HMDs. We choose a sliding algorithm, as it works well with a mouse in desktop systems [Oh and Stuerzlinger 2005]. As this kind of positioning algorithm requires only 2D input, it should also work well with ray-casting. With a seated user wearing an HMD, we hypothesize that the mouse would thus be faster and more accurate than 3D input devices (hand-tracking with the Vive controller) and touchpads (trackpad on the Vive controller) for precise 3D positioning. Due to the potential eye fatigue and discomfort introduced by one-eyed cursor, we also hypothesize that the stereo cursor would perform better than a one-eyed cursor for 3D positioning with HMDs.

We measure completion time and error in a 3D positioning task and collect data from a usability questionnaire. Besides the main effect of input method and cursor display, we also want to analyze if the type of scene surfaces (smooth or irregular) and object density (empty or cluttered) in the scene have a significant impact on user performance. We hypothesize that the type of surfaces would not influence user performance, as sliding is robust to irregular surfaces [Oh and Stuerzlinger 2005]. We also hypothesize that cluttered scenes would yield longer completion time than empty ones, simply because users might (need to) slide around obstacles to find a path to the target position in a cluttered scene. Analyzing the influence of scene surfaces and object density could yield some interesting insights about the differences between input techniques and should not affect the main effects of input method or cursor display.

In this paper, we first review relevant object manipulation methods with various input devices. Then we discuss the sliding algorithm. Subsequently, we describe our user study, including the implementation of sliding in the HTC Vive, with both stereo and one-eyed cursors. Finally, we discuss the results and mention potential future work.

2 RELATED WORK

There has been substantial research in the field of 3D manipulation. Butterworth et al. introduced a 3D modelling program to be used in an HMD [Butterworth et al. 1992]. They used a 6D handheld mouse as the input device. Besançon et al. compared the mouse, tactile, and tangible input for 3D manipulation [Besançon et al. 2017]. They found that the three input modalities provide the same level of accuracy, yet tangible input is the fastest. Krichenbauer et al. compared virtual reality and augmented reality for 3D manipulation [Krichenbauer et al. 2018]. They found that 3D manipulation was more efficient in virtual reality than augmented reality. They also compared the mouse and a 3D input device for 3D manipulation and found no significant difference. Hoppe et al. did a survey on various input and output devices and associated interaction techniques for 3D interaction [Hoppe et al. 2017]. Some 3/6DOF devices perform better than the mouse on specific tasks, e.g., the Control Action Table [Hachet et al. 2003], the GlobeFish and GlobeMouse [Froehlich et al. 2006]. Yet, the mouse is generally more efficient than 3/6DOF devices for accurate 3D placement, despite the lack of a third DOF [Bérard et al. 2009].

Mouse-based 3D manipulation is not without its limitations. First, simultaneous translation along all three directions is not possible, due to the 2D nature of the device. This can be compensated through proper mapping and use of constraints, e.g., [Sun et al. 2016]. Second, 3D rotations are not supported efficiently. Some techniques limit the rotations to a single axis at a given time. Others enable simultaneous manipulation of two axes, e.g., [Zhao et al. 2011].

Manipulation techniques for virtual environments can be classified as exocentric and egocentric [Poupyrev et al. 1998]. In exocentric interaction, users interact with the 3D environment from the outside of it. For example, in the World in Miniature (WIM) technique, the user interacts with a small, handheld copy of the environment [Stoakley et al. 1995]. In egocentric manipulation, the user interacts from inside the environment. Virtual hand and virtual pointer are the two main metaphors for egocentric manipulation. The Go-Go technique extends the virtual hand's reaching distance through a non-linear mapping function applied to the user's real hand extension [Poupyrev et al. 1996]. It allows direct seamless 6DOF object manipulation. Ray-casting allows users to select an object by pointing at it with a virtual (pointer) ray. Ray-casting is a good technique for object selection, but not necessarily for object manipulation. Thus, the HOMER technique used ray-casting for object selection together with hand-centred manipulation [Bowman and Hodges 1997].

Various cursor display methods have been proposed for 3D selection. The silk cursor technique used a semi-transparent volume cursor, which provided additional depth cues through occlusion [Zhai et al. 1994]. In the evaluation of the silk cursor the authors compared mono and stereo display. They found that stereo display performed significantly faster than mono display in a selection task. Vanacken et al. introduced the 3D bubble cursor, a semi-transparent sphere that dynamically resizes to only enclose the closest target [Vanacken et al. 2007]. The 3D bubble cursor was effective for

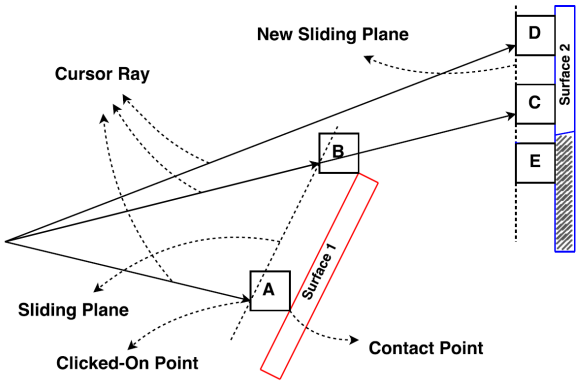


Figure 2: Illustration of sliding movements for an object across the front surfaces of two objects with an upwards mouse movement (positions A-D). The shaded part of surface 2 is occluded by surface 1. Position E can only be reached from C with a downwards mouse movement. (Figure from Sun et al. [Sun et al. 2016])

both sparse and dense environments and it outperformed the 3D point cursor in a selection task. Jáuregui et al. proposed two new 3D cursor metaphors controlled by 2D input devices: The Hand Avatar and The Torch [Jáuregui et al. 2012]. These two metaphors explored both image-based and projection-based cursor visualization techniques. The user evaluation showed that both 3D cursors significantly increased users' depth perception, but at the expense of an increase of the selection time and a decrease of accuracy.

Previous research has demonstrated that the mouse is a reliable input device for 3D manipulation in desktop systems and is well suited for the most fundamental and frequent task (object selection and placement) [Bérard et al. 2009]. However, it is important to evaluate the mouse for placement tasks in immersive systems, as little previous research has investigated this option. Various 2D/3D cursors have been proposed and evaluated for 3D selection, yet the best cursor for 3D positioning has not been identified.

Oh et al. [Oh and Stuerzlinger 2005] presented a sliding algorithm for desktop systems, with a mouse as input device, where the object follows the cursor position directly and slides on any surface behind it, i.e., the moving object always stays attached to other objects. Their user study showed that sliding was easier to understand and significantly improved the efficiency of object manipulation in CAD systems. Shift-Sliding and Depth-Pop are two recently introduced techniques that (i) generalize sliding to support floating and interpenetrating objects, (ii) address the inherent depth ambiguity, and (iii) significantly speed up common 3D positioning tasks [Sun et al. 2016]. The authors identified that the techniques could potentially be used with other input devices and platforms but did not evaluate sliding in an immersive environment. Sliding has an intuitive mapping of input to object movement; thus, we hypothesize that it would also perform well with HMDs and would be a good choice for evaluating input methods and cursor display methods.

3 SLIDING

Sliding maps object movement so that the manipulated object moves along the surface behind it that it is currently in contact with [Oh and Stuerzlinger 2005]. We use the normal vector at the contact point to determine the sliding plane. With this contact constraint, we can directly map 2D motions of the mouse cursor to 3D movement of the object. Figure 2 illustrates sliding. When the user selects an object (at position A), we record the intersection of the mouse ray on the object as the start point. The starting point and the normal vector define the sliding plane. The intersection of a new mouse ray and the sliding plane becomes the end-point of the object translation. By moving the mouse cursor, the user then effectively translates the object parallel to the sliding plane.

4 USER STUDY

We performed a user study that compared three input methods and two cursor displays for object sliding in HTC Vive.

4.1 Participants

We recruited 12 (8 female) undergrad and graduate students from the local university population. Undergraduate students received course credit for participation. We did not screen participants for 3D/VR experience. Our participants had varying gaming expertise, with 25% playing games regularly. All the participants were informed about the potential risk of motion sickness.

4.2 Apparatus

The hardware setup for the experiment used an HTC Vive (headset, base stations, controllers), mouse and keyboard, a 27-inch monitor, and a desktop computer. The Vive HMD has a diagonal field of view of 110 degrees, a display resolution of 1080x1200, and a refresh rate of 90 Hz. We used a desktop computer with 3.6 GHz i7 processor, 16 GB of memory, a NVIDIA GeForce GTX 1080 graphics card, and Unity 5.5 for the implementation. We used the monitor to observe the users' actions during the experiments.

There was a 2-minute training session before each condition, which introduced participants to the techniques in a playground environment, which did not include any version of the experimental tasks.

4.3 Implementation

We adapted the sliding algorithm to enable its use with a VR HMD. We used three different input methods: the mouse, hand-tracking with the Vive controller, and the trackpad on the Vive controller. For each input method, the user had to wear the Vive headset. We calibrated the virtual room to match the real one, so that the user could easily use the system sitting down. Full head tracking (movement and rotation) was enabled. The scale of the virtual environment was significantly bigger than the typical head movements of a user in a seated posture, therefore head movements can essentially be ignored in our experiment.

The implementation with the mouse was similar to sliding on a desktop. While wearing the HMD, the user sat in front of the desk and used the mouse (on the desk surface) to change the cursor ray, which originated from the camera position. The user pressed the left button for selection, but the space bar for confirmation of object placement.

To account for the limited size of the trackpad, we mapped the movement of the cursor ray to relative movement on the trackpad, like on a normal laptop touchpad. The dragging motion of a finger is thus translated into a relative motion of the cursor. The cursor stays static when the user is not touching the trackpad. We fine-tuned the control-display rate in a pilot study. Again, the cursor ray originated from the camera/head position. The translation of the selected object depended on the intersection of the new cursor ray with the sliding plane.

For hand-tracking with the controller, the user could move and rotate the controller freely. The implementation was slightly different, as the controller ray originates from the user's hand and continues in the pointing direction of the controller. For stereo and one-eyed cursor, we used a different cursor ray for sliding. We discuss this in the next two paragraphs. For both hand-tracking and trackpad, the user used the trigger button in the bottom of the Vive controller to select an object.

We implemented both stereo cursor and one-eyed cursor for sliding in the Vive. For all input methods, the stereo cursor was rendered as a blue sphere along the controller or cursor ray that always snaps to the scene geometry. In the stereo cursor condition, the user can see the cursor in both eyes. For the mouse and the trackpad, the cursor ray originated from the camera position. With hand-tracking, the controller ray originated from the user's hand, and we used the intersection of the controller ray and the sliding plane to determine the next position of the object.

For the one-eyed cursor condition, we only displayed the cursor to the dominant eye of the user. This condition does not provide depth cues. We again used a blue sphere as the cursor, yet in this condition the cursor stays on a plane orthogonal to the original camera direction. We placed this plane close to the camera, as we do not want the sphere to be occluded by objects in the scene. For the mouse and the trackpad, the sphere position was derived from the intersection of the cursor ray with the plane. For hand-tracking, the sphere was placed at the intersection of the controller ray with the plane. For sliding, we extended the cursor ray from the camera position to the sphere and used that during interaction.

4.4 Experiment Design

We designed a 3D object positioning experiment and asked participants to move an object to a target position in various scenes. When the user positioned the object at the target position, they pressed the space bar on the keyboard (in mouse condition) or the menu button on the Vive controller (in hand-tracking and trackpad conditions) for confirmation. In the mouse condition, participants could use their other hand to press the space bar. We measured the completion time and error distance from the ideal target position. Completion time was measured in seconds. The error measure was calculated as the absolute distance to the target over the object size. We recorded all actions of each user. The experiment had a 3 (input method) x 2 (cursor display) x 2 (surface) x 2 (object density) within-subject design. We counterbalanced the order of input method, cursor display, surface, and object density conditions to avoid learning effects. Figures 3 to 5 illustrate the three input methods.

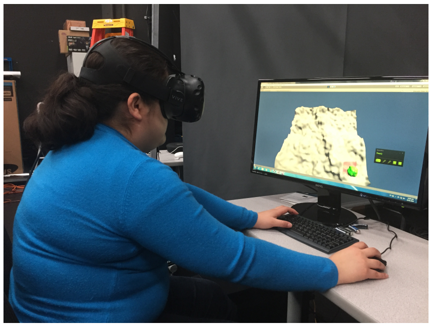


Figure 3: A user performing the task with the mouse. Note the other hand on the space bar for confirming placement.

Figure 4: A participant performing the task with hand-tracking with the controller.

Besides the main effect of input method and cursor display, we also wanted to analyze if the type of surface and object density in the scene influence user performance. Therefore, we used a 2 (surface) x 2 (object density) design for the tasks. Figure 6 shows one sample scene for each combination of surface and object density. The two scenes we used were a room and a terrain, which represented smooth and irregular sliding surfaces. There are multiple paths to move the object to the target position. In the scenes with irregular surfaces, the object often collides with the rest of the scene. The sliding algorithm then "pops" the object automatically towards the viewer to resolve the collision, which obviates the need for manual collision resolution. The two conditions for object density were empty and cluttered. In the cluttered scenes, the target position is usually partially hidden. In this case, the users had to slide around obstacles to position the object correctly.

Each task condition had 5 trials. The target positions were all in contact with the scene. We displayed the scenes in front of the users, so the users do not have to perform head rotations. This reduces any potential confound that could be introduced by such head rotations, e.g., if the user loses sight of the target position. Each user performed all trials, corresponding to a total of 120 (5x3x2x2x2) trials for each user. We asked the participants to perform the tasks as quickly and as accurately as possible.

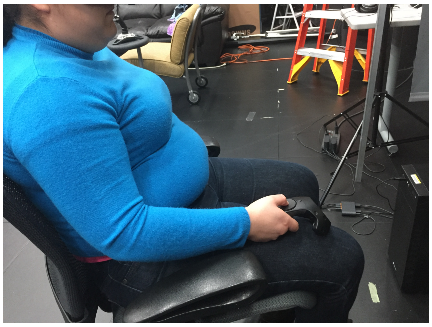
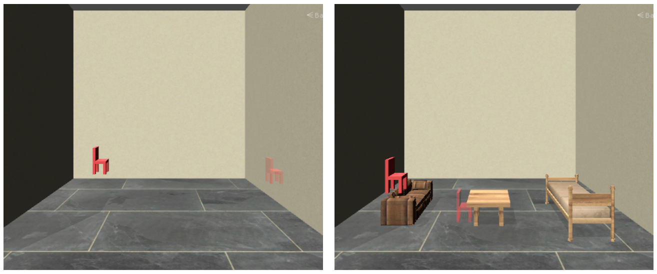
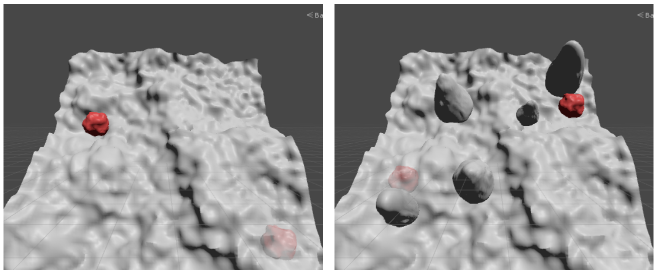


Figure 5: A participant performing the task with the trackpad on the controller.



(a) Smooth-Empty (b) Smooth-Cluttered



(c) Irregular-Empty (d) Irregular-Cluttered

Figure 6: Sample sceneS for each combination of surface and object density. The target positions are rendered as semi-transparent.

After the users finished all the tasks, we asked them to answer questions about the usability of the different input and cursor display methods. The users had to rate the ease of use, ease of learning, and level of comfort for each input and cursor display method. The ratings used a seven-level Likert scale. We also asked participants for their favourite and least favourite input and cursor display method. Finally, we asked them what could be done to improve the interaction further.

The total duration of the experiment varied from 45 minutes to one hour for each participant.

4.5 Results

Shapiro Wilk tests were conducted to assess the normality of the dataset. The results showed that the data of neither completion time nor error measure was normally distributed for any combination of input method and cursor display ($p < .001$). Therefore, we ran an aligned rank transform (ART) [Wobbrock et al. 2011] on the data followed by a within-subject ANOVA on the ranks and report the statistical results accordingly.

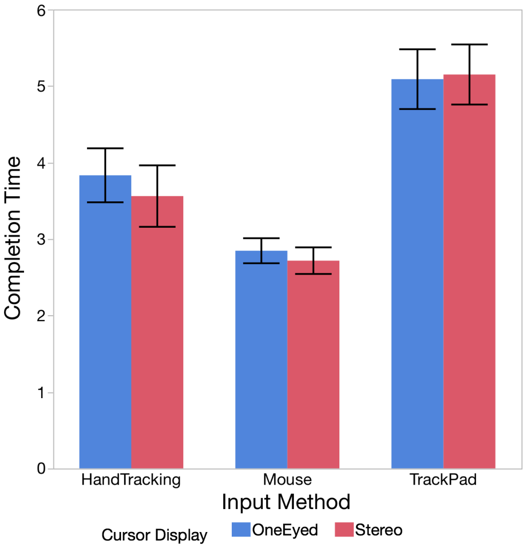


Figure 7: Average completion times (in seconds) for input and cursor display method. Each error bar is constructed using a 95% confidence interval of the mean.

The ANOVA results showed that input method had a significant effect on completion time, $F(2, 22) = 50.29$, $p < .0001$. The Tukey's post hoc analysis showed that the mouse ($M = 2.78$, $SD = 1.26$) was significantly faster than hand-tracking ($M = 3.70$, $SD = 2.82$), and hand-tracking was significantly faster than the trackpad ($M = 5.12$, $SD = 2.92$). The 95% confidence intervals of the three conditions also do not overlap. Stereo cursor ($M = 3.81$, $SD = 2.72$) and one-eyed cursor ($M = 3.92$, $SD = 2.54$) did not show a significant difference on completion time, $F(1, 11) = 1.12$, $p > .05$. The interaction of input method and cursor display was not significant, $F(2, 22) = 0.30$, $p > .05$. See Figure 7.

Surface type had a significant effect on completion time, $F(1, 11) = 19.04$, $p < .005$, where the irregular surface ($M = 3.34$, $SD = 1.76$) was faster than the smooth surface ($M = 4.40$, $SD = 3.19$). Object density had a significant effect on completion time ($F(1, 11.01) = 15.49$, $p < .005$), where the empty scene ($M = 3.68$, $SD = 2.31$) was faster than the cluttered scene ($M = 4.10$, $SD = 2.97$).

The interaction of input method and surface was significant on completion time, $F(2, 22) = 3.95$, $p < .05$, with the combinations of Mouse-Smooth, Mouse-Irregular, and HandTracking-Irregular, being significantly faster than the other three combinations, see Figure 8. The interaction of surface and object density was also significant on completion time ($F(1, 11.03) = 17.88$, $p < .005$), with Smooth-Cluttered being the slowest, see Figure 9. All the other interactions were not significant.

In terms of the error measure, input method had a significant effect, $F(2, 22) = 8.33$, $p < .005$. The Tukey's post hoc analysis showed

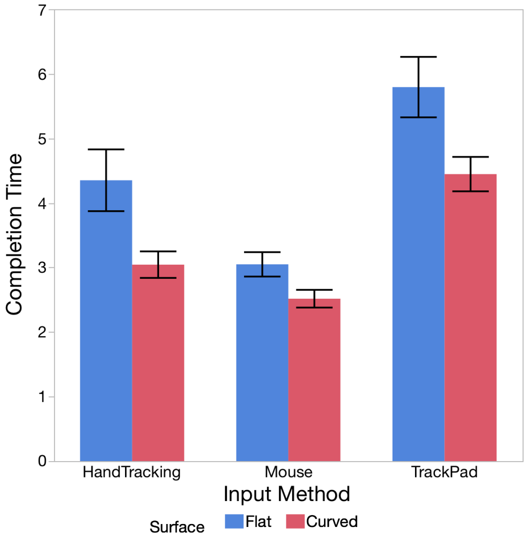


Figure 8: Average completion times (in seconds) for input method and surface type. Each error bar is constructed using a 95% confidence interval of the mean.

that the mouse ($M = 0.103$, $SD = 0.128$) was significantly more accurate than trackpad ($M = 0.152$, $SD = 0.179$). Hand-tracking ($M = 0.143$, $SD = 0.151$) was not significantly different from mouse or trackpad. Stereo cursor ($M = 0.128$, $SD = 0.147$) and one-eyed cursor ($M = 0.138$, $SD = 0.164$) did not yield a significant difference on the error measure, $F(1, 11) = 0.21$, $p > .05$. The interaction of input method and cursor display was not significant, $F(2, 22.3) = 2.42$, $p > .05$. See Figure 10.

Surface type had a significant effect on the error measure, $F(1, 11) = 52.49$, $p < .0001$, where the irregular surface ($M = 0.090$, $SD = 0.095$) was more accurate than the smooth surface ($M = 0.175$, $SD = 0.189$). The empty scene ($M = 0.127$, $SD = 0.142$) was significantly more accurate than the cluttered scene ($M = 0.144$, $SD = 0.171$), $F(1, 11.01) = 12.62$, $p < .01$.

The interaction of input method and object density was not significant for error measure, $F(2, 22.19) = 3.09$, $p > .05$. See Figure 11. The interaction of surface type and object density was not significant for the error measure, $F(1, 11.07) = 4.18$, $p > .05$. All the other interactions were also not significant on error measure.

Eleven out of 12 participants found the mouse easy to use, 9 out of 12 found hand-tracking easy to use, and 7 found the trackpad easy to use. All twelve participants found the mouse easy to learn, 10 found hand-tracking easy to learn, and 9 found the trackpad easy to learn. All participants found the mouse comfortable to use, 10 found hand-tracking comfortable to use, and only 7 found the trackpad comfortable to use.

Six participants rated the mouse as their favourite input method among the three, and 4 participants liked hand-tracking the most. 7 participants rated trackpad as their least favourite, while 5 rated hand-tracking as their least favourite.

All participants found the stereo cursor easy to use, and 10 found the one-eyed cursor easy to use. All participants found the stereo cursor easy to learn, and 11 found the one-eyed cursor easy to learn. All participants found the stereo cursor comfortable to use, and 10 found the one-eyed cursor comfortable to use.

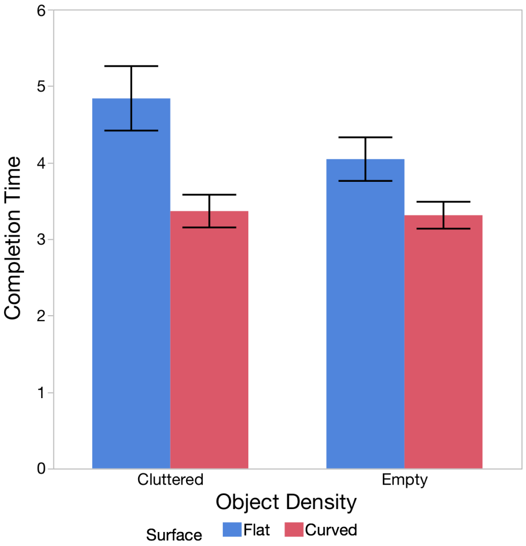


Figure 9: Average completion times (in seconds) for object density and surface type. Each error bar is constructed using a 95% confidence interval of the mean.

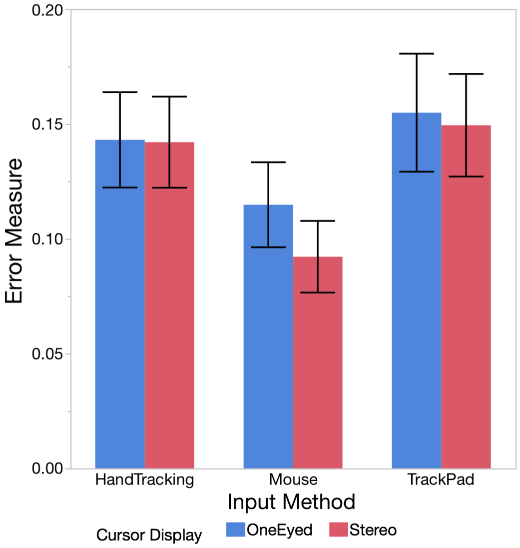


Figure 10: Average error measures for input and cursor display method. Error measure was calculated as the absolute distance to the target over the object size. Each error bar is constructed using a 95% confidence interval of the mean.

Eleven out of 12 participants rated the stereo cursor as their favourite cursor display method, and 1 participant did not have a preference.

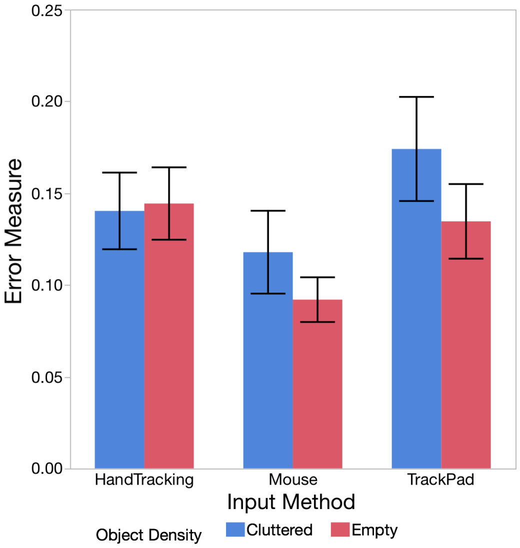


Figure 11: Average error measures for input method and object density. Error measure was calculated as the absolute distance to the target over the object size. Each error bar is constructed using a 95% confidence interval of the mean.

5 DISCUSSION

Results showed that the mouse performed the best in terms of completion time for sliding tasks in the HTC Vive, which supports our first hypothesis in terms of time. In terms of error measure, the mouse was significantly more accurate than the touchpad (but not the hand-tracking condition), which partially supports our first hypothesis. The participants commented that they liked the mouse as it is fast and precise, and they were more familiar with it as they used it daily. Our results might also be partially explainable by the fact that when using the mouse, one can rest the hand using the desk surface, which serves as a stable spatial reference. Such stabilization is not as easy with a single hand in the hand-tracking condition. Thus, it is not surprising that approximately half of the participants used the other hand to stabilize the controller pose while pointing. Interestingly, in the trackpad condition, the controller itself serves as a spatial reference for the thumb interaction. Another reason that could partially explain the mouse being fastest is that it requires less physical movement than hand movements with the controller. However, the trackpad typically requires even less physical movement, but still ended up being the slowest device.

Some participants stated that hand-tracking was easy and natural to control. However, hand jitter made it sometimes hard to position the object precisely. We observed that hand jitter can cause problems in some of our tasks, where the object had to be placed in a specific position, which required higher precision in terms of hand orientation. In this kind of situation, a subset of participants chose to stabilize the controller with both hands to be more accurate. Hand-tracking also produced higher fatigue, likely because it required on average larger hand movements.

Participants commented that it was difficult to be precise with the trackpad. They were not comfortable with using the thumb on such a small input space. Some suggested that using one-to-one position control for the trackpad might help. Yet, this would be difficult to do due the limited trackpad size of the device. Conversely, with the mouse one can use both the wrist and the fingers to accurately position the device.

Our results suggest that the mouse is a reliable input device for (precise) 3D positioning in HMDs, which matches the conclusion of Bérard et al. [Bérard et al. 2009]. Thus, there is a need for designing better user interfaces for situations, where the user can use a mouse, such as being seated or in front of a standing desk. One good alternative for a seated user is a rotatable ergonomic chair with integrated mouse pad, such as the Mobo chairs.

We choose to implement sliding as the main interaction method. Sliding is essentially a 2DOF technique, as object movement is constrained by the surface it is in contact with [Oh and Stuerzlinger 2005]. This could give an advantage to 2DOF input devices such as the mouse, where the 2DOF mouse movement is constrained by the desk surface and to use the desk as a stable reference system. Naturally, 2DOF devices provide intuitive input mappings for sliding, but note that (3D) ray-casting is also (mostly) a 2DOF input technique, corresponding to the two angular degrees of freedom used during pointing. To generalize the results, evaluation with a full 3DOF positioning technique is necessary.

Cursor display did not have a significant effect on either completion time or the error measure, which rejects our second hypothesis. However, eleven out of twelve participants preferred the stereo cursor over the one-eyed cursor, as the one-eyed cursor produced more eye fatigue. Teather et al. [Teather and Stuerzlinger 2013] showed that the one-eyed cursor benefits 3D selection. However, unlike a Fitts' Law experiment [MacKenzie 1992], we did not measure object selection time in the tasks. The timing started the instant when the users selected the object. Overall, we found that in a positioning task, different cursor display methods did not make a difference on performance. The one-eyed cursor may not be as comfortable to use, yet its lack of support for accurate depth perception does not seem to have a negative impact on the positioning time. We guess that this is likely due to the fact that users focus on the moving object during positioning tasks and not the cursor itself, as observed in previous work [Oh and Stuerzlinger 2005].

A (relatively) empty scene required less time to complete tasks than a cluttered one, which matches our hypothesis. This is not surprising, as users naturally slide objects around obstacles. The participants also mentioned that they found this to be the easiest method to reach the target position with an object.

Interestingly, tasks took longer in the smooth scenes (room) than in the irregular scenes (terrain). One potential reason is that in the room scenes, the partially hidden targets were in closer proximity to the rest of the scene, which caused more collisions. In such situations, users had to reposition the object when it was popped to the front to avoid such collisions. Still. this result shows that sliding is robust to the surfaces of the scene.

6 CONCLUSION

We compared three input and two cursor display methods for precise positioning in the HTC Vive. The mouse performed in general better than both hand-tracking and the trackpad. As the 2DOF nature of the technique matches 2DOF input devices such as the mouse, this result might partially due to the use of a sliding technique we choose for our evaluation. Yet, we believe that the result confirms that the mouse is a good input device for precise 3D positioning in an HMD-based VR system in situations where users have a stable surface for the mouse available, such as a table or a chair-integrated mouse pad.

Cursor display did not have a significant effect on either completion time or error measure. Yet, users were more comfortable using the stereo cursor for 3D positioning in the HMD. When designing an interface for HMD, both our and Teather et al.'s [Teather and Stuerzlinger 2013] results have to be taken into consideration. We presume that stereo cursor would outperform one-eyed cursor in a combined task requiring both selecting and positioning, but this needs to be verified by a future study.

Some may argue that the mouse is not suitable for immersive virtual reality. However, we believe that it is currently not practical to expect people to use an HMD in a standing pose for an extended period of time, say for a full workday. Therefore, there is a need for designing better user interfaces for users that use an HMD in a seated posture, or at least in front of a standing desk. The choice of input device depends on the tasks and platforms. With a sliding technique, the mouse is the better device. To generalize the results, we plan to evaluate positioning in a comparative study involving a full 3DOF technique.

In the future, we also plan to look at 3D positioning tasks that require bigger head or body movements. Moreover, we plan to implement Shift-Sliding and Depth-Pop for the HTC Vive to enable full 3D positioning in more general scenes [Sun et al. 2016]. To improve the precision of positioning, non-linear mappings for the trackpad are another avenue to explore. Finally, we are considering another study that records the time for the movement and fine-tuning phases separately, to better analyze how the movement is affected by the input device.

REFERENCES

Ferran Argelaguet and Carlos Andujar. 2009. Visual feedback techniques for virtual pointing on stereoscopic displays. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. ACM, 163–170.

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.

Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. 2017. Mouse, tactile, and tangible input for 3D manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4727–4740.

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

Doug A Bowman and Larry F Hodges. 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*. ACM, 35–ff.

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.

Bernd Froehlich, Jan Hochstrate, Verena Skuk, and Anke Huckauf. 2006. The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 191–199.

Martin Hachet, Pascal Guitton, and Patrick Reuter. 2003. The CAT for efficient 2D and 3D interaction as an alternative to mouse adaptations. In *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 225–112.

Alex Hill and Andrew Johnson. 2008. Withindows: A framework for transitional desktop and immersive user interfaces. In *3D User Interfaces, 2008. 3DUI 2008. IEEE Symposium on*. IEEE, 3–10.

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.

David Antonio Gómez Jáuregui, Ferran Argelaguet, and Anatole Lecuyer. 2012. Design and evaluation of 3D cursors and motion parallax for the exploration of desktop virtual environments. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*. IEEE, 69–76.

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.

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

Anthony Martinet, Gery Casiez, and Laurent Grisoni. 2010. The design and evaluation of 3d positioning techniques for multi-touch displays. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*. IEEE, 115–118.

Ji-Young Oh and Wolfgang Stuerzlinger. 2005. Moving objects with 2D input devices in CAD systems and desktop virtual environments. In *Proceedings of Graphics Interface 2005*. Canadian Human-Computer Communications Society, 195–202.

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*. ACM, 79–80.

Ivan Poupyrev, Tadao Ichikawa, Suzanne Weghorst, and Mark Billinghurst. 1998. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, Vol. 17. Wiley Online Library, 41–52.

Richard Stoakley, Matthew J Conway, and Randy Pausch. 1995. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 265–272.

Junwei Sun, Wolfgang Stuerzlinger, and Dmitri Shuralyov. 2016. Shift-Sliding and Depth-Pop for 3D Positioning. In *Proceedings of the 2016 Symposium on Spatial User Interaction*. ACM, 69–78.

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.

Lode Vanacken, Tovi Grossman, and Karin Coninx. 2007. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In *3D User Interfaces, 2007. 3DUI'07. IEEE Symposium on*. IEEE.

Colin Ware and Kathy Lowther. 1997. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Transactions on Computer-Human Interaction (TOCHI)* 4, 4 (1997), 309–322.

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*. ACM, 143–146.

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

Shumin Zhai, William Buxton, and Paul Milgram. 1994. The "Silk Cursor": investigating transparency for 3D target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 459–464.

Yao Jun Zhao, Dmitri Shuralyov, and Wolfgang Stuerzlinger. 2011. Comparison of multiple 3d rotation methods. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2011 IEEE International Conference on*. IEEE, 1–5.