

Object Speed Control with a Signed Distance Field for Distant Mid-Air Object Manipulation in Virtual Reality

Mucahit Gemici*
Concordia University

Wolfgang Stuerzlinger†
Simon Fraser University

Anil Ufuk Batmaz‡
Concordia University

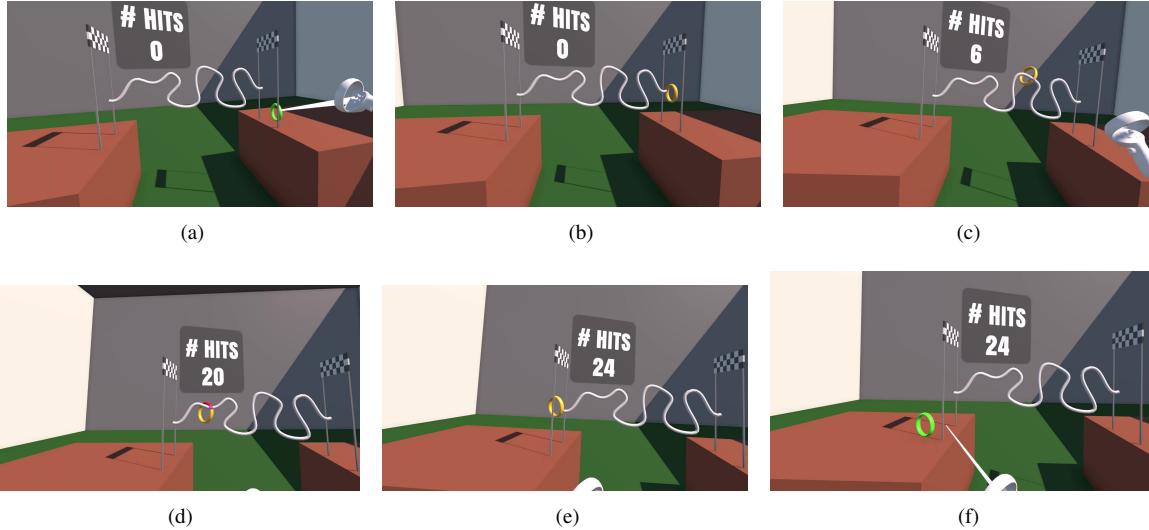


Figure 1: Interaction with Signed Distance Field (SDF): Task 3 (a) Before the start, the ring color is green and the ray is visible until the user holds the ring. (b) Data recording and experiment begin when the ring passes the starting flag. The ring color changes to yellow when the object is held. (c) Moving the ring along the wire as fast and as accurately as possible. (d) When the ring touches the wire, the contact area becomes red for 1 second as visual feedback, an error sound plays, and the error counter in the back increases. (e) Data recording ends when the ring reaches the finishing flag, and the system plays a success sound. (f) When the user drops the ring the ray becomes visible and the ring becomes green.

ABSTRACT

In Virtual Reality (VR) applications, interacting with distant objects relies heavily on mid-air object manipulation. Yet, the inherent distance between the user and the object often restricts movement precision. This paper introduces the Signed Distance Field (SDF) method for mid-air object manipulation and combines it with the ray casting interaction technique to investigate its effect on user performance and user experience. To increase movement accuracy, we leverage the speed-accuracy trade-off to dynamically adjust object manipulation speed based on the SDF algorithm’s output. Our study with 18 participants examines the effects of SDF across three different tasks with different complexity. Our results showed that ray casting with SDF reduces the number of errors in complex tasks without slowing down the participants and improves the user experience. We hope that our proposed assistive system, designed for tasks and applications, can be used as an interaction technique to enable more accurate manipulation of distant objects in fields like surgical planning, architecture, and games.

Index Terms: Signed Distance Field, Ray Casting, Mid-Air Manipulation, Accuracy, Error rate.

*e-mail: mucahit.gemici@mail.concordia.ca

†e-mail: w.s@sfu.ca

‡e-mail: ufuk.batmaz@concordia.ca

1 INTRODUCTION

In virtual environments (VEs), distant object interaction refers to methods of manipulating objects that are not within immediate reach [45]. Techniques such as ray casting, where a virtual ray is projected from the user’s viewpoint or controller to select distant objects, are commonly used. [32]. Such interactions enhance realism and immersion in VR experiences, allowing users to engage with a wide range of scenarios and interact with elements in large-scale virtual landscapes. [4, 42, 41]. A main challenge of manipulating distant objects in Virtual Reality (VR) is the lack of accurate depth cues, which hampers depth perception compared to the real world[9, 20, 62]. VR systems often fail to mimic real-world physics and perspectives accurately, leading to incorrect perception of object positions [47, 8]. Additionally, current VR tracking technologies struggle with replicating fine motor movements over longer distances due to tracking limitations, resulting in decreased interaction precision [11, 10]. All these issues negatively affect user performance and experience when interacting with objects far away.

Manipulating distant objects in VR requires balancing speed and accuracy to ensure effective interactions [13]. Accurate tracking and responses to user movements are essential to maintain precise manipulation in VR, especially for distant objects, where small errors can significantly impact the interaction and user experience [32]. It is also possible for non-accurate interactions to lead to errors and task failures [6, 5]. While accuracy tends to slow down interactions, compromising the speed of rapid-paced tasks [39], optimizing both aspects is critical for enhancing user engagement and performance in distant object manipulation in VR.

Researchers aim to balance speed and accuracy when manipulating distant objects in VR. Employed techniques include predictive modeling, assistive technologies, and haptic feedback but frequently rely on dynamic speed adjustment. Such algorithms modify the speed based on factors such as the object's distance or characteristics, reducing speed when accuracy is essential [29].

In this paper, we define ‘accuracy’ as how close the output is to the correct values, while ‘error’ quantifies the unintended action, decision, or outcome. In Human-Computer Interaction, higher accuracy implies lower error rates, and vice versa.

In this paper, we propose using a Signed Distance Field (SDF) to control an object’s speed in VR, to improve manipulation accuracy and reduce error rates. An SDF is a mathematical method to assign distances from points to surface boundaries, aiding in rendering complex shapes and scenes and other computer graphics applications [51, 31]. Our work specifically uses SDFs to measure proximity and adjust the velocity of manipulated objects in VEs.

Using the SDF provides multiple benefits for measuring the distance between a manipulated object and the environment in VR applications. An SDF can be precomputed and stored, reducing the need for speed-intensive calculations during updates. Alternatively, running the SDF algorithm independently from the VR application can also reduce computational demands. The SDF can manage complex objects with high accuracy, making it superior to other methods. These benefits led us to implement the SDF algorithm in our VR application to automatically regulate object speeds through an assistive system which slows manipulation down **when** objects get close to each other. Unlike other manipulation techniques that use scaling [63] or a varying control-to-display ratio [36], this feature enables the user to manipulate objects with the accuracy needed to accomplish the task.

In this paper, our contributions are **1)** applied SDF to control speed-accuracy trade-off during object manipulation, **2)** showed that SDF reduces the error rate of the participants without slowing participants down, and **3)** increased user experience with using SDF while controlling the speed of the objects.

2 PREVIOUS WORK

2.1 Distant Object Manipulation and Ray Casting

Distant object manipulation in VR refers to the ability to interact with and control objects that are not within immediate physical reach [45]. This capability is crucial to enhance the realism and immersion of VR experiences. Common techniques [32] include ray casting [13], Go-Go [54], and HOMER [13]. These methods expand the range of interaction beyond personal space and enable more complex and engaging VR applications [34, 28].

Challenges of distant object manipulation include several issues. First, precision and accuracy are reduced when interacting with small or distant targets. This makes it difficult to accurately select or manipulate objects at a distance [60]. Second, users often experience physical and mental fatigue due to the need for prolonged and steady hand control [52, 25]. This can lead to discomfort and decreased efficiency in tasks. Third, depth perception issues complicate determining distances in 3D, making spatial judgments challenging [12]. All these difficulties potentially affect user performance adversely. Thus, researchers, practitioners, and developers have proposed novel interaction methods based on ray casting to enhance user performance.

Ray casting is a fundamental technique to facilitate object manipulation and interaction within a VE for distant objects [13]. It involves projecting an invisible ray, typically from the user’s viewpoint or a handheld controller, into the VE [53]. Whichever object this ray intersects first is selected/grabbed and the user can then manipulate it from a distance. Despite its utility, ray casting in VR faces challenges. Precise object manipulation is often challenging

due to the abstract nature of the interaction, where slight movements can result in significant changes [12].

While in the HOMER technique, the virtual object is attached to the hand [13], here we use the RAY CASTING based object manipulation method proposed by Mine [45], where the object is attached to the end of the ray. In traditional RAY CASTING manipulation techniques the control-to-display ratio (CDR), where the physical movements a user makes (control) and how these movements are represented in the digital environment (display), is set to 1, e.g., the amount of control and display are equal and we use CDR = 1 as a benchmark condition in this paper.

2.2 Speed Accuracy Trade-off

In user interfaces the speed-accuracy trade-off refers to the balance between how quickly a user can complete tasks and how accurately they can perform them [39]. Typically, faster interactions may lead to more errors, while slower, more deliberate, actions can increase accuracy [59]. Research on new interaction methods aims to enhance performance, e.g., by reducing both speed and accuracy or by lowering one of these factors while keeping the other constant.

Prior work has investigated different interaction methods based on ray casting, assessing their effectiveness by examining the speed-accuracy trade-off. For instance, *Raycursor* reduced the error rate of the user [7]. The *Bubble mechanism* improved the accuracy of the participants in simple tasks, but not in complex ones [37]. *Ray Casting from the Eye* decreased the time and error rate of the participants [3]. The *VOTE* ray casting technique decreased the speed of the participants compared to traditional ray casting [46]. Both the *ClockRay* [64] technique and the *GazeRayCursor* [16] decreased the speed of the participants compared to ray casting. In addition to these ray-based interaction methods, other techniques, such as Depth Ray and Lock Ray [22], SQUAD [30], iSith [65], and Flexible Pointing [17], all focus on increasing user motor performance in terms of time and accuracy. This resulted in the development of a broad spectrum of interaction techniques [4].

Various manipulation techniques increase user performance for distant objects [41]. Aguerreche et al. suggested a manipulation technique with multiple users to increase task accuracy with ray casting [1]. The *PRISM* technique increases user precision by increasing the control/display ratio [18]. The *NFWBM* technique uses a replica of the object in the near-field to increase user accuracy [66]. Other techniques such as *PinNPivot* [21], *MAIOR* [44], *7-Handle manipulation technique* [48], *SIT6* [2], and separation of degrees of freedom (SDOF) [43] also focus on increasing task accuracy or precision but are targeted at being used in the near field.

Consistent with prior work, we employed the ray-based interaction technique as a benchmark and conducted a comparison between our approach and RAY CASTING to evaluate the speed-accuracy trade-off in our system with different tasks.

2.3 Signed Distance Field (SDF)

The Signed Distance Field (SDF) is a mathematical tool used in computer graphics to represent shapes in a continuous space [55]. It provides the minimum distance from any point in space to the surface of a shape, with the sign indicating whether the point is inside or outside the shape [51]. This representation enables efficient rendering techniques and is particularly useful in real-time rendering and physics simulations. The SDF offers advantages in terms of both performance and visual quality, enabling complex effects such as soft shadows and ambient occlusion [15, 26, 61].

$$d(x, \partial\Omega) = \inf_{y \in \partial\Omega} d(x, y) \quad (1)$$

$$f(x) = \begin{cases} d(x, \partial\Omega) & \text{if } x \in \Omega \\ -d(x, \partial\Omega) & \text{if } x \notin \Omega \end{cases} \quad (2)$$

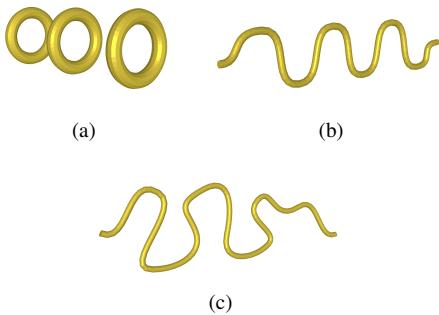
In addition, the SDF is used in collision detection systems in video games and VR, providing a fast and accurate method to manage interactions between objects [35]. Its mathematical properties also facilitate operations such as blending and morphing of shapes, making them invaluable in procedural generation and animation [49, 57]. It is relatively easy to detect the closest points between two objects through the SDF since it provides the minimum distance from a point in space to a surface. This distance information is useful to control the speed of an object and to reduce the number of hits during manipulation.

3 MOTIVATION AND HYPOTHESES

In distant object manipulation in virtual reality, it is challenging to control the object due to the (relative) lack of depth cues, increased control-display ratios, and technical limitations, such as jitter. These challenges can detrimentally affect users' accuracy while manipulating distant objects. Leveraging the speed-accuracy trade-off, reducing the users' speed during object manipulation in proximity to other objects can increase their accuracy, i.e., reduce the number of hits. Given the efficiency and accuracy of the SDF algorithms in measuring distances between objects, we can thus use the SDF to determine how close objects are to each other and control the speed of the object accordingly. In this paper, we hypothesized that according to the speed-accuracy trade-off **H1**. *Using the SDF reduces the number of hits, i.e., increases the accuracy of the participants.* Given the anticipated reduction in errors through the use of the SDF algorithm, we also expect it to enhance the user experience. Thus, we also hypothesize that **H2**. *Employing the SDF algorithm to control the speed-accuracy trade-off of the manipulated objects enhances the overall user experience.*

4 SIGNED DISTANCE FIELD TO CONTROL THE SPEED OF MOVING OBJECTS

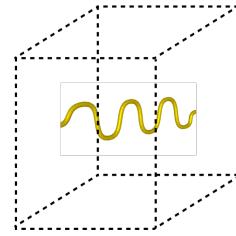
We perform SDF calculations using a Python binding of the C++ library ‘libigl’ [27]. The 3D models for each task were transferred to the Python environment in obj file format. The shapes for each task are shown in [Figure 2](#).



[Figure 2](#): Three-dimensional models of the objects in the environment for (a) Task 1 (Torus) (b) Task 2, and (c) Task 3.

We generated a 3D volume that covers the area around the environmental object(s) through a 3D array as shown in [Figure 3](#). Each point in this array represents a position around the environmental object(s) and the SDF is computed for each position in the array. We call this the *Interaction Volume* for the rest of this paper.

The SDF pre-computation method takes three parameters as input; the first parameter is a list of locations where the SDF is to be calculated, i.e., the regular grid of locations within the Interaction Volume, and the second and third parameters take model information, such as the list of vertex positions and face indices. The number of vertices in complex 3D models can range vary widely



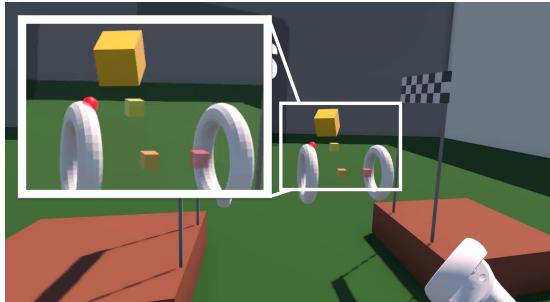
[Figure 3](#): Visual representation of Interaction Volume around the environmental object of Task 2.

and the resolution of the volume required to cover the whole object may need to be increased accordingly. Due to these reasons, we used a location spacing of 1 cm for the Interaction Volume to ensure real-time performance of the VR application. We tested 10 cm, 1 cm, and 1 mm and observed that while 1 mm causes performance problems in the VR application, 10 cm suffers from poor resolution of the Interaction Volume. Also, increasing the SDF resolution increases the array size polynomially, causing increased search time for pre-calculated values. The SDF pre-computation method outputs three parameters: an (associate) array of the smallest signed distances indexed by (and for each) position in the Interaction Volume, face indices corresponding to the smallest distances, and the closest points on the surface of the environmental object(s). The resulting SDF Results, Closest Points, and the generated array (Interaction Volume) were saved as text files and transferred to the VR application development environment, i.e., Unity. Loading these files then enables us to simply look up SDF values at any point around the environmental object(s) in real-time, by simply indexing the corresponding location in the SDF array.

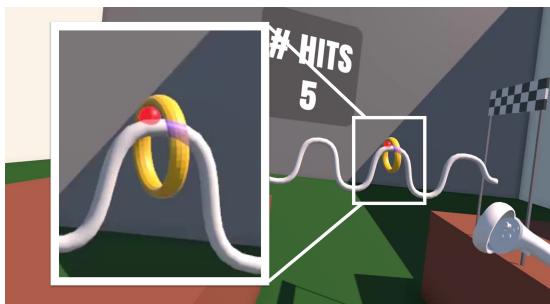
As illustrated in [Figure 5](#) the SDF takes the whole (potentially highly complex) environment into consideration when detecting the distance between the manipulated object's surface and the environment using the object's vertex positions.

The SDF results were used for controlling the manipulation speed of the objects such that lower SDF values resulted in slower manipulation. The system finds the correct SDF results for each vertex of the manipulated object by mapping the vertex world positions of the manipulated object to the Interaction Volume which yields a list of SDF Results and Closest Points. Then, the minimum SDF value is then used to control the speed. Taking the minimum SDF causes the speed to be controlled based on the closest distance between the surface of the manipulated and the environmental objects. We use the minimum SDF value to linearly interpolate the manipulation speed scale factor between the minimum (0.05) and maximum (1.0) to decrease the tracked velocity during object manipulation. The change in speed, which is the same for all three tasks, is illustrated in [Figure 6](#). Upon collision with the environmental object(s), the SDF becomes 0, but we set the speed scale factor to 0.05 to prevent the user from getting ‘stuck’. We used ray casting as the base manipulation technique and added our speed control system on top of the Ray casting technique. When the manipulated object was not inside the Interaction Volume, the speed factor was set to its maximum value (1.0) because the distance between the manipulated and environmental object(s) was large enough to be collision-free. On the other hand, if the manipulated object enters the Interaction Volume, our system finds the minimum SDF and reduces the manipulation speed increasingly when the surface of the manipulated object gets closer to the environmental surface(s).

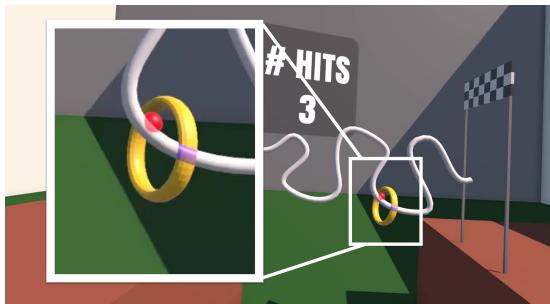
The Closest Point positions calculated from the SDF method is used for finding the closest positions on the surface of the environmental object(s) to the manipulated object. The Closest Point implementation for each task is illustrated in [Figure 4](#). The red



(a)



(b)



(c)

Figure 4: Red spheres represent the closest point on the environmental objects of (a) Task 1 (Torus), (b) Task 2, and (c) Task 3 to the manipulated objects' surface. Red spheres' positions were set by using the Closest Points calculated by the SDF.

sphere on the surface of the environmental object(s) was only used for debugging purposes and not shown during the user studies.

During object manipulation, we track the error rate as the number of hits made onto the environmental object(s). Due to the jitter in the tracking system and instability in the Unity collision detection system, we introduced a timeout interval of 100 ms for sampling the number of hits, to avoid tens or hundreds of incorrect errors occurring within the timespan of one second.

5 USER STUDIES

5.1 Apparatus

The experiment was conducted on a desktop PC with an Intel(R) i7-12700F processor at 2.1GHz, 16 GB RAM, and an NVIDIA GeForce RTX 3070 graphics card. As the VR Head Mounted Display, we used a Meta Quest 2. The 3D models in the VE were designed using Blender 4.0. The VR system was implemented using Unity 2022.3.16f1 with Unity XR Interaction Toolkit 2.0.2.

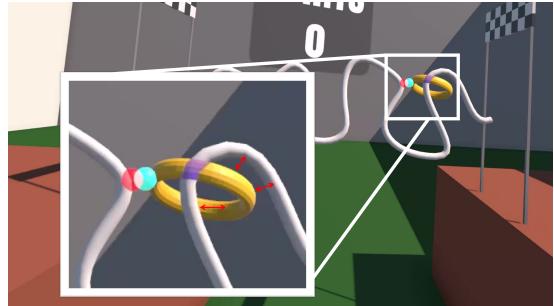


Figure 5: The red sphere shows the closest point on the environmental object and the blue sphere the closest point on the manipulated object. While the distance between the two spheres is the smallest distance between the manipulated and environmental object(s), red arrows illustrate three more example distances calculated by the SDF. We perform this calculation for the whole environment but only a few examples are shown in this figure to demonstrate the capability of the SDF to handle complex environments.

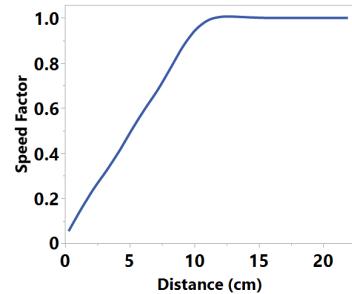


Figure 6: Speed Profile of the system, showing the relationship between Manipulation Speed and Distance. Distance is the closest distance between the surfaces of the environmental object(s) and the manipulated one.

5.2 Participants

We recruited 18 participants with ages ranging between 22 and 39 ($M = 26.44$, $SD = 4.537$. all right-handed). We invited people by posting online fliers and sending emails to people in the department, faculty, and the university. We also reached out to the general public for data collection, such as recruiting people from outside of the university. Out of 18 participants, 3 were female, and 15 were male. All participants either had normal vision or corrected-to-normal vision. 9 participants mentioned that they used VR more than 10 hours before, 4 participants between 0 – 2 hours, 2 participants 2–4 hours, 1 participant 8–10 hours, 1 participant 6–8 hours, and 1 participant mentioned 4–6 hours. For computer use, 10 participants mentioned that they use computers 8+ hours in a day, 5 participants 4–8 hours, and 3 participants mentioned 0–4 hours. Among our participants, 6 participants use mobile phones for 0–2 hours, 10 participants 2–4 hours, and 2 participants use them 4–10 hours in a day. Regarding their gaming experience, 4 participants mentioned that they play computer games 2–6 hours and 14 participants mentioned 0–2 hours in a day.

5.3 Task-1

Task 1 involves manipulating a box following a predefined path represented by small transparent cubes (target points) with the minimum number of hits with the environmental objects, i.e., three tori. To avoid issues caused by depth perception, all the target points in the path share the same distance from the user. The participant in-

teracts with a green box. As soon as the green box was selected and held, its color was changed from green to yellow. We also showed the next three target points of the path to the user. The path representation is shown in Figure 7(a).

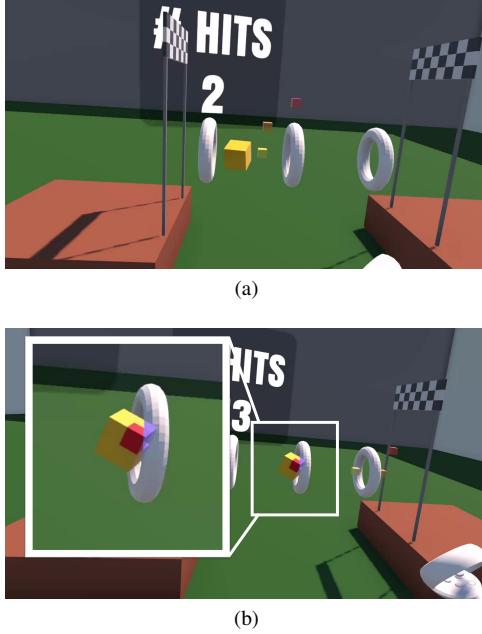


Figure 7: Task-1 (a) The user manipulates the box by following the pre-defined path consisting of multiple target points. Each time the box reaches the next target point, the system plays a success sound, hides the reached point, and then shows the next set of points. (b) The box's surface that touched the torus became red for 1 s. The area behind the torus was highlighted in transparent blue.

When the participant passed the box through the starting flag, the system played a sound indicating the start of the task and started recording data. As the participant reached each target point, that target point disappeared, a success sound played, and the next target points were shown to the participant. The color of the first target point in the path was always transparent yellow to show participants where they needed to move the cube. The second (transparent orange) and third (transparent red) target points were shown to the participants to indicate the future path, to avoid them spending time checking or deciding upon future targets. If the participant touched a torus, the system incremented the number of errors by one and played an error sound. Moreover, the color of the manipulated object's surface that touched the torus became red for 1 s as shown in Figure 7(b). Further, the surface area of the cube occluded by the torus was highlighted in transparent blue to give the participant clear visual feedback. Task-1 was repeated 5 times with the RAY CASTING condition and five times with the RAY CASTING WITH SDF condition. The dimensions of a single torus were 37.5 x 37.5 x 7.5 cm. The width of the hole inside the torus was 22.5 cm. The space between tori was 40 cm and they were 1.5 m away from the participant. The dimensions of the manipulated object, i.e., the cube, were 12.5 x 12.5 x 12.5 cm.

5.4 Task-2

Task-2 is a buzz wire task where the participants were asked to manipulate the position and rotation of the ring located initially at the right until reaching the flag located at the left of the scene. Participants were asked to finish the task with the minimum number of errors and as fast as possible. We replicated the wire shape of Luo

et al. [38] to use as an environmental object in Task-2 and evaluate the RAY CASTING WITH SDF condition. The wire shape for Task-2 is shown in Figure 8(a).

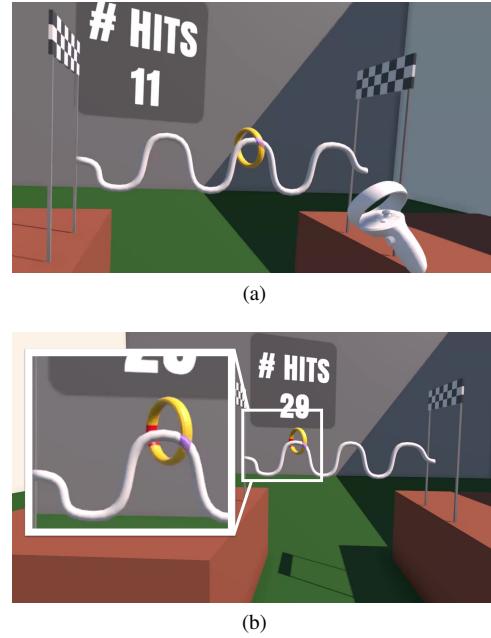


Figure 8: Task-2 (a) The user manipulates the ring through the wire. (b) The area of the ring that touched the wire became red for 1 s and the wire area occluded by the ring was highlighted in a transparent blue color.

At the beginning of the task, the color of the ring was green. When the ring was held by the participant, its color was changed to orange. As the ring passed the starting flag, a sound effect indicated the beginning of the data recording. If the ring touched the wire at any point during the task, the error counter incremented by one together with an error sound. Moreover, the surface area of the ring that touched the wire became red for 1 s as shown in Figure 8(b) to show the error and encourage users not repeat it. The wire surface area occluded by the ring was highlighted in transparent blue to enable participants to see the manipulation process clearly. When the manipulated ring reached the finishing flag, the task ended with a success sound. For RAY CASTING WITH SDF, when the ring got closer to the wire, the system reduced the manipulation speed but increased it as the ring moved away from the wire. The dimensions of the buzz wire were 0.3 x 1.7 m with a wire width of 4 cm, while the ring dimensions were 25 x 25 cm with a width of 5 cm. The size of the hole inside the ring was 20 cm and the wire was 1.8 m away from the participant. Participants were asked to repeat this task 5 times with the RAY CASTING condition and 5 times with the RAY CASTING WITH SDF condition.

5.5 Task-3

Task-3 (Figure 1 and Figure 9) was designed to test the performance of our system with an even more complex shape compared to Task-2. Everything else was identical, including the task conditions and repetitions (Figure 1). To increase the complexity of the task, we randomized the shape of the wire. We also designed the wire to minimize the need for depth manipulation. This task requires both precise movement and rotation to be able to complete the wire with a low error rate. For Task-3, the wire dimensions were 0.5 x 1.8 m with a wire width of 4 cm, while the ring dimensions were identical to Task-2.

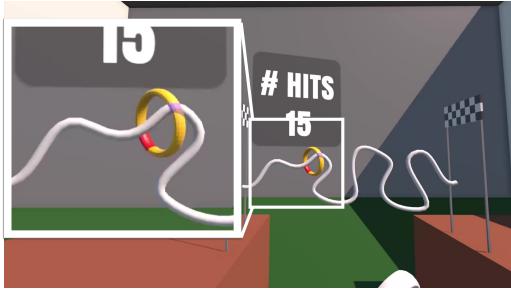


Figure 9: Task-3. When the ring touches the wire, the ring area touching the wire becomes red for 1 s. The highlighted wire area occluded by the ring was shown in a transparent blue color.

5.6 Experimental Design

We used a within-participants experiment design with 2 Interaction techniques ($IT = \{\text{RAY CASTING, RAY CASTING WITH SDF}\}$). To avoid order effects, we used counterbalancing with a Latin Square. As the differences between the 3 tasks ($T = \{\text{Task 1, Task 2, and Task 3}\}$) are entirely predictable, we did not investigate them as an experimental factor but still counterbalanced them with a separate Latin Square.

Due to the single-factor experiment design, T-tests and Wilcoxon signed-rank tests were used for data analysis (instead of RM ANOVA and Friedman tests). For each task, we recorded the following parameters: time(s), and number of hits. At the end of each task, we asked participants to fill out the NASA TLX and SUS questionnaires.

6 RESULTS

The data were analyzed using SPSS 24. We considered it to be normally distributed if Skewness (S) and Kurtosis (K) were within ± 1 [23, 40]. The figures show the mean in the graphs, with error bars representing the standard error of the mean. We used t-tests to analyze time and the average number of hits, and the Wilcoxon Signed-Rank test to analyze SUS and NASA-TLX results.

6.1 Time and Average Number of Hits Results

Task-1 According to Task-1's results, we did not observe any significant difference between RAY CASTING and RAY CASTING WITH SDF in terms of time and average number of hits results. The results are shown in Table 1 and Figure 10.

Table 1: Task-1 time and number of hits results

	Analysis	RAY CASTING	RAY CASTING WITH SDF
Time	$t(17)=1.353$, $p=0.194$, $d=0.319$	17.37 ± 6.02	18.77 ± 3.99
Number of hits	$t(17)=0.241$, $p=0.813$, $d=0.057$	6.92 ± 2.47	6.82 ± 1.94

Task-2 The results for Task-2 indicate that the error rate was lower when participants used the RAY CASTING WITH SDF method compared to the standard RAY CASTING technique. Still, there was no significant difference in the time for both methods. These results are shown in Table 2 and Figure 10.

Task-3 The results for Task-3 indicate that the average number of hits was lower when participants used the RAY CASTING WITH SDF method compared to the standard RAY CASTING. Yet, there was no significant difference in the time for both methods. These results are shown in Table 3 and Figure 10.

Table 2: Task-2 time and number of hits results

	Analysis	RAY CASTING	RAY CASTING WITH SDF
Time	$t(17)=0.926$, $p=0.368$, $d=0.132$	29.44 ± 8.39	30.40 ± 6.39
Number of hits	$t(17)=3.194$, $p<0.01$, $d=0.753$	30.06 ± 15.09	22.12 ± 11.79

Table 3: Task-3 time and number of hits results

	Analysis	RAY CASTING	RAY CASTING WITH SDF
Time	$t(17)=0.926$, $p=0.368$, $d=0.218$	43.9 ± 10.42	47.09 ± 18.36
Number of hits	$t(17)=3.685$, $p<0.01$, $d=0.868$	45.54 ± 21.83	30.76 ± 18.35

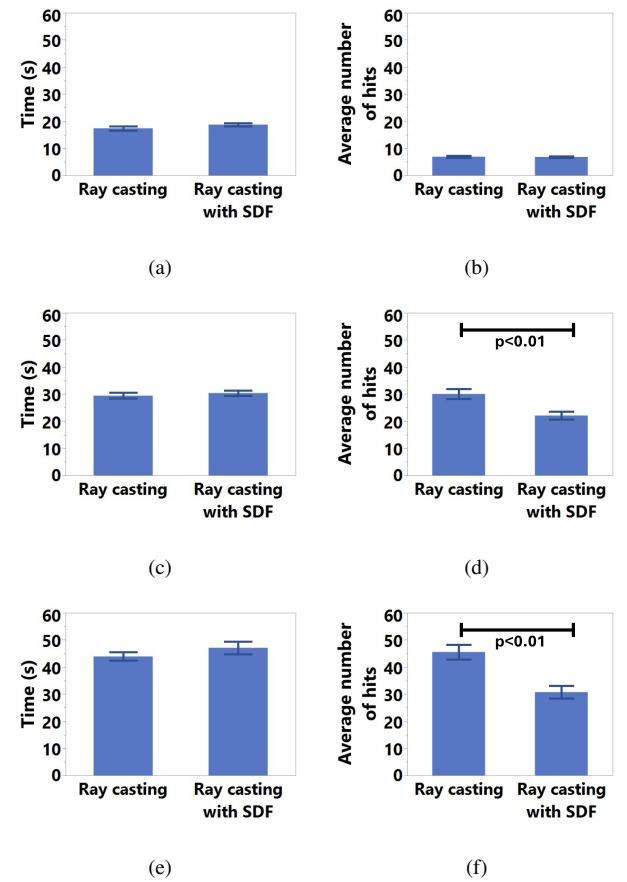


Figure 10: Task 1 (a) time results and (b) average number of hits results. Task 2 (c) time results and (d) average number of hits results. Task 3 (e) time results and (f) average number of hits results.

6.2 User Experience Results

6.2.1 System Usability Scale Results

Based on the SUS scores presented in Table 4, there was a decrease in the SUS for the RAY CASTING condition with increased task difficulty. In Task 3, we observed an average SUS score of 63.75 (Grade = D) for RAY CASTING, which is a *Poor* rating. On the other hand, the average SUS score for RAY CASTING WITH SDF condition for Task 3 was 75.9 (Grade = B), which is a *Good* rating.

For Task 1 we did not observe significant differences for each

Table 4: SUS Scores and Grades

	RAY CASTING	RAY CASTING WITH SDF
Task 1	83.05, Grade = A	84.86, Grade = A
Task 2	73.42, Grade = B	79.44, Grade = B
Task 3	63.75, Grade = D	75.97, Grade = B

SUS question. For Task 2 (Figure 11(b)) we observed a significant difference for Q1-“I think that I would like to use this system frequently.” question ($Z = -2.126$, $p < 0.05$). According to the results in Figure 11(b), participants would like to use the RAY CASTING WITH SDF condition (4.05 ± 0.72) more than the RAY CASTING condition (3.99 ± 0.91). Moreover, we found a significant difference for Q6-“I thought there was too much inconsistency in this system.” question ($Z = -2.667$, $p < 0.01$). According to the results, participants thought there was too much inconsistency in the RAY CASTING condition (2.61 ± 0.84) compared to the RAY CASTING WITH SDF condition (1.88 ± 0.9). We also found a significant difference for Q10-“I needed to learn a lot of things before I could get going with this system.” question ($Z = -2.530$, $p < 0.05$), indicating that participants thought they needed to learn a lot of things with RAY CASTING (1.61 ± 0.84) compared to RAY CASTING WITH SDF (1.16 ± 0.38).

In Task 3 (Figure 11(c)) we observed a significant difference for the Q1-“I think that I would like to use this system frequently.” question ($Z = -2.683$, $p < 0.01$). According to the results in Figure 11(c), participants would like to use the RAY CASTING WITH SDF (3.778 ± 0.87) more than the RAY CASTING condition (2.944 ± 1.16). Moreover, we found a significant difference for the Q2-“I found the system unnecessarily complex.” question ($Z = -2.807$, $p < 0.01$). According to the results, participants found the system unnecessarily complex with the RAY CASTING (2.61 ± 1.09) compared to RAY CASTING WITH SDF (1.94 ± 0.72). Furthermore, we found a significant difference for the Q6-“I thought there was too much inconsistency in this system.” question ($Z = -2.299$, $p < 0.05$). According to the results, participants thought there was too much inconsistency in the RAY CASTING (2.72 ± 1.36) compared to RAY CASTING WITH SDF (2.11 ± 0.75). We also found a significant difference for the Q8-“I found the system very cumbersome to use.” question ($Z = -1.988$, $p < 0.05$). Based on the results, participants found the system very cumbersome to use with RAY CASTING (2.55 ± 1.04) compared to RAY CASTING WITH SDF (1.88 ± 0.75). Further, we observed a significant difference for the Q9-“I felt very confident using the system.” question ($Z = -2.627$, $p < 0.01$). According to the results, participants felt more confident in the RAY CASTING WITH SDF (3.66 ± 1.02) compared to RAY CASTING (2.83 ± 1.04).

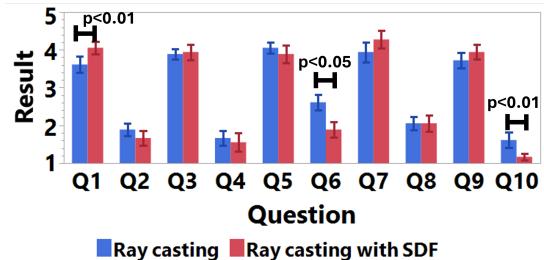
6.2.2 NASA TLX Results

For Task 1 we did not find significant differences in terms of NASA TLX scores. For Task 2 (Figure 12(b)), we found significant differences for mental demand ($Z = -3.108$, $p < 0.01$), temporal demand ($Z = -2.196$, $p < 0.05$), performance ($Z = -2.174$, $p < 0.05$), frustration ($Z = -2.421$, $p < 0.05$), and overall score ($Z = -2.809$, $p < 0.05$). The results revealed significantly more negative outcomes for RAY CASTING compared to RAY CASTING WITH SDF.

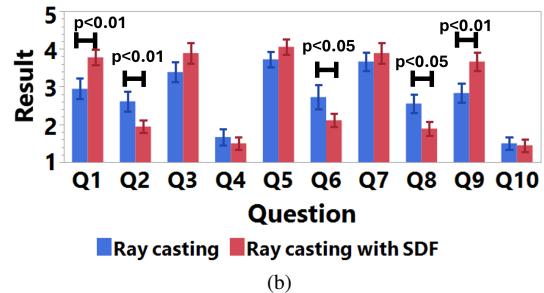
For Task 3 (Figure 12(c)) we found significant differences for mental demand ($Z = -2.108$, $p < 0.05$), physical demand ($Z = -2.089$, $p < 0.05$), temporal demand ($Z = -2.253$, $p < 0.05$), and frustration ($Z = -2.231$, $p < 0.05$). Similar to Task 2, the results revealed significant negative answers for RAY CASTING compared to RAY CASTING WITH SDF condition.

7 DISCUSSION

In this paper, we propose using the SDF to control an object’s speed based on the distance to nearby objects to increase a user’s accuracy

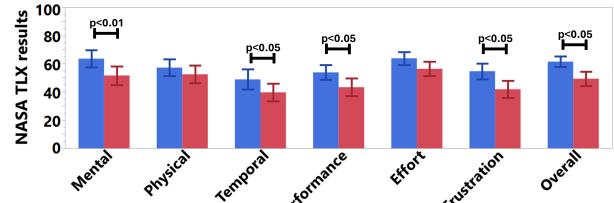


(a)

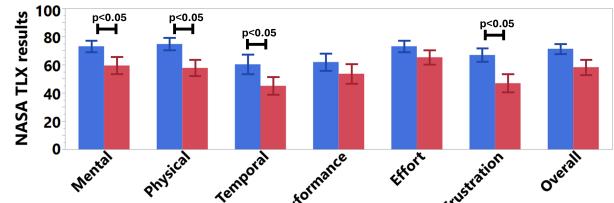


(b)

Figure 11: SUS results for (a) Task 1, (b) Task 2, and (c) Task 3. SUS questions are given in the supplementary materials.



(a)



(b)

Figure 12: NASA TLX Results for (a) Task 2 and (b) Task 3. The NASA TLX questions are given in the supplementary materials.

and their experience during distant object manipulation. We evaluated the performance of our approach with three different tasks.

The execution time of the participants did not change across all three tasks. However, we observed a significant decrease in the average number of hits for Task 2 and Task 3, where participants had to move a ring over a wire. This result shows that when the task is complex, using the SDF algorithm to control the speed of the manipulated objects significantly decreases the average number of hits. This outcome supports our hypothesis **H1**. *Using the SDF reduces the number of hits, i.e., increases the accuracy of the par-*

ticipants. For simple tasks, the SDF method does not seem to have any impact on user performance.

When we analyzed the SUS and NASA TLX results, we observed the superiority of our proposed approach compared to RAY CASTING for Tasks 2 and 3. Regarding the SUS, RAY CASTING’s grade decreased when the tasks became more complex, i.e., in Task 3. This result highlights that with the SDF the speed-accuracy trade-off improves the systems’ usability for complex tasks. Further, when we looked at the SUS questions, we observed that participants preferred to use RAY CASTING WITH SDF more than the RAY CASTING (Q1: “I think that I would like to use this system frequently”). Moreover, participants thought that there were more inconsistencies with RAY CASTING compared to RAY CASTING WITH SDF (Q6: “I thought there was too much inconsistency in this system”). When we look at the NASA TLX results, we observe that the RAY CASTING WITH SDF decreases the mental load, temporal load, and frustration of the users for complex tasks. Moreover, several participants commented on feeling better with the SDF, with one specifically noting that the system felt easier and caused much less frustration. These findings confirm our hypothesis **H2** that *employing the SDF algorithm to control the speed-accuracy trade-off of the manipulated objects enhances their overall user experience*.

The experimental design of this study investigated only a single factor, SDF ON/OFF, which was counterbalanced. The task order was only varied to avoid order effects. We may re-consider this experimental design in future work.

In this paper, we pre-generated the SDF with a resolution of 1 cm with Python. We considered calculating the SDF in real-time with Python, but the communication effort introduced noticeable delays and we abandoned this approach. Implementing the SDF functionality directly in the game engine, e.g., Unity might enable us to increase the resolution, by calculating SDF results in real-time without the need for pre-calculations. Another alternative is to use an adaptive SDF where the resolution changes based on the environmental complexity.

The results of this work can be used in various applications. As Ray casting with the SDF reduced the average number of errors without affecting the execution time any application scenario where users need to perform precise movements, such as surgical planning, virtual assembly tasks, or architectural design, could benefit from it. Moreover, the NASA TLX results show that Ray Casting with SDF reduced the mental and temporal load of the users for complex tasks. Thus, applications or interfaces where users experience cognitive load during interaction, such as complex data visualizations, could benefit from integrating the proposed method. Also, our method could improve the usability of existing systems.

Previous work demonstrated that the use of distances to control travel speed for 3D navigation [19, 14, 58, 33] improves user performance and accuracy. Yet, navigation in 3D user interfaces and object manipulation are not the same. In 3D object manipulation, the goal of the user is to change the position or orientation of an object rather than moving the user’s viewpoint. Moreover, previous interaction techniques, such as scaled HOMER [63], PRISM [18] or Oshawa et al.’s two-hand interaction technique [50], aim to control the speed of the object by varying control-display ratios, but not based on the density and complexity of the environment. In our approach, the SDF is automatically enabled when an object is nearby, controlling the speed of the manipulation based on the environment. The main idea is to use the distance between the manipulated object and the environmental object(s) and to use this distance to increase the efficiency of the manipulation. With our approach, we can quickly and efficiently measure distances from various points, such as vertices, edges, and faces. Because we use vertex positions, we can easily derive the distance information for any point on the surface. This approach takes the local environment complexity into consideration, which is different than changing the

speed globally through a CD Ratio.

In this paper, we also compared our proposed method with the RAY CASTING manipulation technique. There is extensive literature on ray-based selection techniques [4], yet our proposed method is focused on manipulating objects, not selecting them. We hope that our proposed method can be combined with other selection techniques, e.g., [37, 7], or other object manipulation techniques [54, 13, 18, 21, 44] to increase user performance and experience during task execution.

We designed Task 1 to be simple, to check whether our approach is working. For Task 2, we decided to use a task from the literature, which is easy to replicate and requires precise movement. Task 3, which is the most complex one, challenged the participants to see how our system performs in more complex scenarios. For this task, results were more clearly differentiated, as the error rate difference between SDF-on and SDF-off states was higher than the other tasks.

The shape of the environmental object for our Tasks 2 and 3 is in principle suitable for analysis with the Steering Law. However, Tasks 2 and 3 started when the object passed through the start flag and ended when it passed through the finish flag, and there was 8.91 cm space between the start flag and the wires. Moreover, the wire width and wire length were not varied. This makes applying the steering law not easily feasible for either task. Furthermore, Task 3 involved complex curved shapes, which are challenging to model using the steering law. We recommend investigating the relationship between the steering law and our results in the future.

8 LIMITATIONS

In this study, we employed only three different tasks to evaluate both user performance and experience and future work might thus need to investigate additional task scenarios and applications with different difficulties. Also, the effect of the SDF on maintenance or assembly tasks that typically involve complex and dense environments should be investigated in future studies.

For simplicity, we used a fixed resolution of 1 cm for the SDF. For bigger objects or larger environments, it might thus be necessary to adjust the resolution of the SDF array dynamically.

We also evaluated our system on tasks where the minimum depth manipulation was required. Since manipulating objects in depth has its own challenges and several researchers proposed new interaction methods to manipulate objects in depth, e.g., [56, 24, 18], we did not investigate this aspect. Nonetheless, future research should incorporate the SDF into techniques for object manipulation involving depth changes.

We also compared our method with ray casting since it is the standard and frequently used technique for distant object manipulation in VR applications. Still, ray casting-based manipulation is susceptible to hand tremors and tracking jitters. The other interaction method that is vastly used in VEs is the virtual hand interaction technique. For future research, we propose evaluating our new method with the virtual hand technique (and other VR manipulation techniques).

9 CONCLUSION

In this paper, we proposed using a Signed Distance Field to regulate the speed of a manipulated distant object to improve the accuracy of manipulation, leveraging the speed-accuracy trade-off. We compared the performance of our proposed approach with RAY CASTING manipulation technique. Our results indicate that RAY CASTING WITH SDF reduces the number of hits while not slowing down the participants. Moreover, RAY CASTING WITH SDF increases the usability of the system for complex tasks and reduces mental demand, temporal demand, and frustration. We hope that our proposed method can be used as an assistive system with other manipulation techniques in future applications where user accuracy is essential for task completion success.

REFERENCES

- [1] L. Aguerreche, T. Duval, and A. Lécuyer. Short paper: 3-Hand Manipulation of Virtual Objects. In *JVRC 2009*, pp. 4–p, 2009. [2](#)
- [2] D. Almeida, D. Mendes, and R. Rodrigues. SIT6: Indirect Touch-Based Object Manipulation for DeskVR. *Computers & Graphics*, 117:51–60, 2023. doi: 10.1016/j.cag.2023.10.013 [2](#)
- [3] F. Argelaguet and C. Andujar. Efficient 3D Pointing Selection in Cluttered Virtual Environments. *IEEE Computer Graphics and Applications*, 29(6):34–43, 2009. [2](#)
- [4] F. Argelaguet and C. Andujar. A Survey of 3D Object Selection Techniques for Virtual Environments. *Computers & Graphics*, 37(3):121–136, 2013. [1](#), [2](#), [8](#)
- [5] C. Auteri, M. Guerra, and S. Frees. Increasing Precision for Extended Reach 3D Manipulation. *International Journal of Virtual Reality*, 12(1):66–73, 2013. [1](#)
- [6] G. Ballestin, M. Chessa, and F. Solari. A Registration Framework for the Comparison of Video and Optical See-Through Devices in Interactive Augmented Reality. *IEEE Access*, 9:64828–64843, 2021. [1](#)
- [7] M. Baloup, T. Pietrzak, and G. Casiez. Raycursor: A 3D Pointing Facilitation Technique based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019. [2](#), [8](#)
- [8] A. U. Batmaz, M. D. Barrera Machuca, J. Sun, and W. Stuerzlinger. The Effect of the Vergence-Accommodation Conflict on Virtual Hand Pointing in Immersive Displays. In *Conference on Human Factors in Computing Systems*, CHI ’22, May 2022. doi: 10.1145/3491102.3502067 [1](#)
- [9] A. U. Batmaz, M. D. B. Machuca, D. M. Pham, and W. Stuerzlinger. Do Head-Mounted Display Stereo Deficiencies Affect 3D Pointing Tasks in AR and VR? In *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*, pp. 585–592. IEEE, 2019. [1](#)
- [10] A. U. Batmaz, M. R. Seraji, J. Kneifel, and W. Stuerzlinger. No Jitter Please: Effects of Rotational and Positional Jitter on 3D Mid-Air Interaction. In *Proceedings of the Future Technologies Conference*, pp. 792–808. Springer, 2020. [1](#)
- [11] A. U. Batmaz and W. Stuerzlinger. The Effect of Rotational Jitter on 3D Pointing Tasks. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–6, 2019. [1](#)
- [12] A. U. Batmaz and W. Stuerzlinger. Effect of Fixed and Infinite Ray Length on Distal 3D Pointing in Virtual Reality. Extended Abstract, Apr 2020. doi: 10.1145/3334480.3382796 [2](#)
- [13] D. A. Bowman and L. F. Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 35–ff, 1997. [1](#), [2](#), [8](#)
- [14] H. Brument, A.-H. Olivier, M. Marchal, and F. Argelaguet. Does the Control Law Matter? Characterization and Evaluation of Control Laws for Virtual Steering Navigation. In *ICAT-EGVE 2020-International Conference on Artificial Reality and Telexistence & Eu-graphics Symposium on Virtual Environments*, pp. 1–10, 2020. [8](#)
- [15] F. Calakli and G. Taubin. Ssd: Smooth Signed Distance Surface Reconstruction. In *Computer Graphics Forum*, vol. 30, pp. 1993–2002. Wiley Online Library, 2011. [2](#)
- [16] D. L. Chen, M. Giordano, H. Benko, T. Grossman, and S. Santosa. Gazeraycursor: Facilitating Virtual Reality Target Selection by Blending Gaze and Controller Raycasting. In *Proceedings of the 29th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–11, 2023. [2](#)
- [17] A. O. S. Feiner. The Flexible Pointer: An Interaction Technique for Selection in Augmented and Virtual Reality. In *Proc. UIST*, vol. 3, pp. 81–82, 2003. [2](#)
- [18] S. Frees, G. D. Kessler, and E. Kay. PRISM Interaction for Enhancing Control in Immersive Virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(1):2–es, 2007. [2](#), [8](#)
- [19] S. Freitag, B. Weyers, and T. W. Kuhlen. Automatic Speed Adjustment for Travel Through Immersive Virtual Environments Based on Viewpoint Quality. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 67–70. IEEE, 2016. [8](#)
- [20] N. Gerig, J. Mayo, K. Baur, F. Wittmann, R. Riener, and P. Wolf. Missing Depth Cues in Virtual Reality Limit Performance and Quality of Three Dimensional Reaching Movements. *PLoS one*, 13(1):e0189275, 2018. [1](#)
- [21] P. C. Gloumeau, W. Stuerzlinger, and J. Han. Pinnpivot: Object Manipulation Using Pins in Immersive Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 27(4):2488–2494, 2020. [2](#), [8](#)
- [22] T. Grossman and R. Balakrishnan. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pp. 3–12, 2006. [2](#)
- [23] J. F. Hair Jr, W. C. Black, B. J. Babin, and R. E. Anderson. Multivariate Data Analysis, 2014. [6](#)
- [24] J. D. Hincapié-Ramos, K. Ozacar, P. P. Irani, and Y. Kitamura. GyroWand: IMU-Based Raycasting for Augmented Reality Head-Mounted Displays. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, pp. 89–98, 2015. [8](#)
- [25] A. Hobson. Redirected Hands for Reducing Arm Fatigue During Mid-Air Interactions in Virtual Reality. 2023. [2](#)
- [26] J. Hu, M. K. Yip, G. E. Alonso, S. Gu, X. Tang, and X. Jin. Efficient Real-Time Dynamic Diffuse Global Illumination Using Signed Distance Fields. *The Visual Computer*, 37:2539–2551, 2021. [2](#)
- [27] A. Jacobson, D. Panozzo, et al. libigl: A Simple C++ Geometry Processing Library, 2018. [https://libigl.github.io/](#). [3](#)
- [28] R. Jota, M. A. Nacenta, J. A. Jorge, S. Carpendale, and S. Greenberg. A Comparison of Ray Pointing Techniques for Very Large Displays. In *Proceedings of graphics interface 2010*, pp. 269–276. 2010. [2](#)
- [29] J. Kangas, S. K. Kumar, H. Mehtonen, J. Järnstedt, and R. Raisamo. Trade-off Between Task Accuracy, Task Completion Time and Naturalness for Direct Object Manipulation in Virtual Reality. *Multimodal Technologies and Interaction*, 6(1), 2022. doi: 10.3390/mti6010006 [2](#)
- [30] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and Accurate 3D Selection by Progressive Refinement. In *2011 IEEE symposium on 3D user interfaces (3DUI)*, pp. 67–74. IEEE, 2011. [2](#)
- [31] S. Lavallée and R. Szelski. Recovering the Position and Orientation of Free-Form Objects from Image Contours Using 3D Distance Maps. *IEEE Transactions on pattern analysis and machine intelligence*, 17(4):378–390, 1995. [2](#)
- [32] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2017. [1](#), [2](#)
- [33] J.-I. Lee, P. Asente, and W. Stuerzlinger. Designing Viewpoint Transition Techniques in Multiscale Virtual Environments. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 680–690. IEEE, 2023. [8](#)
- [34] S. Lee, J. Seo, G. J. Kim, and C.-M. Park. Evaluation of Pointing Techniques for Ray Casting Selection in Virtual Environments. In *Third international conference on virtual reality and its application in industry*, vol. 4756, pp. 38–44. SPIE, 2003. [2](#)
- [35] P. Liu, Y. Zhang, H. Wang, M. K. Yip, E. S. Liu, and X. Jin. Real-time Collision Detection Between General SDFs. *Computer Aided Geometric Design*, p. 102305, 2024. [3](#)
- [36] X. Liu, L. Wang, S. Luan, X. Shi, and X. Liu. Distant Object Manipulation with Adaptive Gains in Virtual Reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 739–747. IEEE, 2022. [2](#)
- [37] Y. Lu, C. Yu, and Y. Shi. Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality. In *2020 IEEE Conference on virtual reality and 3D user interfaces (VR)*, pp. 35–43. IEEE, 2020. [2](#), [8](#)
- [38] R. Luo, M. Zolotas, D. Moore, and T. Padir. User-Customizable Shared Control for Fine Teleoperation via Virtual Reality. *arXiv preprint arXiv:2403.13177*, 2024. [5](#)
- [39] I. S. MacKenzie and P. Isokoski. Fitts’ Throughput and the Speed-Accuracy Tradeoff. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1633–1636, 2008. [1](#), [2](#)
- [40] P. Mallory and D. George. *SPSS for Windows Step by Step: A Simple Guide and Reference*. Pearson, 2003. [6](#)
- [41] D. Mendes, F. M. Caputo, A. Giachetti, A. Ferreira, and J. Jorge. A Survey on 3D Virtual Object Manipulation: from the Desktop to Im-

- mersive Virtual Environments. In *Computer graphics forum*, vol. 38, pp. 21–45. Wiley Online Library, 2019. 1, 2
- [42] D. Mendes, D. Medeiros, M. Sousa, E. Cordeiro, A. Ferreira, and J. A. Jorge. Design and evaluation of a novel out-of-reach selection technique for VR using iterative refinement. *Computers & Graphics*, 67:95–102, 2017. 1
- [43] D. Mendes, F. Relvas, A. Ferreira, and J. Jorge. The Benefits of DOF Separation in Mid-Air 3D Object Manipulation. In *Proceedings of the 22nd ACM conference on virtual reality software and technology*, pp. 261–268, 2016. 2
- [44] D. Mendes, M. Sousa, R. Lorena, A. Ferreira, and J. Jorge. Using Custom Transformation Axes for Mid-Air Manipulation of 3D Virtual Objects. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, pp. 1–8, 2017. 2, 8
- [45] M. R. Mine. Virtual Environment Interaction Techniques. *UNC Chapel Hill CS Dept*, 1995. 1, 2
- [46] A. G. Moore, J. G. Hatch, S. Kuehl, and R. P. McMahan. VOTE: A Ray-Casting Study of Vote-Oriented Technique Enhancements. *International Journal of Human-Computer Studies*, 120:36–48, 2018. 2
- [47] A. K. Ng, L. K. Chan, and H. Y. Lau. Depth Perception in Virtual Environment: The Effects of Immersive System and Freedom of Movement. In *Virtual, Augmented and Mixed Reality: 8th International Conference, VAMR 2016, Held as Part of HCI International 2016, Toronto, Canada, July 17–22, 2016. Proceedings* 8, pp. 173–183. Springer, 2016. 1
- [48] T. T. H. Nguyen, T. Duval, and C. Pontonnier. A New Direct Manipulation Technique for Immersive 3D Virtual Environments. In *ICAT-EGVE 2014: the 24th International Conference on Artificial Reality and Telexistence and the 19th Eurographics Symposium on Virtual Environments*, p. 8, 2014. 2
- [49] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart. Signed Distance Fields: A Natural Representation for Both Mapping and Planning. In *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. University of Michigan, 2016. 3
- [50] N. Osawa. Two-Handed and One-Handed Techniques for Precise and Efficient Manipulation in Immersive Virtual Environments. In *International Symposium on Visual Computing*, pp. 987–997. Springer, 2008. 8
- [51] B. A. Payne and A. W. Toga. Distance Field Manipulation of Surface Models. *IEEE Computer graphics and applications*, 12(01):65–71, 1992. 2
- [52] K. Pietroszek. 3D Pointing with Everyday Devices: Speed, Occlusion, Fatigue. 2015. 2
- [53] K. Pietroszek. *Raycasting in Virtual Reality*, pp. 1–3. Springer International Publishing, Cham, 2018. doi: 10.1007/978-3-319-08234-9_180-1 2
- [54] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. 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*, pp. 79–80, 1996. 2, 8
- [55] T. Reiner, G. Mückl, and C. Dachsbaecher. Interactive Modeling of Implicit Surfaces Using a Direct Visualization Approach with Signed Distance Functions. *Computers & Graphics*, 35(3):596–603, 2011. 2
- [56] H. Ro, J.-H. Byun, Y. J. Park, N. K. Lee, and T.-D. Han. AR Pointer: Advanced Ray-Casting Interface Using Laser Pointer 7 for Object Manipulation in 3D Augmented Reality Environment. *Applied Sciences*, 9(15):3078, 2019. 8
- [57] D. Seyb, A. Jacobson, D. Nowrouzezahrai, and W. Jarosz. Non-Linear Sphere Tracing for Rendering Deformed Signed Distance Fields. *ACM Transactions on Graphics*, 38(6), 2019. 3
- [58] A. Singh, K. Joshi, M. Shuaib, S. Bharany, S. Alam, and S. Ahmad. Navigation and Speed Regulation Aimed at Travel through Immersive Virtual Environments: A Review. In *2022 IEEE International Conference on Current Development in Engineering and Technology (CCET)*, pp. 1–6. IEEE, 2022. 8
- [59] R. W. Soukoreff and I. S. MacKenzie. An Informatic Rationale for the Speed-Accuracy Trade-Off. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2890–2896. IEEE, 2009. 2
- [60] R. Stoakley, M. J. Conway, and R. Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 265–272, 1995. 2
- [61] Y. W. Tan, N. Chua, C. Koh, and A. Bhojan. RTSDF: Real-Time Signed Distance Fields for Soft Shadow Approximation in Games. *arXiv preprint arXiv:2210.06160*, 2022. 2
- [62] C. Vienne, S. Masfrand, C. Bourdin, and J.-L. Vercher. Depth Perception in Virtual Reality Systems: Effect of Screen Distance, Environment Richness and Display Factors. *IEEE Access*, 8:29099–29110, 2020. 1
- [63] C. Wilkes and D. A. Bowman. Advantages of Velocity-Based Scaling for Distant 3D Manipulation. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pp. 23–29, 2008. 2, 8
- [64] H. Wu, X. Sun, H. Tu, and X. Zhang. Clockray: A Wrist-Rotation Based Technique for Occluded-Target Selection in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2
- [65] H. P. Wyss, R. Blach, and M. Bues. iSith-Intersection-Based Spatial Interaction for Two Hands. In *3D User Interfaces (3DUI'06)*, pp. 59–61. IEEE, 2006. 2
- [66] X. Zhang, W. He, M. Billinghurst, L. Yang, S. Feng, and D. Liu. Design and Evaluation of Bare-Hand Interaction for Precise Manipulation of Distant Objects in AR. *International Journal of Human-Computer Interaction*, pp. 1–15, 2022. 2