Fully-Occluded Target Selection in Virtual Reality

Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, Jorge Goncalves

Abstract—The presence of fully-occluded targets is common within virtual environments, ranging from a virtual object located behind a wall to a datapoint of interest hidden in a complex visualization. However, efficient input techniques for locating and selecting these targets are mostly underexplored in virtual reality (VR) systems. In this paper, we developed an initial set of seven techniques techniques for fully-occluded target selection in VR. We then evaluated their performance in a user study and derived a set of design implications for simple and more complex tasks from our results. Based on these insights, we refined the most promising techniques and conducted a second, more comprehensive user study. Our results show how factors, such as occlusion layers, target depths, object densities, and the estimation of target locations, can affect technique performance. Our findings from both studies and distilled recommendations can inform the design of future VR systems that offer selections for fully-occluded targets.

Index Terms—Pointing selection, object selection, visualization, occlusion, virtual reality, hidden target, head-mounted displays.

1 Introduction

Virtual reality (VR) enables users to achieve what may not be possible in the physical world. Though many user interfaces have been developed for simulating or adapting real-world features (such as providing realistic tactile feedback [5]), it has long been argued that the real power of VR lies in creating a "better" reality by utilizing "magical" techniques that while being unrealistic, provide a better user experience [1, 46, 63, 65, 67]. One primary advantage of such interaction techniques is to overcome human limitations in terms of cognitive, perceptual, physical, and motor capabilities [46]. For example, existing techniques enable the user to interact with distant objects [78] and teleport around virtual environments [37], which are impossible in the physical world. This research focuses on one such interaction—selecting fully-occluded targets in VR.

The challenge of interacting with fully-occluded targets is prevalent within virtual environments. Structural elements, like walls, can easily hide and prevent users from accessing the objects behind them [23, 47, 79] (see Figure 1). In another example, high-dimensional data visualizations are also likely to obscure a datapoint of interest from being acquired by analysts [6, 20, 54, 76]. Further, when building 3D models in virtual environments, it might be cumbersome to select and thus manipulate hidden components, such as an engine hidden inside a virtual model of a motor vehicle [4].

However, existing selection techniques in VR are limited in their effectiveness for selecting fully-occluded targets. Based on the available literature on the topic, we argue that the main challenges are **(1)** the deficiency of the formulation of the problem in VR and general strategies to solve it; **(2)** the lack of effort in combining 3D occlusion management techniques to facilitate the discovery phase of the selection process [3]; and **(3)** the absence of a thorough evaluation and comparison of techniques that manipulate the key factors related to fully-occluded target selection. We aim to fill these gaps in this paper.

We first formulate the fully-occluded target selection problem and frame an approach to address it. We then derive a design space, which inspired seven potential techniques for selecting fully-occluded targets in VR. We present a user study that compares these techniques based on both simple and complex tasks. Based on the study results, we refined the more promising techniques and introduced a second, more in-depth study aimed at assessing technique performance under different environmental factors including occlusion layers, target depths, object densities, and the estimation of the target locations. Following, we discuss the findings from both studies and suggest recommendations to inform the design of future VR systems that offer selections for fully-occluded targets.

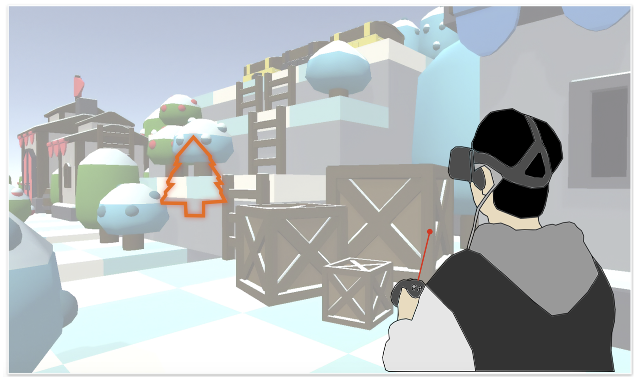

Fig. 1. Example Scenario: A user is constructing an environment in VR and intends to select and manipulate a hidden tree (outlined in orange) that is fully-occluded from the view of the user.

2 Problem Formulation and General Strategy

Considering previous work regarding the occlusion problem in 3D environments [31], we formulate the problem space and propose a general problem-solving strategy for the selection of fully-occluded targets in head-mounted display (HMD)-based VR systems.

Within a 3D virtual environment, there are *selectable*, and *unselectable* objects. Users can pick up or interact with the selectable objects, but not the objects that are unselectable since they serve other purposes within the virtual environment, such as decoration to enhance the realism of the scene. Among the selectable objects, there is commonly one primary *target* that the user intends to interact with, while all the selectable objects act as *distractors*. The target can switch when the user's intention changes. A target is defined to be fully-occluded from a viewpoint if it can not be seen from any viewing direction of the user. Different objects within a virtual environment can become fully-occluded at some point during the interaction.

To select a fully-occluded target, the user needs first to form an intention. With that intention, although the user cannot directly see the target at this stage, they typically have an awareness of the areas where it might occur—we call them *occurrence areas*. The estimated size of the occurrence area depends on the user's confidence. If the user has no idea of where the target might locate, the occurrence area

- *The authors are with School of Computing and Information Systems, The University of Melbourne. E-mails: {difengy, qiushiz2}@student.unimelb.edu.au; {joshua.newn, tilman.dingler, eduardo.velloso, jorge.goncalves}@unimelb.edu.au.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

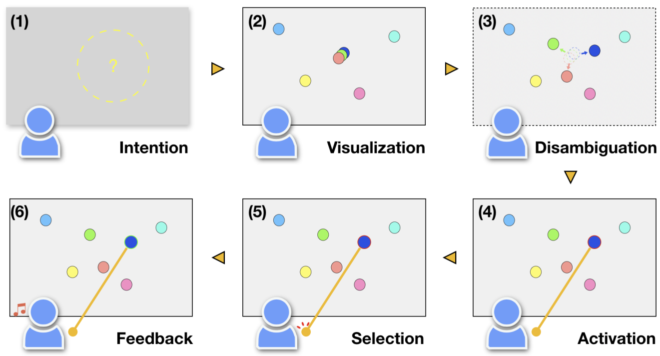

Fig. 2. To select fully-occluded targets, a user 1) forms an intention. The user is then provided with 2) visualization of the objects, 3) an optional disambiguation technique to spread the clustered objects, 4) an indicator of the current pointing object (e.g. highlighting), 5) a selection trigger to confirm the selection, and 6) feedback (e.g., visual, sound, haptic) after selection.

is the whole visualization. After that, the user can use a supporting technique to "locate" the fully-occluded target and then pinpoint the target to perform the selection.

In line with the problem space discussed above, we propose a general strategy to tackle the fully-occluded target selection problem. Once a user has a selection intention, a visualization of the target needs to be displayed to make it "visible" to the user from their viewport. However, as the technique will not know which object the user is aiming at, a group of potential objects, possibly within the user's estimated occurrence areas, will be presented. Next, the technique helps the user to disambiguate the list of selectable objects and provides activation feedback when an object is being pointed at. Finally, the technique should allow the user to select an object by pressing a trigger and receive confirmation feedback. The general strategy described above for selecting fully-occluded targets is summarized in Figure 2.

3 Related Work

In this section, we first introduce previous work regarding occluded target selection in VR. We then present the techniques related to visualization and selection, which are the two main steps in our general problem-solving strategy.

3.1 Occluded Target Selection in VR

Previous research in VR has explored object selection under cluttered and occlusion conditions, mostly for partially-occluded targets. One successful technique is Depth Ray [38], which attaches a movable marker onto the selection ray. The object intersected by a ray and closest to the marker can be selected. In another study, Depth Ray was shown to be effective for selecting a target that was completely occluded by making the objects adjacent to the ray semi-transparent [72]. However, the target was first visible to the user for 2 seconds before it got fully-occluded by distractors, which is unlikely to be the case in real applications. Furthermore, only one layer of occlusion was introduced, and the target shape was different from the distractors, which made the task much simpler. In real scenarios, multiple layers of occlusion can be presented, and the target can also be similar to the distractors [54]. Future work has refined [11], and applied similar techniques for collaborative work [2] and object manipulation [57].

Some techniques try to separate objects in a cluster by translating them into new positions. For example, Flower Ray uses a two-step approach: the user first point at an object cluster through a virtual ray, and then press to trigger to separate them into a marking menu [38, 44]. An updated version of Flower Ray uses a fixed-size cone to replace the ray to avoid missing small targets [26]. Both techniques have not been tested under dense conditions, where objects could still be partially- or fully-occluded even if they are translated to new positions.

Progressive refinement techniques that take the advantages of rearranging the objects in a more organized way, commonly into a new view, can also be suitable for selecting fully-occluded targets. SQUAD [7, 42] allows users to cast a sphere onto multiple objects and interactively narrow down their selections using a quad-menu. Another technique called Expand [9, 18, 19] (not in VR HMD) enables users to zoom into the target area and reorganize the objects onto a grid for a second phase selection. Expand was shown to perform faster than SQUAD in dense environments. Later works extend such techniques by using a mobile touchscreen as input [25] and arranging objects in different layouts (circular layout rather than a grid) [49, 56]. However, none of them have been formulated under the context of fully-occluded target selection, nor have they been thoroughly compared to other techniques presented in this section. Nevertheless, we drew inspiration from these techniques when developing our techniques for fully-occluded target selection in VR.

There are some other techniques that are promising for selecting fully-occluded targets: flexible pointer [51] uses a curved ray which could bypass the distractors, iSith [77] determines the target by using the interaction point of two rays, VirtualGrasp [78] retrieves an object by simulating the gesture as if grasping the target object, X-Ray Vision [40] reveals hidden content by looking at a "scaffolding pattern", and Outline Pursuits [64] selects an occluded target by matching its outline with smooth pursuit eye movement [73]. While these techniques provide interesting concepts, substantial tweaks would be needed for them to be suitable for general fully-occluded target selection scenarios. For example, VirtualGrasp [78] can not deal with objects with an identical shape.

We summarise the following three gaps in the literature:

- The fully-occluded target selection problem has not been established in VR. Previous work normally assumed that the target location was known or only partially hid the target. In addition, important factors, such as layers of occlusions, were not identified.
- Limited work has tried to combine occlusion visualizations to support the discovery phase of the targets, as they mainly focus on the selection phase. However, as fully-occluded targets can cause some uncertainties with their locations, visualizations that help with the search phase are essential.
- A thorough evaluation and comparison of different types of techniques that could be potentially used for fully-occluded target selection are missing.

3.2 3D Occlusion Visualization

Elmqvist and Tsigas reviewed fifty 3D occlusion management techniques for visualizations [31] and extracted five design patterns from these techniques. Next, we highlight important work in the three patterns that are more relevant to our research.

Multiple Viewports. The multiple viewports pattern is characterized by embedding alternate (often separate) viewports/windows to the main view. Examples include World In Miniature (WIM) [67, 70], which generates a small, handheld copy of the entire world, and Worldlets [33], which inserts multi-perspective viewpoints of an environment into the main view [13, 58, 76]. Recent work presents 3DMini-map [79], which helps to convey distance and direction information of off-screen and occluded targets. However, selecting objects directly on these visualizations is still underexplored.

Virtual X-Ray. The virtual X-ray pattern makes objects visible by turning occlusion surfaces invisible or semi-transparent. Making front objects transparent can benefit the discovery of objects that hide behind [27, 30, 41, 68, 82]. However, it is known to suffer from the "Superman's X-ray vision" problem [48]—when there are too many occlusion layers, users are not able to make sense of the depth relationships of objects. Others have explored a cutaway view [16, 23, 28, 34], which eliminates or cuts holes over unwanted distractors.

Volumetric Probes. Volumetric probes normally use a probe to transform objects by removing or separating them. The above-mentioned disambiguation techniques, which reorganized potential targets on a new view [74], could be counted as one substream. Other techniques have attempted to scale [22], translate [8,15,29,32,55], and distort [17,24] objects in the scene in order to reveal the hidden objects. The transformation of the object needs to be carefully controlled so that the object is not occluded by new distractors, especially in a dense environment [32].

3.3 Selection Techniques in VR

There are two main categories of selection techniques in VR: virtual hand and virtual pointing [3]. Since a plethora of techniques have been proposed under those two categories, we direct interested readers to surveys on the topic [3,46], and more recent works [11,12,71,80]. RayCasting [11,50] it is one of the most common techniques for 3D object selection in virtual environments. In RayCasting, a visible ray emanates from the tracked hand position to the direction of where the hand is pointing at, and the first object that is intersected by the ray can be selected [46]. Despite its usefulness, the performance of RayCasting deteriorates when selecting distant or small objects. Researchers have been actively seeking solutions to enhance its performance, especially in dense environments [36,72]. Recent work has compared different visual feedforwards for RayCasting and suggests that highlighting the nearest target was the most efficient way in terms of selection performance [11]. Another approach is to try to minimize input noise with the use of algorithms and computational models [11,80]. We utilized some of the techniques mentioned above to strengthen our fully-occluded target selection techniques. We illustrate this aspect in more detail in the description of the developed techniques.

4 Design Space

Our general strategy for selecting fully-occluded target suggests that the problem can be solved in five steps (visualization, disambiguation, activation, selection, and feedback). Here, we focus on the three main steps, which are visualization, disambiguation, and selection. We maintain the other two the same across all the techniques. The activation indication was provided by outlining the target, and the confirmation feedback was given by sound. Regarding the three focused steps, we have identified the following six primary considerations for designing fully-occluded target selection techniques.

Visualization Patterns: which type(s) of the visualization pattern, among the ones that are identified by Elmqvist et al. [31] (typically multiple viewports, virtual X-ray, and volumetric probe) is/are utilized to visualize the target?

Visualization Size: what is the size of the visualization area? Are we applying the visualization to only limited areas, or more extensive areas (even the whole environment)? For instance, to visualize the objects, we can make a small area transparent, however, we can also tweak the whole scene to do so.

Visualization Versatility: will users be able to specify which area(s) they want to apply the visualization? How precise can it be (in an arbitrary shape or a constrained region)? In real use cases, the users will have different estimations of where the target might occur, thus it is important to define their belief/guess accurately.

Disambiguation Invariances: when applying the disambiguation technique, which property (or properties) of the original objects will be maintained? These properties may include object position, size, relation, and appearance. For example, if we are asked to select a datapoint among other datapoints that have the same appearance, re-arranging all of them into new positions might not be ideal.

Selection Techniques: what type of selection techniques will be applied? Are we embedding selection enhancement techniques or filter out the noisy input? These decisions are likely to be highly related to selection performance. In the initial exploration, we mainly focus on the selection techniques that are based on pointing (Raycasting) and virtual hands without adding selection enhancements.

Input Modality: which input modality (modalities) are used for selecting the fully-occluded target? While many types of input modality exist (voice, gaze, gesture, etc.), we focus on controller input. A survey of currently available controllers on the market showed that most controllers were equipped with at least a touchpad or a joystick (2 degrees-of-freedom input, 2DOF), a trigger (1DOF input), and buttons (only on/off). The controller itself can be 3DOF (only rotation can be detected) or 6DOF (both rotation and translation can be recognized). Different techniques might need to employ different inputs. In our research, we used a joystick, a trigger, and a button of an Oculus Touch controller throughout the studies. The design space can be expanded in the future when investigating other input modalities to achieve the functionalities of each technique (e.g. hand-tracking).

5 Potential Techniques

Based on the design space, we developed the following nine potential techniques with several iterations and pilot tests. These techniques are summarized and visualized in Figure 3. The following technique descriptions adhere to the design space. For an explicit mapping between the design space and the techniques, please refer to our supplementary materials.

Alpha Cursor: this technique is inspired by previous work that attaches a movable cursor onto the selection ray [11,38]. With *AlphaCursor*, users control the cursor to come closer or go deeper into the environment at a constant speed by pushing the joystick forward or backward (see Figure 3b). In contrast to previous work, if the distance between the cursor and the user is larger than the distance between an object to the user, the object becomes fully transparent. The transparency manipulation is applied to the whole environment, and all objects maintain their original position and size during the disambiguation phase. RayCasting, which uses the trigger for selection confirmation, then selects the desired object.

Flower Cone: in *FlowerCone* (see Figure 3c), users select objects in two phases. First, the user controls a cone to match the estimated area of where the target might occur. The size of the cone can be adjusted by tilting forward/backward the joystick. When pressing the trigger, the user enters the second selection phase, in which all objects within that cone are presented on a grid. The user can select the target directly on the grid with RayCasting, or, if the target is not on the grid, the user can press the button to go back and resize the cone again. This technique combines visualization and disambiguation by using the grid layout. The visualization size can be controlled through the size of the flat circular base of the cone. However, the grid layout changes the original location and size of the object.

Gravity Zone: as shown in Figure 3d, *GravityZone* translates all objects in the scene to come closer or further away in a constant speed by tilting the joystick forward or backward. If the distance between an object and the user is smaller than a threshold, the object will be fully transparent. It is similar to *AlphaCursor* in that both of them make the objects transparent based on their relative depth. However, in contrast to *AlphaCursor*, *GravityZone* manipulates all the objects in the scene rather than the cursor. The location and size of the objects are changed during the translation, but their relative position is not altered. Raycasting is used to make the selection.

Grid Wall: inspired by Expand [19], in this technique, when the user presses the controller button, all objects are arranged on a grid (see Figure 3e) with a constant scale factor. We did not use the zoom-in feature from Expand as it can make participants dizzy in VR. *GridWall* completely reorganizes all objects in the scene to a new location with a different size. The user can select the target on the grid with RayCasting. The original location information of the object is temporarily lost with the grid layout.

Lasso Grid: with *LassoGrid*, users draw a trace in any shape by long-pressing the trigger (see Figure 3f). All objects within the trace, are presented on a grid layout when releasing the trigger for the second stage of selection. If the trace is not closed, the program completes it automatically. RayCasting is used to select the target on the grid. Pressing the button allows the user to go back and draw the trace again.

Magic Ball: inspired by previous work [79] (which only explored visualization rather then selection), *MagicBall* removes unselectable distractors and creates a 3D mini-map of all the selectable objects inside a transparent sphere (see Figure 3g). The objects' size and the distance between each other are both scaled-down, but the relative size and location information are both maintained. The user can select directly on the semi-transparent object proxies by moving the tip of the virtual

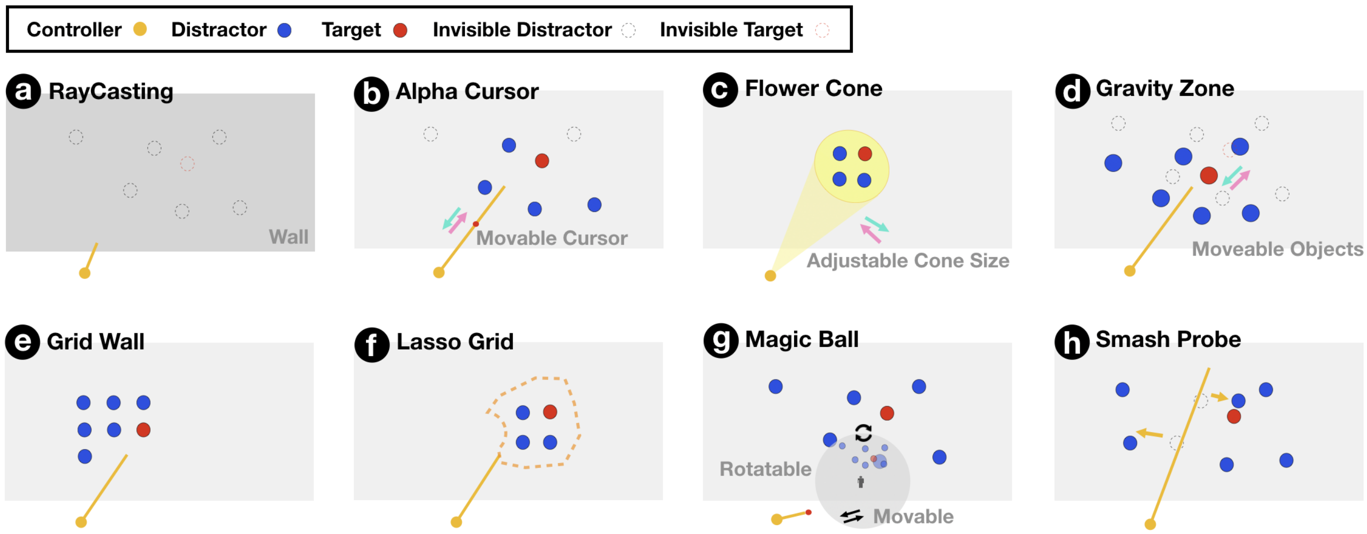

Fig. 3. The RayCasting technique (a) and techniques devised for fully-occluded target selection, including Alpha Cursor (b), Flower Cone (c), Gravity Zone (d), Grid Wall (e), Lasso Grid (f), Magic Ball (g), and Smash Probe (h).

stick onto the proxy and pressing the trigger. The user can also rotate and translate the mini-map by tilting the joystick.

Smash Probe: if more than one object intersects with the ray, *Smash-Probe* spreads these objects to a random direction within a fixed range (see Figure 3h). It only alters a small area per spread; however, multiple spreads can disarray the whole environment. The objects are translated back to their original position after a pre-defined time. RayCasting is used for selecting objects, and the user can disable or re-enable the spread function by pressing the button.

Depth Ray (discarded): previous work [38, 72] has described Depth Ray, which attaches a depth marker onto the selection ray. The objects that are close to the ray are rendered as semi-transparent so that the occluded targets could become visible. The one that is closest to the marker can be selected. However, during the pilot testing, we found that users were not able to distinguish the target and the distractors when multiple occlusion layers showed up using this technique, even with semi-transparent and border highlighting. Thus, we discarded this technique from the study.

Fly-Through (discarded): the technique allows users to fly through any objects and navigate freely across the virtual environment. However, following our pilot testing, it became clear that this technique was not efficient for this purpose and could cause motion sickness, and therefore, we also discarded this technique from the study.

All our techniques introduce a superimposed selection mode, which removes the unselectable objects in the scene for the simplicity of selectable target acquisition. While conducting user studies to optimize each technique was not feasible, and outside the scope of this research, we tuned all of their variables to the best of our ability during informal testing. The values of the variables are made available in our supplementary materials for replication purposes.

6 Evaluation Framework

6.1 Variables of Interest

We identified a set of factors that we hypothesized could impact the techniques' performance for fully-occluded target selection. As suggested by Fitts's law [35, 66], *target size* and *movement amplitude* are likely to have a significant effect on selection performance. However, rather than replicating the findings from the extensive previous work on the topic, we consider variables that are related to occlusion properties. These variables are:

- *Occurrence Area*. As discussed before, a user normally has an awareness of where a fully-occluded target might be located. The more uncertain the user is, the larger the occurrence area might be. Different sizes of the occurrence area is likely to influence the target searching time.
- *Occlusion Layer*. It specifies the number of selectable objects that can fully overlap the target from a user's point of view. It is more challenging for the user to find the target when more occlusion layers are present.
- *Environmental Density*. It is the number of selectable objects in the whole virtual space. Although some of them might not hinder the selection performance directly, it can cause distraction and are quite likely to appear in real application scenarios (unwanted objects are spread across the whole environment).
- *Target Depth*. It is the distance between the target and the user. A higher target depth value can make the target appear smaller to the user and raise more challenges for selection.
- *Density Space*. Density space [21, 38] offers more precise control of the object density within the target area. Similar to previous research, we place six distractors around the target (front, behind, up, down, left, and right). Density space is the distance between the six distractors surrounding the target.

6.2 Experimental Setup

To frame the experimental task for this research, we first consulted the past literature regarding target selection in 3D space. Existing tasks with perceivable patterns (e.g., [11, 66, 69, 81]), which users were required to select a set of fixed targets in a constant sequence, are not applicable in our case. This is because we wanted to vary the occurrence areas, which requires some randomness in the allocation of the target. Meanwhile, tasks based on interaction scenarios with the presence of some degree of unexpectedness, such as a game [19], might pose challenges to the control of variables. We decide to use more controlled tasks, which would still allow the randomization of target locations (such as [7, 38, 45, 53, 72]). However, as there is little work regarding fully-occluded target selection, we had to develop a new and reusable experimental task. Based on previous research, we designed the task as follows.

In the task, the user aimed to select a fully-occluded target among a set of distractors in a virtual environment. The target and the distractors had different colors, and the colors were generated from a pre-prepared list (we used seven colors in our case which were chosen to be easily distinguishable, see Figure 4). The task was divided into two phases:

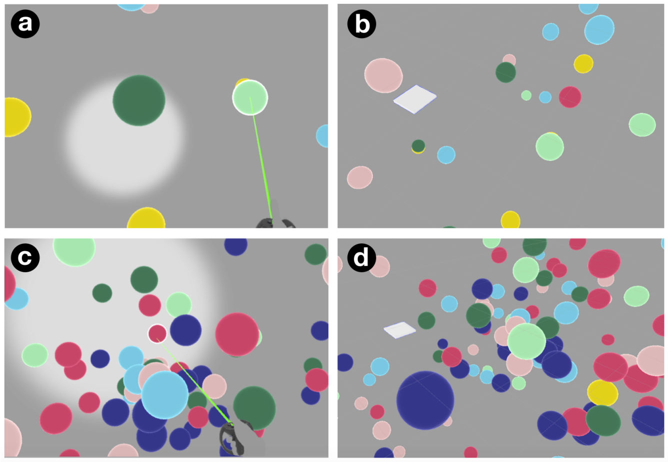

Fig. 4. Demonstrations of the experimental testbed including sparse environment first-person view (a) and third-person view (the square indicates where the user should stand) (b) and dense environment first-person view (c) and third-person view (d).

preparation and formal trial. During the preparation stage, the user started by pointing at a fixed *home object*, which had an identical color as the goal target. The home object ensured that the ray started from the same direction in the formal trial, and the user could press the selection trigger to proceed to the formal trial. During the formal trial, the objects, including both a target and distractors, and an indicator of where the target might be located (the occurrence area, marked as white in Figure 4a, c) were generated. The occurrence area could show up in any direction related to the home object, but the distance between them was always the same in our setting. The user then needed to use the corresponding input technique to select the target. We envision that, by modifying the variables of interest mentioned above, this task can capture a broad range of interaction scenarios in real cases.

7 Study 1 – Initial Exploration

We conducted an initial exploration and evaluation of the seven potential techniques for selecting fully-occluded targets in virtual environments. We aimed to extract design features that perform well in different interaction scenarios and determine potential aspects of our techniques that might need refinement.

7.1 Participants, Apparatus, and Materials

We recruited 21 participants (13F/8M), aged between 19-39 (M = 24.5 $\pm$ 4.3) with a diverse set of educational backgrounds (economics, arts, law, engineering, etc.) from a local university campus. All participants had normal or corrected-to-normal vision and rated their familiarity with VR as moderate (average 3.0 $\pm$ 1.6 out of a 7-point scale). Participants wore an Oculus Rift CV headset and interacted with our application through an Oculus Touch wireless controller.

7.2 Design and Procedure

The study employed a within-subjects design where we compared the performance of the seven developed techniques: (*AlphaCursor*, *FlowerCone*, *GravityZone*, *GridWall*, *LassoGrid*, *MagicBall*, and *SmashProbe*). The techniques were tested on two levels of task complexity (low and high). The higher complexity task had much larger occurrence areas, more occlusion layers before the target, higher environmental density, higher target depths, and larger density space than the lower complexity one. We ensured that there were considerable differences between the two levels of complexities—the detailed parameters are provided in the supplementary material. The order of the techniques was counterbalanced using a Latin Square approach, and the order of complexities was randomized. Following recommendations from previous work on target selection performance [53], we used two subsequent tasks (***search*** and ***repeat***). The search task required users to search for one target in a new scene and then select it, while the repeat task asked users to select the same target in the exact same scene.

We collected both performance data and subjective feedback from participants. Both selection time (the elapsed time between when the objects appear and when the selection is made) and error rate (the percentage of error trials for each condition) were recorded. We also measured the easiness of the techniques with the Single Easement Questionnaire [60] and the intrusiveness caused by them [52] on a 7-point scale. In addition, we asked participants to provide their preference ranking after finishing each technique and optionally also provide free-form feedback. We monitored the experiment from a computer, which showed the user's current view in VR, to observe the use of the techniques.

The whole procedure lasted around 40 minutes for each participant. At the beginning of the study, participants were briefed about the purpose of the research and signed a consent form. They also completed a pre-experiment demographic questionnaire. After that, they were introduced to the VR device and the experimental task, where we required them to finish as fast and as accurately as possible. They then wore the VR headset and familiarised themselves with the virtual environment. Next, they proceeded to the formal experiment within a fixed physical area. The experiment was divided into seven parts (corresponding to the evaluation of seven techniques). In each part, there were three phases: practice, perform formal trials, and answer questions. In the practice phase, participants were taught about how to use the technique, and they could practice it as long as they wanted until they got familiar with it. They then completed a series of formal trials. Finally, they were asked to complete the questionnaires mentioned above. Participants were allowed to rest between each condition. They were compensated with a $10 voucher at the end of the study.

7.3 Performance Results

In total, we collected 4704 data points (21 participants $\times$ 7 techniques $\times$ 2 complexities $\times$ 2 tasks $\times$ 8 repetitions) from the experiment. To analyze selection time, we discarded trials in which participants made a wrong selection (374 error trials, 8.0%), and removed outliers, in which the selection time was above three standard deviations from the mean ($mean + 3std.$) in each condition (92 trials, 2.0%). Such outliers are typically removed as they are likely to not represent the typical selection performance (e.g., small distraction during the experiment), and can skew results in a particular condition [72, 80]. The data regarding selection time were shown to be normally distributed (evidence from Kolmogorov-Smirnov tests and visual inspections), while the error rate data were not normally distributed and underwent pre-processing through Aligned Rank Transform (ART) [11, 75]. Next, we performed a repeated-measures ANOVA (RM-ANOVA) and Bonferroni-adjusted pairwise comparisons in each experiment scenario to analyze the selection time and error rate in each experimental condition[1]. The degrees of freedom produced by RM-ANOVA regarding selection time was adjusted using Greenhouse-Geisser correction. Both results are summarized in Figure 5.

7.3.1 Search Task - Low Complexity

Technique was shown to exhibit a significant main effect on selection time, with a large effect size ($F_{2.893,57.869} = 22.516, p < .001, \eta_p^2 = 0.530$) in low-complexity search task. *GravityZone* was the fastest, being significantly faster than most techniques ($p = 0.036$ for *AlphaCursor* and $p < .001$ for others) except *GridWall* ($p = .219$).

There was a statistically significant difference between Techniques regarding error rates ($F_{6,120} = 3.710, p = .002$). Post-hoc analysis indicated that *FlowerCone* had a significantly higher error rate than *AlphaCursor* ($p = .003$) and *GravityZone* ($p = .002$).

7.3.2 Search Task - High Complexity

Technique had a significant main effect on selection time, with a large effect size ($F_{3.221,64.425} = 13.276, p < .001, \eta_p^2 = 0.399$). *GridWall*

[1]For readability, we here report statistics in APA style (6th Edition). For the exact p-value when $p < .001$, please refer to the supplementary material.

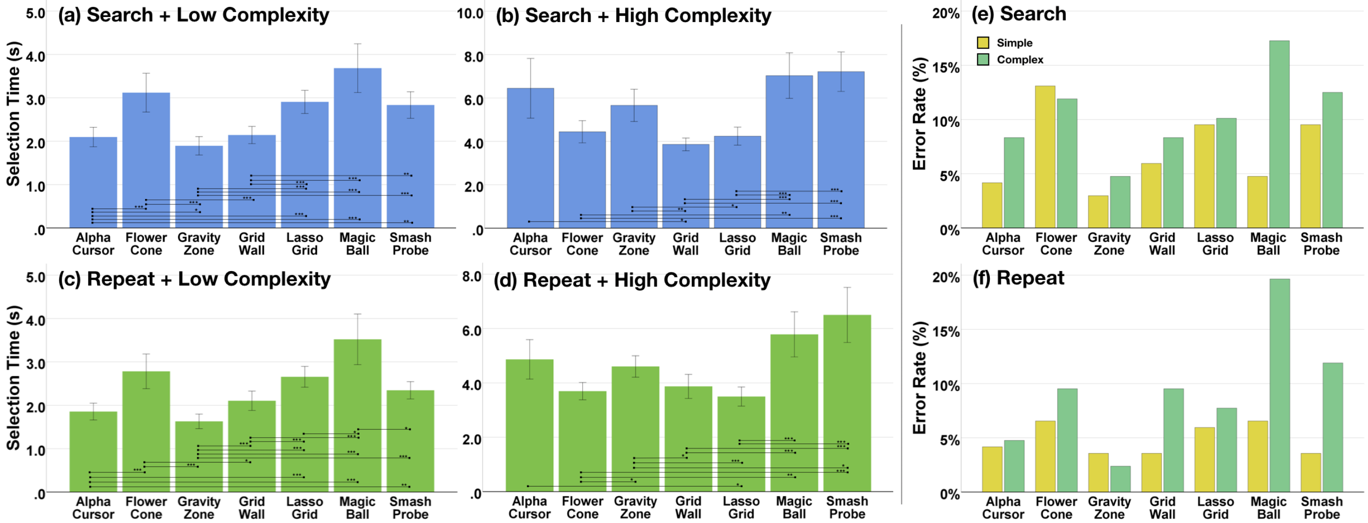

Fig. 5. Plots of selection time for the seven potential techniques regarding the search task with low complexity (a) and high complexity (b) and the repeat task with low complexity (c) and high complexity (d). Error bars indicate the 95% confidence interval. Statistical significant effects are marked (* = p < .05, ** = p < .01, and *** = p < .001). Plots of error rate for the techniques regarding the search task (e) and the repeat task (f).

was the fastest technique for the complex task. It was significantly faster than *AlphaCursor* ($p = .017$), *GravityZone* ($p = .002$), *MagicBall* ($p < .001$), and *SmashProbe* ($p < .001$). There was no statistically significant difference when compared to *FlowerCone* ($p = .362$) and *LassoGrid* ($p = 1.000$).

There was a statistically significant difference between TECHNIQUES regarding error rates ($F_{6,120} = 3.204, p = .006$). *GravityWall* was shown to have significant lower error rate than *MagicBall* ($p = .001$).

7.3.3 Repeat Task - Low Complexity

TECHNIQUE was found to have a statistically significant effect on selection time, with a large effect size ($F_{2.379,47.581} = 26.061, p < .001, \eta_p^2 = 0.566$). *GravityZone* took the least time for selection when compared to most other techniques ($p < .001$), except *AlphaCursor* ($p = .078$).

There was no statistically significant difference between TECHNIQUES regarding error rates ($F_{6,120} = 0.945, p = .466$).

7.3.4 Repeat Task - High Complexity

TECHNIQUE was found to have a statistically significant effect on selection time, with a large effect size ($F_{3.282,65.646} = 18.412, p < .001, \eta_p^2 = 0.479$). *LassoGrid* was the fastest, but was similar to *FlowerCone* and *GridWall* ($p = 1.000$). *LassoGrid* was significantly faster than *AlphaCursor* ($p = .031$) and the remaining techniques ($p < .001$).

There was a statistically significant difference between TECHNIQUES regarding error rates ($F_{6,120} = 6.491, p < .001$). *AlphaCursor* ($p = .018$) and *GravityZone* ($p < .001$) had much lower error rates than *MagicBall*. *GravityZone* led to less error than *SmashProbe* ($p = .029$).

7.3.5 Search Task vs. Repeat Task

In terms of selection time, most techniques (all $p < .003$) had significant improvements in the repeat phase of the low complexity condition except *GridWall* ($p = .405$) and *MagicBall* ($p = .057$). For high complexity condition, there was no statistically significant effect of task on selection time for *GridWall* ($p = .970$) and *SmashProbe* ($p = .143$), but there was for all the others (*MagicBall*: $p = .031$ and others: $p < .001$).

Regarding error rates, only *MagicBall* ($p = .035$) and *SmashProbe* ($p = .043$) improved in the low complexity condition. No significant difference was revealed in the high complexity condition (all $p > .050$).

7.4 User Feedback Results

The overall easiness and intrusiveness of the techniques were calculated by averaging the 7-point Likert scale results. We also computed the mean ranking and counted the number of first/second place for each technique. The results from both questionnaires are summarized in Table 1.

In terms of the free-form feedback, the comments were mostly focused on *GridWall*, *MagicBall*, and *SmashProbe*. Several participants (N=7) felt *GridWall* was somewhat "boring" because it simply arranged all the objects in a 2D grid. In contrast, *SmashProbe* was seen as "fun" to use (N=4). Some participants thought *MagicBall* provided a good overview of the objects (N=4) but was quite difficult for selecting the target when the object number was high (N=3).

Table 1. The mean value (standard error) of easiness rating, intrusiveness rating, and preference ranking for all the techniques in Study 1. The last column shows the number of times a technique is ranked as the first/second. For Easy, higher is better; for Intrusiveness and Rank, lower is better.

Technique	Easy	Intrusiveness	Rank	#1/2
AlphaCursor	5.38 (0.33)	1.90 (0.34)	4.05 (0.41)	2/3
FlowerCone	5.81 (0.27)	1.95 (0.36)	3.38 (0.43)	4/3
GravityZone	5.86 (0.27)	1.76 (0.26)	3.05 (0.35)	5/3
GridWall	6.33 (0.16)	1.57 (0.36)	3.48 (0.42)	6/1
LassoGrid	5.76 (0.22)	1.86 (0.26)	3.86 (0.40)	1/7
MagicBall	4.33 (0.37)	3.19 (0.39)	5.19 (0.41)	0/3
SmashProbe	4.76 (0.34)	3.10 (0.28)	5.00 (0.47)	3/1

7.5 Summary and Discussion

The results show that performance improved for most techniques when participants moved from the search task to the repeat task. This is particularly true for the complex tasks, where selection time was significantly shortened in the repeat task. However, *GridWall* did not gain an advantage from the repeated selection, as the object order was randomized on the grid. *SmashProbe* did not improve significantly in the high complexity condition during the repeat phase. The selection phase of these techniques took a significantly longer time to complete when compared to the searching phase. As the repeat task was a replay of the previous task, the learning effect can also reduce the selection time and help users correct errors. Interestingly, the ranking of the techniques based on selection time almost did not change from the search task to the repeat task. This is likely caused by the fact that the first selection only narrowed down the participants' estimation of the "occurrence area" of the target, while some searching was still needed in the subsequent selection.

For low complexity tasks, *GravityZone* and *AlphaCursor* performed better (both with shorter selection time and lower error rates). *GridWall* also yielded good performance, whereas other techniques were shown to take more time or have higher error rates. One possible cause for this is that for simpler tasks, *GravityZone*, *AlphaCursor*, and *GridWall* can reveal the target quickly with straightforward manipulations, while techniques like *FlowerCone* and *LassoGrid* required an extra layer of area specification. The performance of *SmashProbe* was comparable to *FlowerCone* and *LassoGrid*. In contrast, *MagicBall* was the slowest, mostly because it required some precision to select the small proxies of objects.

For high complexity tasks, techniques that arranged the objects on a grid were the most successful in terms of selection performance, with *GridWall*, *LassoGrid*, and *FlowerCone* clearly outperforming other techniques. For instance, performance when using *AlphaCursor* and *GravityZone* suffered when the task got complex. Searching the target became difficult for participants, as once missing the target, which was surrounded by the sea of distractors, the participant had to move back and forth (the cursor of *AlphaCursor* or the object clusters of *GravityZone*) to search for them. Navigating to the correct depth where the target located was cumbersome. Similarly, *SmashProbe* performed poorly, as the target can sometimes "jump" to places where it was still fully-occluded by others.

Furthermore, participants found that if they kept spreading all objects in such dense environments, it would lead to significant distraction. The complex scenario also further exacerbated the problems with *MagicBall*. This is because participants needed to have very high precision for selecting the duplicates of the objects, while pressing the trigger on the controller could easily cause hand tremors [14], which can lead to the wrong selection.

Regarding easiness and intrusiveness, all techniques were rated better than the middle point of the 7-point Likert scale. Participants rated *GridWall* the easiest technique, which also caused the least distraction. However, participants felt bored when using this technique as it no longer felt like 3D interaction. *GravityZone*, *LassoGrid*, *FlowerCone*, and *AlphaCursor* all got positive feedback in term of these two scales. On the other hand, *MagicBall* and *SmashProbe* were rated lower, given the difficulty of selecting targets with these techniques. *MagicBall* can cause a wrong selection due to the handshaking, and *SmashProbe* might lead to an unexpected spread of the objects when performing the selection. However, they were both seen as interesting by the participants. *MagicBall* built a nice overview of the objects, while *SmashProbe* created a level of unexpectedness, which could be fun for gaming purposes [59]. With regard to preference ranking, *GravityZone* was ranked highest, followed by techniques that employed the grid feature and *AlphaCursor*.

Based on the study results and our observations, we extracted a set of design lessons for different kinds of scenarios and application purposes regarding fully-occluded target selection.

L1. Use techniques with the grid feature (*GridWall*, *LassoGrid*, and *FlowerCone*) for dense environments. Our results showed that these techniques had much better performance in complex tasks. However, according to the design space, be aware that these techniques would not preserve the original scene (the original locations of objects).

L2. Depth-based techniques (*AlphaCursor* and *GravityZone*) provide simple solutions to lower complexity tasks. They can also preserve the location information of objects. However, when many distractors are clustered with the target, it might be difficult for these techniques to navigate to the exact depth where the target is located.

L3. A smaller-scaled duplicate of the whole environment (like *MagicBall*) can help provide location awareness in virtual environments [79]. However, requiring users to perform direct selection on the small object proxies can pose challenges, such as hand tremors [14].

L4. It can be beneficial to use techniques that have some sort of unpredictability for recreational purposes (like *SmashProbe*). However, in dense and complex environments, such unexpectedness can obstruct the primary selection task. In addition, applying 3D features in a virtual environment rather than only using 2D surfaces (e.g., *GridWall*) could lead to a more enjoyable experience.

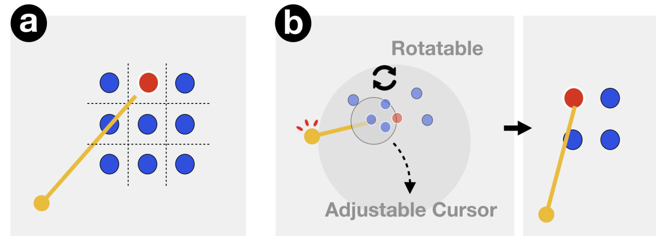

Fig. 6. (a) We implemented a selection enhancement technique on the grid layout, which would select the closest object to the ray; (b) *MagicBall+* embedded an adjustable cursor which could transform all object proxies into a grid layout for accurate selection.

After summarizing the findings from this first study, we were interested in refining the most promising techniques further. We also wanted to explore how specific environmental factors (like the size of the occurrence areas, occlusion layers, target depths, and object densities) would affect the performance of the techniques.

8 Technique Refinement

The seven techniques can be categorized into three sets: grid-based (*GridWall*, *LassoGrid*, and *FlowerCone*), depth-based (*GravityZone* and *AlphaCursor*), and others (*MagicBall* and *SmashProbe*). As the techniques in the different sets are better suited for different application purposes, and the ones within a set have similar strengths and weaknesses, we decided to improve them according to their general features. Based on the results and our observations from Study 1, we refined the techniques as follows.

We first improved the techniques that used grids (*GridWall*, *LassoGrid*, and *FlowerCone*). In the experiment, we found that the RayCasting technique for selecting objects on the grid sometimes led to errors during the fast-paced movements, as a correct selection was confirmed only when the ray was "crossing through" the target. Therefore, we decided to add selection enhancement techniques for RayCasting in the grid selection phase. We highlighted the nearest object to the ray and confirmed the selection on it when the user pressed the trigger, as this was shown to be the most efficient visual feedforward by recent work [11] (see Figure 6a). Additionally, to preserve depth information of the objects, we scaled the distances between the objects on the grid and the user according to the object's real distances to the user. This also adds some 3D features on the 2D grid surface. Although the visual size of the objects changed, our selection enhancement ensured that the effective size for selection was the same. Furthermore, we identified that randomizing the positions of the objects on the grid every time (for search and repeat tasks) was not efficient when there were a large number of objects. It could be more effective if objects were arranged by their distance to the ray (or center of the cone), from the closest to the farthest.

We also aimed to enhance the depth-based techniques (*GravityZone* and *AlphaCursor*).For some users, we observed that when the distractors in the front of the target were fairly close to it, navigating to the exact depth where the target located was laborious. Meanwhile, using a constant cursor speed might not be ideal for every user, as some might need it to be faster, while others want it to be slower. As a result, we made the cursor speed adjustable through the joystick input. The harder it was pushed/tilted, the faster the cursor became.

In addition, we found that most users had difficulties when selecting object proxies inside the mini-map, especially in the case where there was a large number of objects. To improve the selection, we combined the grid feature, which was shown to be effective for selection into *MagicBall*. Instead of employing the forward and backward movements of the joystick to translate the mini-map (which was not that useful according to our inspection), it was used to scale up and down the transparent point cursor, which was used for selecting objects. Once the selection trigger was pressed, and the scaled-up cursor enclosed more than one object proxy, the objects that were inside the cursor would be arranged onto a grid for the second phase of selection (see Figure 6b). Users could still select objects from the mini-map directly if only one object proxy collided the cursor.

We first picked two techniques, *LassoGrid* from the grid-based techniques and *GravityZone* from the depth-based techniques, according to the empirical performance and user feedback, and applied the refinements as mentioned above. We also improved *MagicBall*, as many users preferred the small overview of objects, and the main problem with the technique was the difficulty caused by selection. Consequently, we evaluated the three refined techniques (*LassoGrid+*, *GravityZone+*, and *MagicBall+*) in the second study.

9 Study 2 - In-Depth Evaluation

To have a more thorough understanding of how different environmental factors might affect the performance of the techniques, we conducted a second study based on three refined techniques (*GravityZone+*, *LassoGrid+*, and *MagicBall+*).

9.1 Environmental Factors

Initially, we were interested in five essential environmental factors (occurrence area, occlusion layer, environmental density, target depth, and density space) which can have a substantial impact on target selection with the different techniques. However, evaluating all of them might pose a high workload for participants.

As a result, we combined environmental density and density space to one single factor called area density, as both these factors are related to the number of distractors inside a space unit. Area density specified the density of the objects within the occurrence area, intending to maintain the same level of difficulty for techniques within the targeting area. We assumed that objects within the targeting area might raise more challenges than the ones that were spread around the whole space. In this case, the density of the objects within the whole environment (outside of the occurrence area) would be set as constant. We ended up with four environmental factors, which are OccurrenceArea, AreaDensity, OcclusionLayer, and TargetDepth.

9.2 Method

We recruited another set of 16 participants (9F/7M) between the ages of 20-32 (M = 24.6 $\pm$ 3.3) with different educational backgrounds from a local university campus. All participants had normal or corrected-to-normal vision. They rated their familiarity with VR as moderate (3.6 $\pm$ 1.7 on a 7-point scale). We used the same apparatus and devices as in the first study.

The study employed a within-subjects design with five factors: Technique (*GravityZone+*, *LassoGrid+*, and *MagicBall+*), OccurrenceArea (small and large), AreaDensity (low and high), OcclusionLayer (less and more), and TargetDepth (low and high). The details of the variables are summarized in the supplementary material. Three techniques appeared in a random sequence, while for each technique, we varied the four counterbalanced environmental factors. The techniques were well-distributed in terms of their order according to our post-hoc evaluation. In this study, only the search task was used, rather than including both search and repeat task, because 1) it decreased the workload of the participants, 2) we found techniques had similar rankings based on the selection time for both tasks, and 3) the search phase is likely to be more relevant to real application scenarios. In total we collected 3072 trials of data (16 participants $\times$ 3 techniques $\times$ 2 occurrence areas $\times$ 2 area densities $\times$ 2 occlusion layers $\times$ 2 target depths $\times$ 4 repetitions).

As with Study 1, we gathered selection time and error rate data, and observed each experiment. Additionally, we used two standardized questionnaires to assess the task workload and user experience. The workload was measured by RAW NASA-TLX [39], and the user experience was quantified by the short version of the User Experience Questionnaire (UEQ-S) [61]. These questionnaires are more comprehensive than the ones used in the first study, which were used for the simplicity of the experiment given the higher number of tested techniques. Both questionnaires were presented inside the virtual environment as previous work has shown that it can reduce study duration and user disorientation [62].

Table 2. The results from the short version of User Experience Questionnaires (UEQ-S) which outline the pragmatic quality, hedonic quality, and overall quality of each technique. In the table, ">avg." means "above-average", "exc." means "excellent".

Technique	Pragmatic	Hedonic	Overall
GravityZone+	1.31 (>avg.)	1.38 (>avg.)	1.34 (>avg.)
LassoGrid+	1.83 (exc.)	1.77 (good)	1.80 (exc.)
MagicBall+	1.56 (good)	2.05 (exc.)	1.80 (exc.)

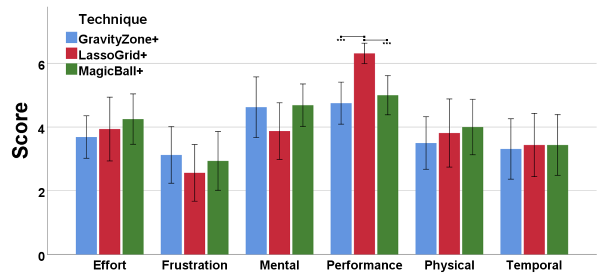

Fig. 7. The results from Raw NASA-TLX Questionnaires. Error bars indicate the 95% confidence interval. Statistical significant effects are marked (*** = p < .001).

The study lasted approximately 35 minutes for each participant. A similar procedure as the first study was used. Participants were compensated with a $10 voucher.

9.3 Results

As in the first study, we discarded the error trials (174 errors, 5.7%) and the outliers (78 trials, 2.5%) to analyze the selection time. We employed a RM-ANOVA with Greenhouse-Geisser correction for analyzing the effect of each factor. Pairwise comparisons with Bonferroni adjustment were used for technique comparison. Error rate data was transformed using ART [75] and was then analyzed through a RM-ANOVA. Regarding user feedback, we summarised the results from the questionnaires in Table 2 and Figure 7.

Since we were interested in how the techniques were affected by different environmental factors, we only present the effects and interactions related to the factor Technique.

9.3.1 Selection Time

A RM-ANOVA indicated that Technique ($F_{1.375,20.627} = 20.039, p < .001, \eta_p^2 = 0.572$) had a significant main effect on selection time, with a large effect size. A post-hoc test revealed that *LassoGrid+* was significantly faster than *GravityZone+* ($p < .001$) and *MagicBall+* ($p < .001$). *GravityZone+* was also indicated to be faster than *MagicBall+* ($p = .048$).

There were interaction effects between Technique $\times$ AreaDensity ($F_{1.356,20.336} = 6.090, p = .015, \eta_p^2 = 0.289$), Technique $\times$ OcclusionLayer ($F_{1.864,27.954} = 7.365, p = .003, \eta_p^2 = 0.329$), and Technique $\times$ TargetDepth ($F_{1.747,26.206} = 14.584, p < .001, \eta_p^2 = 0.493$), all with medium to large effect size. We present these interaction effects in Figure 8. No other interaction effects were found. Although there was no interaction between Technique and OccurrenceArea, OccurrenceArea itself did have a significant main effect on selection time ($F_{1,15} = 61.186, p < .001, \eta_p^2 = 0.803$).

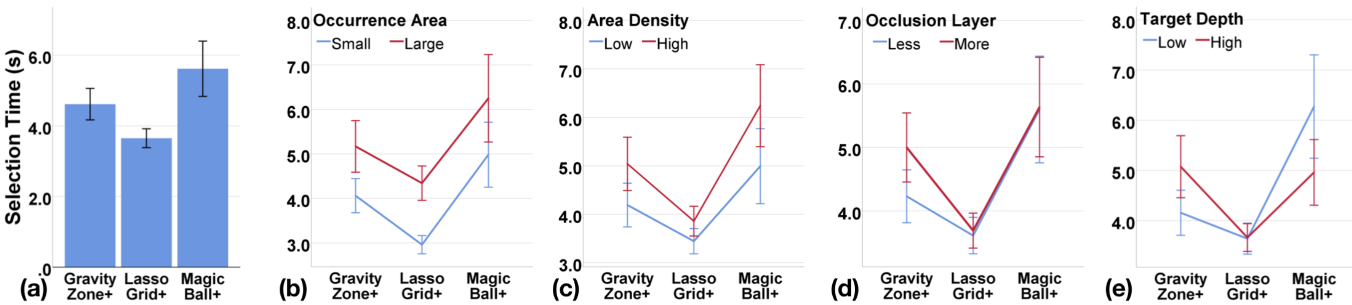

Fig. 8. Plots of selection time for the three improved techniques. These plots include techniques' overall selection time (a) and their selection time in different levels of Occurrence Area (b), Area Density (c), Occlusion Layer (d), and Target Depth (e). Error bars indicate the 95% confidence interval.

9.3.2 Error Rate

TECHNIQUE had a significant main effect on error rate ($F_{2,705} = 8.027, p < .001$). A post-hoc test showed that *GravityZone+* (5.96%, $p = .032$) and *LassoGrid+* (3.51%, $p < .001$) had statistically significantly lower error rates than *MagicBall+* (7.52%), with no significant difference between the two ($p = .490$).

A RM-ANOVA also revealed further interactions among the other factors. However, as the error rate was relatively low for all techniques (according to [11, 66]) and better techniques clearly outperformed the worse ones in terms of performance (lower error rate techniques as well as lower selection time), we do not examine these results here in further detail. For detailed statistics, please refer to our supplementary materials.

9.4 Summary and Discussion

In this section, we first examine the influence of the four environmental factors on the three techniques and then compare the input techniques from different perspectives.

Occurrence area had a similar impact on all techniques—as it increased, the selection time of the three techniques also increased significantly. This was expected as an inaccurate prior estimate of where the target might be located (larger occurrence area) leads to a higher search time, prolonging the selection process.

Area density affected the selection performance of all techniques, but the magnitude of the effect was different, as indicated by the significant interaction effect. As the density increased, search and selection for *GravityZone+* and *MagicBall+* became much more difficult than for *LassoGrid+*. One potential reason for this difference is the fact that organizing the objects on a grid-like 2D layout demanded less effort for searching rather than the original clustered and overlapped 3D arrangements [19].

Occlusion layer only influenced the performance of *GravityZone+*. With *GravityZone+*, it can feel somewhat cumbersome navigating through multi-layers of distractors. However, when using *LassoGrid+*, which arranged objects in the target area on a grid, and *MagicBall+*, which provided a quick overview of all the objects, users were not impeded by these layers at all.

Target depth affected the performance of *GravityZone+* and *MagicBall+*, but not of *LassoGrid+*. *LassoGrid+* was invariant to the change of target depth, as it transformed the 3D region to a grid, regardless of the real depth of the objects. Although we added depth information on *LassoGrid+*, it only changed the visual size of the objects, but not its effective size with the selection enhancement technique [35, 66]. For *GravityZone+*, when the target was located further, participants were required to navigate more to reach it, thus induced longer selection time. However, participants spent more time in selecting the lower depth target using *MagicBall+* than the higher depth ones. This was because participants were observing the whole environment from the outside of the mini-map, lower depth target actually looked farther away. Hence, there might be more distractors on the way of getting the goal target.

After seeing how each environmental factor affected the performance of the techniques, we compared the techniques in terms of different measurements below. The performance data were consistent in terms of selection time and error rates. *LassoGrid+* had the lowest selection time and error rate, while for *MagicBall+* they were the highest. The NASA-TLX results also show a similar trend. Participants were more satisfied with their performance and had lower frustration and mental workload levels when using *LassoGrid+*. Concerning the UEQ-S results, *LassoGrid+* was shown to have excellent pragmatic value, while *MagicBall+* was rated outstanding in the hedonic quality. They both had excellent overall quality. However, *GravityZone+* was rated just above-average on all aspects of the UEQ-S. It seemed to suffer from the “middle children syndrome” [43], where it did not look as innovative as *MagicBall+* and was not as effective as *LassoGrid+*. Therefore, its ratings from the participants were relatively low.

10 DESIGN RECOMMENDATIONS

Based on the results from both studies, we distill design recommendations regarding choosing input techniques for the selection of fully-occluded target in virtual environments.

R1. When the goal of the task is rapid selection, we suggest using grid-based techniques (*GridWall*, *FlowerCone*, and *LassoGrid+*) to ensure optimal user performance. Use *LassoGrid+* when it is difficult to decide which one of them to use, as it allows users to define their estimate of where targets might occur freely, and only one trigger/button will be needed for the whole selection process. Consider adding selection enhancement techniques (like highlighting the closest object) to improve performance further.

R2. When maintaining the object location information is essential (e.g., 3D plots), we recommend using depth-based techniques (*AlphaCursor* and *GravityZone+*) or *MagicBall+*. Our results indicate that *GravityZone+* should be favored if better performance is needed. *AlphaCursor* and *MagicBall+* can be used when it is not desirable to move the objects in the scene.

R3. If the technique is used for recreational purposes (like game applications), consider use *SmashProbe* and *MagicBall+* as they are more exciting or have higher hedonic quality. However, avoid using *SmashProbe* when there are too many objects in the scene, as it could be very distracting.

R4. Be sure to consider the environmental factors (occlusion layers, target depths, object densities, and the estimation of target location) of the application and how they might influence the performance of the technique. If the environment constantly changes, as a rule of thumb, use *LassoGrid+* as it was shown to be relatively robust in terms of performance.

11 DEMONSTRATIONS

Based on our findings, we have developed two proof-of-concept demonstrations in VR showing the techniques in real application scenarios (see Figure 9). The first demo shows an ocean exploration scenario in VR, which belongs to the case of exploring complex 3D data visualizations. Users are immersed under the ocean and surrounded by a large number of underwater creatures. With our techniques, they can select an animal of interest that lives in certain areas or is hidden by corals to delve into its detailed information (like name, habitat, life cycles, etc.). A similar scenario would be to explore specific locations occluded by buildings in a 3D city visualization. The second demo mimics a 3D modeling scenario. Users can acquire fully-occluded objects in the scene and perform consequent manipulations like translation and duplication. Both applications are demonstrated in the supplementary video.

Fig. 9. (a) In the sea exploration scenario, a user used *LassoGrid+* to learn about animals (which might be fully-occluded) living within a particular area. (b) *AlphaCursor* reveals the hidden tree in the modeling scene.

12 Limitations

We have identified several limitations in our work. First, for simplicity, we simulated a user's estimated area of where a target might occur only in a circular form. However, in a real-world scenario, multiple occurrence areas can exist, and they can be in any shape, even with some depth.

Second, we did not fine-tune the parameters of all the techniques through user studies, as it was not the primary goal of this work. For example, instead of arranging objects on a grid, other layouts are also possible (e.g., rings [10]), which could further improve the performance of the techniques.

Third, we did not include unselectable objects in the scene, as we envision a superimposed scenario that culls out the unselectable objects for the ease of selection. However, future work might want to investigate how unselectable objects can be embedded into the scene and how various properties related to these objects (like sizes and placements) can affect the selection.

Fourth, our experiments feature more abstract tasks that enabled us to control the variables of interest precisely, however, we did not evaluate technique performance under practical scenarios. To strike a balance between internal and external validity of our findings, though two proof-of-concept demonstrations are provided, more work is necessary to understand how the techniques can perform and how we can adjust them in realistic workflows. For example, future work can explore how the techniques could be applied to disambiguate vertex or edge selection in 3D modeling applications.

13 Conclusion

In this paper, we explored fully-occluded target selection in virtual reality environments. Based on the existing literature on the topic, we highlighted three open challenges within this research topic in terms of problem formulation, combining occlusion visualization with selection techniques, and in-depth evaluation. To address them, we first framed a general problem-solving strategy and, according to that, devised the design space. We then designed seven potential techniques and evaluated them through a user study.

Based on the study results, we derived design implications and refined the most promising techniques. We conducted a second study to analyze how four environmental factors (occlusion layers, target depths, object densities, and the estimation of target locations) influence technique performance. Based on our findings, we offer a set of distilled recommendations for future virtual reality systems that offer fully-occluded target selection. We believe our design approaches and proposed techniques can trigger the creation of exciting user interfaces and applications related to fully-occluded selection. Future work can optimize further the techniques, as well as develop new methods for selecting fully-occluded targets in VR.

Acknowledgments

We thank our participants for their interest in the project and insightful discussions. We also appreciate the reviewers for their professionalism and dedication that helped improve our paper. This research was supported by the Melbourne Research Scholarship provided by The University of Melbourne.

References

[1] D. H. Abelow. Reality alternate, Nov. 10 2015. US Patent 9,183,560.

[2] Agustina and C. Sun. Xpointer: An x-ray telepointer for relaxed-space-time wysiwis and unconstrained collaborative 3d design systems. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pp. 729–740. ACM, New York, NY, USA, 2013. doi: 10.1145/2441776.2441857

[3] F. Argelaguet and C. Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121 – 136, 2013. doi: 10.1016/j.cag.2012.12.003

[4] F. Argelaguet, A. Kunert, A. Kulik, and B. Froehlich. Improving co-located collaboration with show-through techniques. In *Proceedings of the 2010 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 55–62, March 2010. doi: 10.1109/3DUI.2010.5444719

[5] J. Arora, A. Saini, N. Mehra, V. Jain, S. Shrey, and A. Parnami. Virtualbricks: Exploring a scalable, modular toolkit for enabling physical manipulation in vr. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 56:1–56:12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300286

[6] B. Bach, R. Sicat, J. Beyer, M. Cordeil, and H. Pfister. The hologram in my hand: How effective is interactive exploration of 3d visualizations in immersive tangible augmented reality? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):457–467, Jan 2018. doi: 10.1109/TVCG.2017.2745941

[7] F. Bacim, R. Kopper, and D. A. Bowman. Design and evaluation of 3d selection techniques based on progressive refinement. *International Journal of Human-Computer Studies*, 71(7):785 – 802, 2013. doi: 10.1016/j.ijhcs.2013.03.003

[8] F. Bacim, M. Nabiyouni, and D. A. Bowman. Slice-n-swipe: A free-hand gesture user interface for 3d point cloud annotation. In *Proceedings of the 2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 185–186. IEEE, 2014. doi: 10.1109/3DUI.2014.6798882

[9] F. Bacim de Araujo e Silva. *Increasing Selection Accuracy and Speed through Progressive Refinement*. PhD thesis, Virginia Tech, 2015.

[10] G. Bailly, E. Lecolinet, and L. Nigay. Visual menu techniques. *ACM Computer Survey*, 49(4):60:1–60:41, Dec. 2016. doi: 10.1145/3002171

[11] 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*, CHI '19, pp. 101:1–101:12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300331

[12] L. Besançon, M. Sereno, L. Yu, M. Ammi, and T. Isenberg. Hybrid touch/tangible spatial 3d data selection. In *Computer Graphics Forum*, vol. 38, pp. 553–567. Wiley Online Library, 2019. doi: 10.1111/cgf.13710

[13] C. Bichlmeier, S. M. Heining, M. Feuerstein, and N. Navab. The virtual mirror: A new interaction paradigm for augmented reality environments. *IEEE Transactions on Medical Imaging*, 28(9):1498–1510, Sep. 2009. doi: 10.1109/TMI.2009.2018622

[14] D. Bowman, C. Wingrave, J. Campbell, and V. Ly. Using pinch gloves (tm) for both natural and abstract interaction techniques in virtual environments. 2001.

[15] S. Bruckner and M. E. Groller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, Sep. 2006. doi: 10.1109/TVCG.2006.140

[16] M. Burns and A. Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. In *Proceedings of the ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pp. 154:1–154:7. ACM, New York, NY, USA, 2008. doi: 10.1145/1457515.1409107

[17] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *Proceedings IEEE Symposium on Information Visualization '96*, pp. 46–53, Oct 1996. doi: 10.1109/INFVIS.1996.559215

[18] J. Cashion. Intelligent selection techniques for virtual environments. 2014.

[19] J. Cashion, C. Wingrave, and J. J. L. Jr. Dense and dynamic 3d selection for game-based virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):634–642, April 2012. doi: 10.1109/TVCG.2012.40

[20] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson. Immersive analytics. In *Proceedings of the 2015 Big Data Visual Analytics (BDVA)*, pp. 1–8, Sep. 2015. doi: 10.1109/BDVA.2015.7314296

[21] O. Chapuis, J.-B. Labrune, and E. Pietriga. Dynaspot: Speed-dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1391–1400. ACM, New York, NY, USA, 2009. doi: 10.1145/1518701.1518911

[22] M. C. Chuah, S. F. Roth, J. Mattis, and J. Kolojejchick. Sdm: Selective dynamic manipulation of visualizations. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp. 61–70, 1995.

[23] C. Coffin and T. Hollerer. Interactive perspective cut-away views for general 3d scenes. In *Proceedings of the 3D User Interfaces (3DUI'06)*, pp. 25–28, March 2006. doi: 10.1109/VR.2006.88

[24] C. Correa, D. Silver, and M. Chen. Illustrative deformation for data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1320–1327, Nov 2007. doi: 10.1109/TVCG.2007.70565

[25] H. G. Debarba, J. G. Grandi, A. Maciel, L. Nedel, and R. Boulic. Disambiguation canvas: A precise selection technique for virtual environments. In *Proceedings of the Human-Computer Interaction – INTERACT 2013*, pp. 388–405. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-40477-1_24

[26] S. Deng, J. Chang, S.-M. Hu, and J. J. Zhang. Gaze modulated disambiguation technique for gesture control in 3d virtual objects selection. In *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, pp. 1–8, June 2017. doi: 10.1109/CYBConf.2017.7985779

[27] J. Diepstraten, D. Weiskopf, and T. Ertl. Transparency in interactive technical illustrations. *Computer Graphics Forum*, 21(3):317–325, 2002. doi: 10.1111/1467-8659.t01-1-00591

[28] J. Diepstraten, D. Weiskopf, and T. Ertl. Interactive cutaway illustrations. *Computer Graphics Forum*, 22(3):523–532, 2003. doi: 10.1111/1467-8659.t01-3-00700

[29] N. Elmqvist. Balloonprobe: Reducing occlusion in 3d using interactive space distortion. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '05, pp. 134–137. ACM, New York, NY, USA, 2005. doi: 10.1145/1101616.1101643

[30] N. Elmqvist, U. Assarsson, and P. Tsigas. Dynamic transparency for 3d visualization: design and evaluation. *The International Journal of Virtual Reality*, 1(8):65–78, 2009.

[31] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management techniques. In *Proceedings of the 2007 IEEE Virtual Reality Conference*, pp. 51–58, March 2007. doi: 10.1109/VR.2007.352463

[32] N. Elmqvist and M. E. Tudoreanu. Occlusion management in immersive and desktop 3d virtual environments: Theory and evaluation. *Internaltional Journal of Virtual Reality*, 6(2):21–32, 2007.

[33] T. T. Elvins, D. R. Nadeau, and D. Kirsh. Worldlets—3d thumbnails for wayfinding in virtual environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, p. 21–30. Association for Computing Machinery, New York, NY, USA, 1997. doi: 10.1145/263407.263504

[34] S. K. Feiner and D. D. Seligmann. Cutaways and ghosting: satisfying visibility constraints in dynamic 3d illustrations. *The Visual Computer*, 8(5):292–302, Sep 1992. doi: 10.1007/BF01897116

[35] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954. doi: 10.1037/h0055392

[36] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pp. 95–96. ACM, New York, NY, USA, 1996. doi: 10.1145/237091.237105

[37] M. Funk, F. Müller, M. Fendrich, M. Shene, M. Kolvenbach, N. Dobbertin, S. Günther, and M. Mühlhäuser. Assessing the accuracy of point & teleport locomotion with orientation indication for virtual reality using curved trajectories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 147:1–147:12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300377

[38] 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*, UIST '06, pp. 3–12. ACM, New York, NY, USA, 2006. doi: 10.1145/1166253.1166257

[39] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, pp. 904–908, 2006. doi: 10.1177/154193120605000909

[40] T. Hirzle, J. Gugenheimer, F. Geiselhart, A. Bulling, and E. Rukzio. A design space for gaze interaction on head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 625:1–625:12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300855

[41] D. Kalkofen, E. Veas, S. Zollmann, M. Steinberger, and D. Schmalstieg. Adaptive ghosted views for augmented reality. In *Proceedings of the 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1–9. IEEE, 2013. doi: 10.1109/ISMAR.2013.6671758

[42] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and accurate 3d selection by progressive refinement. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 67–74, March 2011. doi: 10.1109/3DUI.2011.5759219

[43] J. Kotin. *Getting started: An introduction to dynamic psychotherapy*. Jason Aronson, 2004.

[44] G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pp. 482–487. ACM, New York, NY, USA, 1993. doi: 10.1145/169059.169426

[45] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, and M. Billinghurst. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 81:1–81:14. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173655

[46] 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.

[47] K. Lilija, H. Pohl, S. Boring, and K. Hornbæk. Augmented reality views for occluded interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 446:1–446:12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300676

[48] M. A. Livingston, J. E. Swan, J. L. Gabbard, T. H. Hollerer, D. Hix, S. J. Julier, Y. Baillot, and D. Brown. Resolving multiple occluded layers in augmented reality. In *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 56–65. IEEE, 2003. doi: 10.1109/ISMAR.2003.1240688

[49] D. Mendes, D. Medeiros, E. Cordeiro, M. Sousa, A. Ferreira, and J. Jorge. Precious! out-of-reach selection using iterative refinement in vr. In *Proceedings of the 2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 237–238, March 2017. doi: 10.1109/3DUI.2017.7893359

[50] M. R. Mine. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept*, 1995.

[51] A. Olwal and S. Feiner. The flexible pointer: An interaction technique for augmented and virtual reality. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST*, pp. 81–82, 2003.

[52] J. Petford, I. Carson, M. A. Nacenta, and C. Gutwin. A comparison of notification techniques for out-of-view objects in full-coverage displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 58:1–58:13. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300288

[53] J. Petford, M. A. Nacenta, and C. Gutwin. Pointing all around you: Selection performance of mouse and ray-cast pointing in full-coverage displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 533:1–533:14. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3174107

[54] B. Preim, A. Baer, D. Cunningham, T. Isenberg, and T. Ropinski. A survey of perceptually motivated 3d visualization of medical image data. *Computer Graphics Forum*, 35(3):501–525, 2016. doi: 10.1111/cgf.12927

[55] G. Ramos, G. Robertson, M. Czerwinski, M. Czerwinski, D. Tan, P. Baudisch, K. Hinckley, K. Hinckley, and M. Agrawala. Tumble! splat! helping users access and manipulate occluded content in 2d drawings. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pp. 428–435. ACM, New York, NY, USA, 2006. doi: 10.1145/1133265.

1133351
[56] G. Ren and E. O'Neill. 3d selection with freehand gesture. *Computers & Graphics*, 37(3):101 – 120, 2013. doi: 10.1016/j.cag.2012.12.006
[57] H. Ro, S. Chae, I. Kim, J. Byun, Y. Yang, Y. Park, and T. Han. A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2873–2878. IEEE, 2017. doi: 10.1109/SMC.2017.8123063
[58] M. Röhlig, M. Luboschik, and H. Schumann. Visibility widgets for unveiling occluded data in 3d terrain visualization. *Journal of Visual Languages & Computing*, 42:86 – 98, 2017. doi: 10.1016/j.jvlc.2017.08.008
[59] R. Rouse III. *Game design: Theory and practice*. Jones & Bartlett Learning, 2010.
[60] J. Sauro and J. S. Dumas. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1599–1608. ACM, New York, NY, USA, 2009. doi: 10.1145/1518701.1518946
[61] M. Schrepp, A. Hinderks, and J. Thomaschewski. Design and evaluation of a short version of the user experience questionnaire (ueq-s). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6):103–108, 2017. doi: 10.9781/ijimai.2017.09.001
[62] V. Schwind, P. Knierim, N. Haas, and N. Henze. Using presence questionnaires in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 360:1–360:12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300590
[63] B. Shneiderman. Why not make interfaces better than 3d reality? *IEEE Computer Graphics and Applications*, 23(6):12–15, Nov 2003. doi: 10.1109/MCG.2003.1242376
[64] L. Sidenmark, C. Clarke, X. Zhang, J. Phu, and H. Gellersen. Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376438
[65] R. B. Smith. Experiences with the alternate reality kit: An example of the tension between literalism and magic. *SIGCHI Bulletin*, 18(4):61–67, May 1986. doi: 10.1145/1165387.30861
[66] R. W. Soukoreff and I. S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies*, 61(6):751 – 789, 2004. doi: 10.1016/j.ijhcs.2004.09.001
[67] 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*, CHI '95, pp. 265–272. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995. doi: 10.1145/223904.223938
[68] J. Sun and W. Stuerzlinger. Selecting and sliding hidden objects in 3d desktop environments. 2019. doi: 10.20380/GI2019.08
[69] R. J. Teather and W. Stuerzlinger. Pointing at 3d target projections with one-eyed and stereo cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 159–168. ACM, New York, NY, USA, 2013. doi: 10.1145/2470654.2470677
[70] R. Trueba, C. Andujar, and F. Argelaguet. Complexity and occlusion management for the world-in-miniature metaphor. In *Smart Graphics*, pp. 155–166. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
[71] H. Tu, S. Huang, J. Yuan, X. Ren, and F. Tian. Crossing-based selection with virtual reality head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 618:1–618:14. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300848
[72] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In *Proceedings of the 2007 IEEE Symposium on 3D User Interfaces*, March 2007. doi: 10.1109/3DUI.2007.340783
[73] E. Velloso, M. Carter, J. Newn, A. Esteves, C. Clarke, and H. Gellersen. Motion correlation: Selecting objects by matching their movement. *ACM Transactions on Computer-Human Interaction*, 24(3), Apr. 2017. doi: 10.1145/3064937
[74] A. M. Webb, A. Kerne, Z. Brown, J.-H. Kim, and E. Kellogg. Layerfish: Bimanual layering with a fisheye in-place. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, ISS '16, pp. 189–198. ACM, New York, NY, USA, 2016. doi: 10.1145/2992154.2992171
[75] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 143–146. ACM, New York, NY, USA, 2011. doi: 10.1145/1978942.1978963
[76] M.-L. Wu and V. Popescu. Multiperspective focus+context visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(5):1555–1567, May 2016. doi: 10.1109/TVCG.2015.2443804
[77] H. P. Wyss, R. Blach, and M. Bues. isith - intersection-based spatial interaction for two hands. In *Proceedings of the 3D User Interfaces*, 3DUI '06, pp. 59–61. IEEE Computer Society, Washington, DC, USA, 2006. doi: 10.1109/VR.2006.93
[78] Y. Yan, C. Yu, X. Ma, X. Yi, K. Sun, and Y. Shi. Virtualgrasp: Leveraging experience of interacting with physical objects to facilitate digital object retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 78:1–78:13. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173652
[79] D. Yu, H.-N. Liang, K. Fan, H. Zhang, C. Fleming, and K. Papangelis. Design and evaluation of visualization techniques of off-screen and occluded targets in virtual reality environments. *IEEE Transactions on Visualization and Computer Graphics*, 2019. doi: 10.1109/TVCG.2019.2905580
[80] D. Yu, H.-N. Liang, X. Lu, K. Fan, and B. Ens. Modeling endpoint distribution of pointing selection tasks in virtual reality environments. *ACM Transactions on Graphics*, 38(6):13, November 2019. doi: 10.1145/3355089.3356544
[81] D. Yu, Q. Zhou, B. Tag, T. Dingler, E. Velloso, and J. Goncalves. Engaging participants during selection studies in virtual reality. In *Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 500–509. IEEE, 2020. doi: 10.1109/VR46266.2020.00071
[82] S. Zhai, W. Buxton, and P. Milgram. The partial-occlusion effect: Utilizing semitransparency in 3d human-computer interaction. *ACM Transaction on Computer-Human Interaction*, 3(3):254–284, Sept. 1996. doi: 10.1145/234526.234532