



# Global Scene Filtering, Exploration, and Pointing in Occluded Virtual Space

Yuan Chen<sup>1,2(✉)</sup>, Junwei Sun<sup>2</sup>, Qiang Xu<sup>2</sup>, Edward Lank<sup>1</sup>, Pourang Irani<sup>3</sup>,  
and Wei Li<sup>2</sup>

<sup>1</sup> University of Waterloo, Waterloo, ON, Canada  
{y2238che,lank}@uwaterloo.ca

<sup>2</sup> Huawei Technologies, Markham, ON, Canada  
{junwei.sun,qiang.xu1,wei.li.crc}@huawei.com

<sup>3</sup> University of Manitoba, Winnipeg, MB, Canada  
irani@cs.umanitoba.ca

**Abstract.** Target acquisition in an occluded environment is challenging given the omni-directional and first-person view in virtual reality (VR). We propose Solar-Casting, a global scene filtering technique to manage occlusion in VR. To improve target search, users control a reference sphere centered at their head through varied occlusion management modes: Hide, SemiT (Semi-Transparent), Rotate. In a preliminary study, we find SemiT to be better suited for understanding the context without sacrificing performance by applying semi-transparency to targets within the controlled sphere. We then compare Solar-Casting to highly efficient selection techniques to acquire targets in a dense and occluded VR environment. We find that Solar-Casting performs competitively to other techniques in known environments, where the target location information is revealed. However, in unknown environments, requiring target search, Solar-Casting outperforms existing approaches. We conclude with scenarios demonstrating how Solar-Casting can be applied to crowded and occluded environments in VR applications.

## 1 Introduction

While Virtual Reality (VR) presents end-users with rich 3D environments, an on-going challenge within VR involves the acquisition of targets to support subsequent manipulation. As a result, 3D selection in VR remains an active research area, particularly given the variety of possible 3D environment configurations

Y. Chen—The work was done when the author was an intern at Huawei Canada.

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-85607-6\\_11](https://doi.org/10.1007/978-3-030-85607-6_11)) contains supplementary material, which is available to authorized users.

necessitating interactive support [2]. Most VR systems use some variant of Ray-Casting for 3D pointing [5], which enables users to select using a virtual ray.

Among the many concerns with 3D target acquisition, targeting occluded objects is a significant challenge with Ray-Casting [2,20]. Two types of occlusion exist within virtual environments: an object is either partially-occluded or fully-occluded by other objects in the scene. Each type of occlusion has its own issues. While a partially-occluded target can be seen, selecting it efficiently and accurately is challenging because of distracting neighbour targets [9,23]. An example of partial occlusion might be selecting a single piece of data from a 3D visualization. As for fully-occluded scenarios, since targets are completely hidden, it is hard to locate targets, resulting in subsequent difficulties selecting and interacting. Examples of this include tasks such as browsing 3D models [34] or interacting with 3D scenes.

To motivate this work, we have identified two design requirements (DRs) absent from existing Ray-Casting techniques based on the presented literature. These are **(1)** the lack of ability to handle both partial-occlusion and full-occlusion in virtual environments; and **(2)** the need to explore the environment and dynamically control the interaction space to overcome visual occlusion. These DRs are absent because target acquisition techniques that require some augmentation to control depth [4,20] or progressive refinement [9,23] demand one degree of control such that adding another degree of control (to limit or overcome occlusion visually or trajectoryally) on the occlusion space will increase the complexity of manipulating the controller. It is also difficult to manage visual occlusion and selection-based disambiguation because target selection is a local interaction, whereas visual occlusion – and techniques to allow users to overcome it – require global interaction in the immersive environment.

Motivated by these design requirements, we propose Solar-Casting, a 3D pointing technique designed to support rapid, and low-error target selection across different levels of occlusion. We use a reference sphere or *Solar* that is centered at the user’s head and works as an interactive volume. Users can effortlessly control the radius of *Solar* with a touch enabled surface. The sphere plays a vital role. It first filters inner items, by either hiding (Hide) or turning them semi-transparent (SemiT). This has the effect of making items outside the egocentric sphere visible to users, thereby reducing the effects of occlusion. In one variant (Rotate), a simple manipulation on the touch surface rotates items on *Solar*’s surface; in another a ray or other interaction metaphor can be used to select objects on the surface. Finally, since this occurs uniformly, by simply looking around, the user can identify items of interest.

To evaluate Solar-Casting, we conducted two experiments using different occlusion levels. In the first study, we compared Solar-Casting with existing Ray-Casting techniques in an occluded environment where a visual cue was presented so users could quickly locate the target. Our results showed that Solar-Casting had competitive performance to competing techniques. In the second study, we evaluated Solar-Casting without any visual cues such that users needed to search for occluded targets. The results highlight Solar-Casting’s strengths in occluded environments, particularly increasing accuracy in dense and highly-

occluded instances. In summary, our contributions include: (1) proposing an efficient pointing technique for selecting targets at different occlusion levels; (2) conducting experiments to evaluate and investigate Solar-Casting’s pros and cons in occluded environments with/without visual guidance.

## 2 Related Work

Numerous techniques have been proposed for 3D selection in virtual environments [2]. Ray-Casting [24] and the virtual hand [27] appear as two key 3D pointing metaphors [36]. Bowman et al. [7] introduce Ray-Casting with reeling to add a single degree of freedom, making it possible to change the length of the ray. In the ray-depth technique [37], users are able to adjust control display (C/D) ratio while adjusting the depth of the ray. Both Ray-Casting and virtual hand techniques offer complementary strengths and weaknesses [12, 36, 43]. Ray-Casting works well with distant objects, but the virtual hand needs augmentations, such as the Go-Go technique [35] to acquire out-of-reach objects. Argelaguet et al. [1] propose Ray-Casting from the eye, where a ray originates from the user’s eye, yet is controlled by their hand. They found that Ray-Casting from the eye was helpful for selection tasks in cluttered environments. A variation of using the head as a focal point for cursor manipulation has also been adopted in AR headsets such as the Microsoft HoloLens.

While the majority of techniques are tuned for selecting visible targets virtual environments [2], techniques for selecting occluded targets through transparency or cut-outs have received some attention. Considering first, transparency, the BalloonProbe [15] uses an invisible or transparent (wireframe) sphere that forces distortion in a 3D virtual environment, reducing occlusion. XPointer [41] is an X-ray mouse cursor technique for 3D selection, which enables users to select initially hidden objects via object-based transparency by changing the cursor’s penetration. Tilt-Casting [33] uses a virtual plane as a filtering surface, and objects above the surface are invisible, allowing objects on and below the plane to be seen and selected, and Sun et al. [42, 44] propose a layer-based transparency technique to select and manipulate objects that are originally fully occluded. Finally, Elmqvist et al. [16] propose an image space algorithm to achieve dynamic transparency for managing occlusion of important target objects in 3D. Cut-away techniques [13, 34] permit a user to look “through” occluding objects by interactively cutting holes into the occluding geometry. Similarly, the Cut Plane technique [30] uses a plane to split the entire scene, makes one side of the plane translucent, and reveals occluded targets on the other side. Many of these techniques are either geared toward 3D views [33], are tuned to select occluded targets at a known location [15, 41], or provide limited support for visual search during selection tasks [42, 44].

Beyond transparency and cut-outs, there exist a number of other input or view manipulations that allow users to select partially occluded targets or that rearrange targets spatially to handle occlusion. For example, the flexible pointer [18] bends a ray to avoid obstacles, thus allowing the user to select partially, but not fully occluded objects. Similarly, Outline Pursuits [40] enables

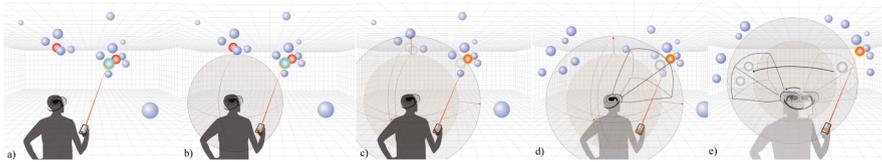
users to select a partly occluded target by following its outline with their gaze. Flower ray [20] presents all intersected targets with a ray in a marking menu and allows users to select the intended target. SQUAD [3, 23] adopts a similar design. Users first specify a spherical volume containing all targets of interest and then refine the initial selection progressively by selecting the subset of objects containing the desired one from a four-item menu until the target is finally selected. Expand [9] is an extension of SQUAD, where objects are placed in a grid for users to choose from. In a more dense environment (e.g. point clouds), the Slicing Volume [28] supports selection of occluded targets by first defining a specific region, followed by intersecting desired targets with a slicing plane. Most recently, Yu et al. [52] propose various Ray-Casting techniques to select fully occluded targets, including filtering targets globally or rearranging targets in a local space. However, all of these view manipulation techniques are designed based on the assumption that targets are either always at least partially visible to the users, that their location is known a priori, or that their location context can be disrupted without cost. In many scenarios, we cannot make any of these assumptions.

Regardless of how a previously occluded target becomes visible, the next step is to acquire it precisely. This is challenging for Ray-Casting techniques in virtual reality due to unintentional hand tremor and distant target location magnifying small rotation changes into large-scale distant movements. Grossman et al. [20] adopt various disambiguation mechanisms (e.g. depth-aware bubble cursor, and two-step selection) into Ray-Casting techniques so users can disambiguate different targets along a ray and select targets effectively. The bubble cursor is perhaps the most popular selection facilitation technique [19, 47]. Alongside its use by Vanacken [47], in RayCursor [4], a 3D bubble cursor is implemented to select the closest object to the cursor when users manually move a cursor on the ray to disambiguate object selection, and Moore et al. [29] find that choosing the closest target (de facto a bubble cursor technique) performs the best for virtual hand pointing. More recently, Lu et al. [25] investigate multiple variants of the bubble mechanism for Ray-Casting, i.e., selecting the closest object to the ray or cursor, and argue that the bubble mechanism significantly improves the performance and preference for Ray-Casting selection.

Other selection mechanisms have also been explored. Sun et al. [45] proposed a manual disambiguation mechanism for positioning 3D objects in depth. Multiple degrees-of-freedom (DOF) are often implemented to manipulate a ray or other virtual representations, such as a plane or sphere, and then perform precise 3D target acquisition [21, 22, 33]. Finally, multiple devices have also been explored: ARPen [48] uses a pen and smartphone to perform image plane selection [32] in a AR system; and Slicing Volume [28] uses controllers, a pen and a tablet to select targets intersecting with a plane in a scalable region. In contrast, our technique, Solar-Casting, uses a single touch surface (in our case, a smartphone display that also drives the VR display) to control the necessary degrees-of-freedom to simultaneously control the interaction space, remove occlusion, explore the scene, and, finally, select targets precisely in an occluded VR environment.

### 3 Solar-Casting

Our Solar-Casting technique is built in several steps. We begin by creating our own instance of the Ray-Casting technique with an interactive sphere to support global filtering and pointing with one control action. We design three global interaction modes: SemiT (Semi-Transparent), Hide and Rotate, and then evaluate these modes in a preliminary study. Last, We add head control to constrain the interaction region of the global occlusion function and two facilitation techniques for fast and stable manipulation of the cursor and the ray, as in [4]. This section describes each of these aspects of Solar-Casting in turn.



**Fig. 1.** Illustration of Solar-Casting with SemiT mode: (a) a red sphere is occluded by blue spheres, when *Solar* turns on (b) and users move up on the touchscreen to increase *Solar*'s radius, objects inside *Solar* and within field-of-view (fov) become translucent (c&d), so users can easily select the occluded target. For objects inside *Solar* but outside fov, their status will automatically update after entering fov (e). (Color figure online)

#### 3.1 Solar Metaphor

As we observe the evolution of pointing and selecting in 3D environments, we see a clear progression from using rays [4, 18, 41], to surfaces [30, 33] and volumes [28] to enable filtering and target selection. As existing Ray-Casting techniques that support depth manipulation already require a degree of control [4, 20], one challenge is that to support occlusion management, at least one other degree of control is required; otherwise, the occlusion management space is local and fixed [47] to the ray. This means that, while users can easily select a target when its location is revealed, it is difficult to identify occluded desired targets at unknown locations if some mechanism for visual search of the virtual space is not supported.

To achieve dynamic global filtering and local pointing with only one degree of control, we use a semi-transparent sphere (*Solar*) centered at the user's head.

*Solar*'s status depends on touch events, provided by the entire touchscreen of the connected device. When the touch area is held over  $T_{trigger}$  (*Solar* trigger time), *Solar* turns on and users observe the virtual environment through it. When fingers leave the touch surface, *Solar* turns off and users are able to observe objects with their original appearance.  $T_{trigger}$  is set to 300 ms, a value that reduces Heisenberg effect [6] and works well in practice.

With *Solar* on, a ray and a cursor are defined in relation to *Solar*'s volume. When a ray does not intersect with any object, the ray extends to the inner

surface of *Solar* (Fig. 3(2)). However, if the extension of the ray beyond *Solar*'s surface intersects with an object *outside* the sphere, the ray snaps to the intersected object (Fig. 3(3)). Furthermore, when *Solar* is activated, users can either rotate or scale the translucent sphere to interact with occluders. By integrating these motions with progress refinement [9, 20, 23] and a virtual x-ray [2, 17], we propose three modes to handle occlusion using *Solar*: Rotate, Hide and Semi-Transparent (SemiT). Users perform defined interactions as described in the following section to enter modes and the corresponding outcome is preserved even when *Solar* is off.

We can envision manipulating *Solar* in one of two ways, either by rotating it, sheering the display space such that hidden objects become visible; or by expanding and contracting it, i.e. by scaling *Solar* along its radius. As with flower-ray [20] and Expand [9], **Rotate** mode rearranges objects in the environment to minimize occlusion. With **Rotate** mode, objects that intersect with *Solar*'s surface can rotate around *Solar*'s center, similar to the Arcball Rotation [38] (Fig. 2). Therefore, when the environment is not fixed and a target is occluded by multiple objects, it is easier to point at the target after rotating the object away from the occluders.

Alongside **Rotate**, *Solar* can also be scaled, and objects inside *Solar* can be filtered using complete or partial transparency. We introduce **Hide** and **SemiT** modes, a virtual x-ray metaphor [2, 17, 28, 33]. This metaphor presents two alternative filters that permits users to see through objects between the user and the surface of the sphere, thus making the occluded (beyond the sphere's surface) objects visible. As their names suggest, Hide is used to hide occluders while SemiT turn occluders semi-transparent. We use a similar design as [28, 33] where the spatial relation between objects and the *Solar*'s surface defines objects' selectability. When Hide is on, objects intersecting with *Solar* or outside of it are visible and selectable while inside *Solar*, they are invisible and unselectable. When SemiT is on, objects inside *Solar* become semi-transparent and unselectable. Figure 2 shows these filter mechanisms.

### 3.2 Interaction Mapping

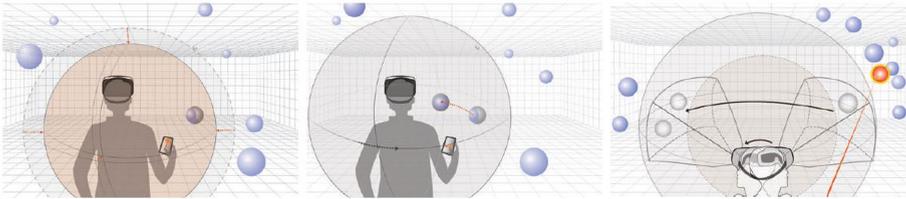
To support filtering with *Solar*, the user must be able to control rotation or scaling. To rotate objects intersecting with *Solar*, we map the horizontal movement of a finger on the touchscreen ( $\Delta_h$ ) with their rotation ( $\theta$ , in degrees) on *Solar* around its Yaw axis using the following function:  $\theta = f(v_{\Delta_h}) \cdot \Delta_h$ . Here,  $v_{\Delta_h}$  is the finger speed and  $f(v_{\Delta_h})$  is a transfer function for both movement mapping and gain control of rotation. When users want to point at a occluded object but not move its neighbour objects too far, they should move their fingers horizontally on the touch surface slowly. Therefore,  $f(v_{\Delta_h})$  is designed as a bounded linear interpolation depending on finger speed [4].

The scaling of *Solar* has a similar design, where the vertical movement of a finger on the touchscreen ( $\Delta_v$ ) maps to the radius of the *Solar* ( $R$ , in meters). When a finger moves up/down on the touchscreen, the radius increases/decreases correspondingly. We use the distance from the head position to the position of

the connected device when Solar is initialized, defined as  $R_o$ . Solar’s mapping function is, therefore, defined as:

$$R = \begin{cases} R_o & R \text{ is initialized or smaller than } R_o \\ R + g(V_{\Delta_v}) \cdot \Delta_v & \text{otherwise} \end{cases} \quad (1)$$

Similar to  $f(v_{\Delta_h})$ ,  $g(V_{\Delta_v})$  is also a bounded linear interpolation gain function. They both depend on the finger speed but not the cursor position, as gain functions on finger speed are more reliable and faster [4].

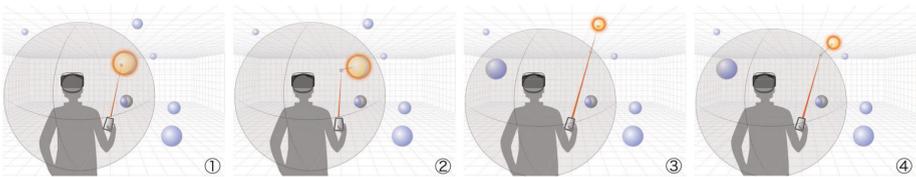


**Fig. 2.** Left: moving up/down on the touchscreen of a smartphone scales up/down Solar (Hide or SemiT). Mid: moving left/right on the touchscreen rotates objects intersecting with *Solar* left/right around *Solar*’s yaw axis (Rotate). Right: modes will only be applied to targets within *Solar* and a user’s fov.

### 3.3 Facilitation Techniques

To facilitate selection, *Solar* can include pointing facilitation. We include both a bubble cursor mechanism [19] and jitter filtering, as follows.

**Bubble Mechanism.** *Solar*-Casting’s bubble mechanism is implemented by selecting the nearest on- or outside-*Solar* targets by calculating their Euclidean distance from the point where the ray intersects the *Solar* to surrounding targets when the ray does not intersect with any object. The target selected by the bubble mechanism is highlighted with a different colour (Fig. 3 (2&4)).

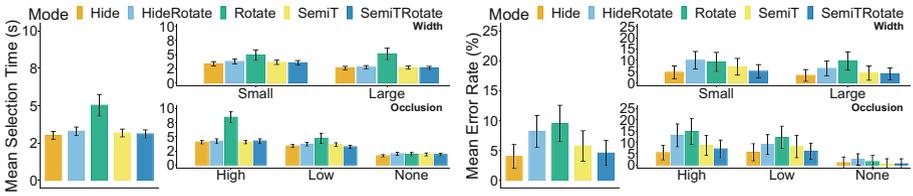


**Fig. 3.** Bubble mechanism in ray: (1) If a ray intersects with an object on *Solar*, the object is highlighted; (3) If a ray intersects with an object outside of *Solar*, the cursor snaps to the target and the ray gets extended; (2&4) If a ray does not intersect with any object, the closest target to the cursor will be highlighted.

**Ray Filtering.** Jitter is a common issue in Ray-Casting for 3D target acquisition, especially when pointing at small and distant targets, due to unintentional hand tremor and noisy sensing from input devices. Ray filtering smooths ray related movement during selection. In the Solar-Casting technique, we use the 1€filter [10] as it offers a good trade-off between precision and latency.

### 3.4 Preliminary Study on Solar-Casting Mode

We conducted informal preliminary tests with 5 adults for feedback on our three occlusion mitigation modes (Hide, SemiT, and Rotate) and their combinations using Solar-Casting (Fig. 4).



**Fig. 4.** Mean selection time (left) and error rate (right) results for the preliminary study, with 95% confidence intervals.

The results showed that, among all modes, Hide, SemiT and their combinations with Rotate had similar, good performance in terms of selection time, but Hide had a trend to lower error rate than all other modes. This is consistent with previous findings [2], suggesting that semi-transparency might compromise the task as spatial relationships between semi-transparent targets and unfiltered targets become unclear. However, for Hide, as context is removed, participants have more difficulty searching for targets. As P2 commented, *Hiding is bad for real UI, how will you know things are hidden?*

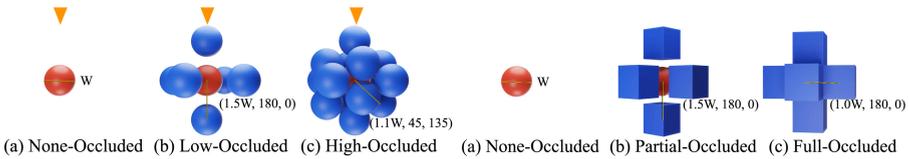
Rotate was slower than other modes because two steps were needed to handle occlusion. Participants first moved vertically on the touchscreen to scale *Solar* such that it intersected with occluders. Then, they moved horizontally to rotate occluders away. Participants reported that Rotate was difficult to use if a goal target was occluded not only by its neighbors but also by other distant targets that blocked the intersecting ray. Unlike Hide and SemiT, which provided straightforward visual feedback, sometimes it was hard for participants to identify intersecting objects on *Solar*'s surface.

Regarding mode combinations, some participants felt that they caused unexpected results: when they intended to use Hide or SemiT, Rotate was triggered by accident as they carelessly swiped horizontally, and objects intersecting with *Solar*'s surface were rotated away. Essentially, when in combination, Rotate was difficult to use and resulted in mode errors. Rotate alone was also slower than other visualizations.

## 4 General Experimental Setup

We designed and conducted two studies to evaluate Solar-Casting for selecting targets in 3D virtual environments, at different occlusion levels. Given the results of the pilot study, we eliminated *Solar* rotation and used SemiT in both studies, as Hide partially sacrifices context by suppressing part of the virtual environment (inside Solar’s sphere).

In Study 1, a visual indicator (orange triangle) was provided and a small portion of targets were noticeable even in the high occlusion environment (Fig. 5 (left)) to investigate how visual guidance influences selection in an occluded environment [31]. In Study 2, the visual indicator was removed and targets were fully blocked in the full-occlusion environment (Fig. 5 (right)). Aside from this significant difference, the two studies had a similar experimental design and protocol.



**Fig. 5.** Three occlusion levels in Study 1 (left) and Study 2 (right): a red sphere (target) is surrendered by blue occluders, which are generated around the target in a spherical coordinate system  $(r, \theta, \phi)$ ,  $r$  in meter,  $\theta, \phi$  are in degree. (Color figure online)

### 4.1 Apparatus

We used a Huawei VR Glass as the head-mounted display (HMD) in both studies. It has a resolution of  $3200 \text{ px} \times 1600 \text{ px}$ , with a pixel density of 1058 ppi. The display of this VR Glass is rendered by a connected smartphone, which conveniently provides both a touch surface and a Ray-Casting device for use in the VR environment. We note that smartphones are frequently explored as an input device for selection [11] and manipulation [51], and current VR controllers generally include a touchpad or joystick which can be used for controlling *Solar*.

### 4.2 Procedure

In both studies, participants were welcomed to the experiment room and sat in a swivel chair. They first read the study instructions and verbal consent was obtained. Before the study, they were first asked to answer a questionnaire on demographic information (gender, age) and daily and weekly usage of mobile and VR devices. Participants were warned about potential motion sickness induced from VR, and were allowed to have breaks between experimental blocks. If they felt uncomfortable at any time during the study, the study immediately stopped.

After sanitizing the VR glass and phone with alcohol wipes, participants wore the VR glass, and started a training session to practice each technique with 6 trials. They then entered the experimental scene and finished the study with the provided techniques. Techniques were counter-balanced using a Latin square [49]. When participants completed the study, they removed the VR glasses and completed a survey of 7-scale Likert items grading their experience.

## 5 User Study 1: With Visual Guidance

In the first study, our goal was to evaluate Solar-Casting’s performance in dense and occluded environments, where visual cues can help to identify objects.

We designed a comparative study including four interaction techniques: Solar-Casting, Expand [9], Depth Ray with a localized transparency function [47] and Depth-Casting, a technique similar to Tilt-Casting [33] or Slicing-Volume [28], which were chosen because of their characteristics (Table 1).

**Table 1.** Characteristics of the selected baseline techniques.

Technique	Interaction space	Strategy
Expand	Limited	Progressive refinement
Depth ray	Limited	Visual filtering (virtual X-ray)
Depth-Casting	Scaled	Visual filtering (virtual X-ray)

While Expand, Depth Ray and Tilt-Casting were all previously used in physical displays for occlusion management, and Slicing Volume used two controllers for volume control, we adapted them to a VR HMD and a phone-control environment with the following modification:

- Depth Ray: The cursor movement along the ray was controlled by the phone by moving the finger on the touchscreen and any target within 0.1 m to the cursor is rendered as semi-transparent while the closest one is highlighted via a bubble cursor mechanism [19].
- Expand: A semi-transparent sphere with radius 0.1 m, is rendered when a ray is intersecting with any target. When a finger is moving down on the touchscreen, i.e. a pull gesture, targets intersecting with the sphere are cloned, arranged into a grid, and presented in front of the user.
- Depth-Casting: The tilted plane is always displayed in front of the user vertically, which dynamically scales and occupies over 90% of the view frustum based on the depth. Targets in front the plane are rendered semi-transparent and only targets intersecting with the tilted plane are selectable. Second, tilting of the phone changes the depth of the plane instead of rotating the plane around its pitch axis [33, 46]; third, as the cursor is restricted to the tilted plane, cursor position is controlled by the head movement. We also implemented a relative mapping [39] on the phone to accommodate one-hand use and target positions which are hard to acquire with the head.

## 5.1 Experimental Design

Participants were instructed to perform a sequential pointing task as quickly and accurately as possible. A block design was adopted. In each block, given the technique, participants were asked to perform selection on 10 goal targets for each occlusion level (Fig. 5) and targets of all occlusion levels were generated around users at the same time (all were at least 0.6 m away from users).

Participants first pointed to a white sphere (dummy target) in order to fix the initial cursor position and start the timer for measuring the selection time of the sequential trials. Target width was 0.06 m and the distance  $D$  between two consecutive goal target was chosen from (0.4 m, 1.0 m, 1.6 m), so ID ranged from 2.94 to 4.79 bits [26]. To select a target, participants either had to position the cursor over it or move the cursor as close as possible, to allow selection with the bubble mechanism. When either hovering over a target or moving as close as possible to it, the target would be highlighted (in orange), indicating it was selectable. The pointing task moved to the next trial only when the target was correctly selected. If participants made an erroneous selection, a misclick was recorded. Once a goal target was correctly selected, this goal target became blue and the next goal target became red. There was only one goal target in red at a time and an orange indicator was shown above it to help participants quickly locate where they should point to. The connected phone would vibrate for 0.1 s and 1 s when participants completed a block and a technique respectively.

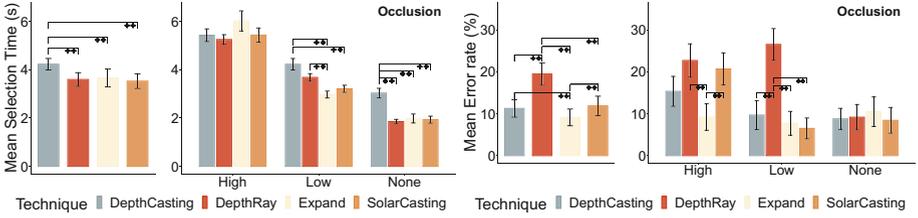
In summary, we designed a within-subjects experiment with independent variables: *Technique* (Depth Ray, Depth-Casting, Expand, Solar-Casting), *Occlusion* (None-Occluded, Low-Occluded, High-Occluded), and *Block* (1–4). Therefore, each participant needed to perform  $4 \text{ Technique} \times 3 \text{ Occlusion} \times 4 \text{ Block} \times 10 \text{ trials} = 480 \text{ trials}$ .

## 5.2 Participants

We recruited 12 participants from our organization, of which 11 reported their ages (22 to 26,  $\mu = 23.6$ ,  $\sigma = 1.3$ ). Among 12 participants, 1 was female, all were right-handed and 1 an experienced VR user. The study lasted about 60 min.

## 5.3 Results

We conducted a multi-way repeated-measure ANOVA ( $\alpha = 0.05$ ) for selection time (ST) and error rate ( $\%_{Err}$ ) respectively on three independent variables: *Technique*, *Occlusion* and *Block*. ST refers to the time elapsed between selections while  $\%_{Err}$  refers to the percentage of erroneous trials among 10 trials. Taking into account the non-normal distribution of ST and  $\%_{Err}$ , a Box-Cox transformation [8] and Aligned Rank Transform (ART) [14, 50] was applied to the data respectively. Greenhouse-Geisser corrections was applied to the DoFs when sphericity was violated using Mauchly's test. Pairwise tests with Bonferroni corrections were conducted when significant effects were found. Effect sizes were reported as partial eta squared ( $\eta_p^2$ ) values.

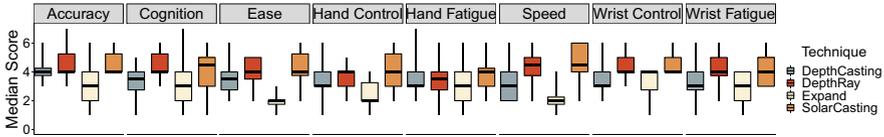


**Fig. 6.** Mean selection Time (left) and error Rate (right) for *Technique* and its interaction effect with *Occlusion*. Statistical significances are marked (+++ =  $p < 0.01$ ). Error bars represent 95% confidence intervals.

**Learning Effect.** With Box-Cox transformation ( $\lambda = -0.1$ ) and Aligned Rank Transform (ART) on ST and  $\%_{Err}$  respectively, we found only a significant effect of *Block* ( $F_{3,33} = 4.46$ ,  $p < 0.05$ ,  $\eta_p^2 = 0.03$ ) on ST. Pairwise t-test did not report any significance between blocks. Therefore, all 4 blocks were kept.

**Selection Time.** We found a significant effect of *Technique* on ST ( $F_{3,33} = 14.83$ ,  $p = 0.001$ ,  $\eta_p^2 = 0.28$ ). As shown in Fig. 6 (left), the pairwise t-test showed that Depth-Casting (mean = 4.23 s) took significantly longer ( $p < 0.001$ ) than other techniques: Solar-Casting (3.53 s), Depth Ray (3.60 s) and Expand (3.66 s). We also found a significant effect of *Occlusion* on ST ( $F_{2,22} = 1312.59$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.85$ ). It took participants significantly longer ( $p < 0.001$ ) selecting targets in high-occlusion environments (5.53 s) than in low-occlusion environments (3.52 s) and non-occluded environments (2.20 s). We also found a significant interaction effect of *Technique*  $\times$  *Occlusion* ( $F_{6,66} = 40.11$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.28$ ) on ST. In non- and low-occlusion environments, Solar-Casting (1.93 s & 3.21 s respectively) was significantly faster ( $p < 0.001$ ) than Depth-Casting (3.03 s & 4.23 s), but in high-occlusion environments, they had similar performance.

**Error Rate.** We found a significant effect of *Technique* on  $\%_{Err}$  ( $F_{3,553} = 33.03$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.15$ ). The pairwise comparison showed that Solar-Casting (11.87%), Depth-Casting (11.26%) and Expand (9.12%) caused significantly less erroneous selection than Depth Ray (19.45%,  $p < 0.001$ ). Meanwhile, Expand caused significantly ( $p < 0.005$ ) less erroneous selection than Solar-Casting and Depth-Casting. We found a significant effect of *Occlusion* on  $\%_{Err}$  ( $F_{3,553} = 27.71$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.09$ ). Obviously, participants made significantly more erroneous selection ( $p < 0.001$ ) in the high-occlusion environment (16.97%) than in the non- and low-occlusion environments (9.20% & 12.60% respectively). We also found a significant interaction effect between *Technique* and *Occlusion* ( $F_{3,553} = 15.23$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.14$ ). Looking at Fig. 6 (right), in both non- and low-occlusion environments, Solar-Casting (8.42% & 6.50%) and Depth-Casting (8.77% & 9.68%) had relatively low error rates, and in the high-occlusion environments, the error rate increased (Solar-Casting: 20.67% and Depth-Casting: 15.32%); while Expand kept relatively low-error rate across three levels of occlusion environments (Fig. 7).



**Fig. 7.** The median score and lower/upper quartiles are visualized for each measure. Lower scores indicate worse performance.

**Preference.** Friedman tests reported a significant effect of *Technique* on all attributes except for *Hand Fatigue*:  $\chi^2_{Accuracy}(4) = 15.13$ ,  $p < 0.005$ ,  $\chi^2_{Cognition}(4) = 8.70$ ,  $p < 0.05$ ,  $\chi^2_{Ease}(4) = 21.11$ ,  $p < 0.001$ ,  $\chi^2_{HandControl}(4) = 8.50$ ,  $p < 0.05$ ,  $\chi^2_{Speed}(4) = 21.28$ ,  $p < 0.001$ ,  $\chi^2_{WristControl}(4) = 12.79$ ,  $p < 0.01$ ,  $\chi^2_{WristFatigue}(4) = 13.02$ ,  $p < 0.001$ . The pairwise Wilcoxon test showed that participants felt Solar-Casting significantly easier and faster to use ( $p < 0.05$ ) and had better accuracy, hand and wrist control ( $p < 0.05$ ) than Expand. Regarding user feedback, P1 commented that it would be better if Solar’s sphere could reset after each successful selection. P1, P4, and P8-9 mentioned that Expand did not work well with high occlusion scenes, as it was difficult to intersect the desired region where the target object was. P2, and P10 complained about the wrist fatigue in Depth-Casting, as users needed to precisely control the plane tilt using their wrist. P1-P3, and P6-P10 mentioned that it was difficult to move the cursor in Depth Ray (P3: *It would be nice to filter all objects along the ray*). In the non-occluded condition, participants indicated a preference for selecting the target directly rather than moving the cursor along the ray.

**Discussion.** Solar-Casting was at least as fast and accurate as the other three techniques. However, note that the target indicator was the dominant factor in selection performance, which drastically decreased the visual search time in the task. Users were given the visual cue of the low- and high-occluded targets and they only needed to reveal and select them. Therefore Solar-Casting’s visual search benefits were not represented in the above analysis. As expected, occlusion level had a significant impact on selection time. All four techniques performed more slowly in high-occlusion environments. Although users were informed of the target’s location by the target indicator, it still took longer for them to reveal the occluded target for subsequent selection. In terms of error rate, Expand had a consistently low error rate through different occlusion levels. This is understandable given the characteristics of progressive refinement, considering that the visual representation of objects during selection is consistent in Expand. Meanwhile for the other three techniques, occlusion had a more significant impact on error rate.

## 6 User Study 2: Without Visual Guidance

Different from the former study, in the second study, the visual cue was removed and a new occlusion design (Fig. 5 (right)) was adopted so that participants

needed to use the provided technique to explore the scene and point at targets across different occlusion levels.

In our pilots, in the full-occlusion environment, Expand was extremely time-consuming and demanded significant effort, as participants needed to try every cluster with a pull gesture on the touchscreen. Without a visual cue, users felt dizzy quickly rotating their head and turning around to search for occluded targets. Therefore, Expand was eliminated from User Study 2 and targets were all presented in front of the participants. Since targets were all presented in front of participants, the front cube might block the user’s view, therefore, for the partial-occlusion environment, we slightly rotated cubes round the goal target to a random degree ( $0^\circ - 45^\circ$ ) to ensure that at least a small portion of a target was visible to participants.

## 6.1 Experimental Design

Similar to Study 1, a within-subjects experiment was designed with independent variables: *Technique* (Depth Ray, Depth-Casting, Solar-Casting), *Occluded* (None-Occluded, Partial-Occluded, Full-Occluded), and *Block* (1–4). As targets were all presented in front the user (at least 0.6 m away from users), the distance  $D$  between two consecutive goal target was chosen from 0.3 m, 0.6 m, 0.9 m, so ID ranged from 2.58 to 4.00 bits given that target width was still 0.06 m. Given the difficulty of the task, the number of trails per condition was reduced to 8. Therefore, each participant performed  $3 \text{ Technique} \times 3 \text{ Occlusion} \times 4 \text{ Block} \times 8 \text{ trials} = 288 \text{ trials}$ .

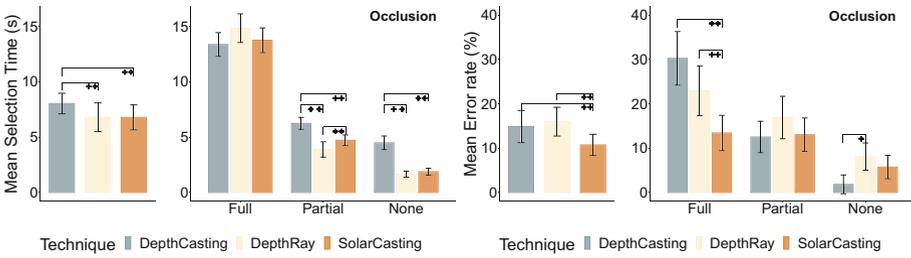
## 6.2 Participants

We recruited 12 participants, exclusive from User Study 1, from our organization, aged from 21 to 31 ( $\mu = 25.5$ ,  $\sigma = 3.1$ ), of which 4 were female, all were right-handed and 2 were experienced VR users. The study lasted about 90 min.

## 6.3 Results

**Learning Effect.** With Box-Cox transformation ( $\lambda = -0.1$ ), a repeated measures ANOVA found a significant effect of *Block* ( $F_{3,33} = 2.79$ ,  $p = 0.04$ ,  $\eta_p^2 = 0.05$ ) on ST. However, pairwise t-test did not report any significance between blocks. Therefore, all 4 blocks were kept.

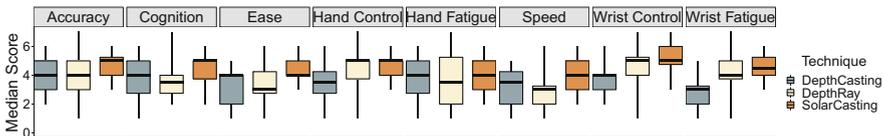
**Selection Time.** We found a significant effect of *Technique* on ST ( $F_{2,22} = 23.13$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.39$ ). As shown in Fig. 8 (left), the pairwise t-test showed that Depth-Casting (mean = 8.02 s) took significantly longer ( $p < 0.001$ ) than other techniques: Solar-Casting (6.77 s), Depth Ray (6.78 s). We found a significant effect of *Occlusion* on ST ( $F_{2,22} = 593.98$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.90$ ). The pairwise t-test showed that participants spent significantly longer time in a partial-occluded environment (4.94 s) than in a none-occluded environment



**Fig. 8.** Mean selection Time (left) and error Rate (right) for *Technique* and its interaction effect with *Occlusion*. Statistical significances are marked (++ =  $p < 0.01$ , + =  $p < 0.05$ ). Error bars represent 95% confidence intervals.

(2.67 s,  $p < 0.001$ ), yet significantly shorter time than in a full-occlusion environment (13.96 s,  $p < 0.001$ ). We also found a significant interaction effect between *Occlusion* and *Technique* on ST ( $F_{4,44} = 57.00$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.38$ ). In both non- and partial-occluded environments, Depth-Casting (4.48 s & 6.23 s respectively) was significantly slower ( $p < 0.001$ ) than Solar-Casting (1.88 s & 4.70 s respectively) and Depth Ray (1.65 s & 3.88 s respectively) while in the full-occlusion environment, there was no significant difference though Depth Ray (14.81 s) was relatively slower than Depth-Casting (13.34 s) and Solar-Casting (13.72 s).

**Error Rate.** We found a significant effect of *Technique* on  $\%_{Err}$  ( $F_{2,412} = 9.19$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.04$ ). The pairwise comparison showed that Solar-Casting (10.72%) was significantly more accurate than Depth-Casting (14.85%) and Depth Ray (15.96%). We also found a significant effect of *Occlusion* on  $\%_{Err}$  ( $F_{2,412} = 62.77$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.23$ ). Obviously, error rate increased while the occlusion level increased. Participants performed significantly more accurate selection in the non-occluded environment (5.21%) than in the partially- (14.14%) and fully-occluded (22.18%) environments. We also found a significant interaction effect of *Technique* and *Occlusion* on  $\%_{Err}$  ( $F_{4,412} = 10.83$ ,  $p < 0.001$ ,  $\eta_p^2 = 0.10$ ). While Depth-Casting and Depth Ray made significantly more erroneous selections in the full-occlusion environment than in the partial-occlusion environment, Solar-Casting still achieved high accuracy in both environments (fully-occluded: 13.41%, partially-occluded: 13.02%).



**Fig. 9.** The median score and lower/upper quartiles are visualized for each measure. Lower scores indicate worse performance.

**Preference.** Friedman tests reported a significant effect of *Technique* on Wrist Control and Fatigue:  $\chi^2_{WristControl}(4) = 14.00$ ,  $p < 0.001$ ,  $\chi^2_{WristFatigue}(4) = 12.19$ ,  $p < 0.01$ . The pairwise Wilcoxon test reported that DepthCasting demanded significantly ( $p < 0.05$ ) more wrist control and caused significantly ( $p < 0.05$ ) more wrist fatigue than Solar-Casting. Figure 9 also showed that Solar-Casting had relatively high scores across all attributes.

Given the local transparency function of Depth Ray, some participants reported that it was difficult to see where the goal target was in the full-occlusion environment as targets outside the transparency volume and in front of the goal target in depth might block the view. To handle this issue, some participants (P1,P3-P6,P9) believed that a global filtering mechanism along the ray would be much welcome in Depth Ray, instead of on the cursor. As Solar-Casting supported global filtering, P5 and P7 felt Solar-Casting needed careful manipulation, (P5: *I like Solar-Casting, but I found Depth-Casting very fast at searching targets*). Similarly, P3 commented that the tilting mechanism in Depth-Casting was better than finger swiping in Solar-Casting or Depth Ray, yet P6 prefer swiping over tilting.

**Discussion.** In this study, we noted that, across all occlusion levels, Solar-Casting outperformed Depth-Casting in terms of speed, and outperformed both Depth-Casting and Depth Ray in terms of error rate. While Solar-Casting’s and Depth Ray’s speed was near identical in our experiment – their speeds differ by less than 2%, not significant – Solar-Casting’s lower overall error rate argued for the effectiveness of SolarCasting for elegantly supporting selection regardless of level of occlusion.

Looking, specifically, at levels of occlusion, we saw that Solar-Casting’s significant accuracy advantage was particularly acute for the full occlusion condition, and that Depth Ray was consistently less accurate. Further, we saw that, regardless of occlusion level, Depth-Casting was significantly slower than both Solar-Casting and Depth Ray. Solar-Casting’s and Depth Ray’s speed parity holds across occlusion levels (see Fig. 8).

## 7 Overall Discussion

As we note in our second study, our final Solar-Casting design supports high accuracy, particularly for fully-occluded contexts, and its speed is at least as good as competing techniques regardless of occlusion levels. Together, these results demonstrate that Solar-Casting achieved our design goals.

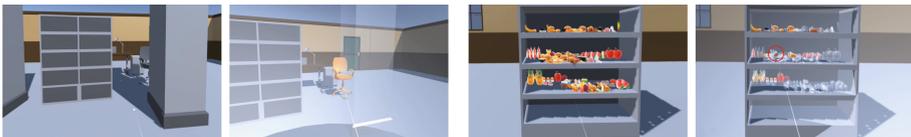
The target layouts, Fig. 5, were initially inspired by [47], and serve to create occluding distractors around targets. We noted that occlusion in VR could happen with/without visual guidance, and within/outside the fov; therefore, our designs in both studies were, essentially, a generalized version of [52]. This balance between preserving aspects of prior study design while seeking more representative tasks is challenging, and does represent a potential limitation.

While Solar-Casting appears highly effective for supporting selection despite occlusion, there are ways that Solar-Casting could be enhanced, particularly with respect to control of the Solar’s volume. Compared with techniques using a spherical volume [9,23] at the end of a ray, Solar-Casting required increased effort to point at occluded objects at a large distance because users needed to perform repeated scaling actions, i.e. *clutching*, to increase the volume of *Solar* and filter occluders. We continue to explore clutching-mitigation strategies, including tuned CD gain functions for controlling Solar’s radius, which may reduce clutching and further speed performance.

Looking specifically at Depth Ray and Solar-Casting, we discovered that compared with local filtering, global filtering greatly increased the accuracy but not speed of selection. During the search-point process, participants’ attention was limited regardless of the filtering range, while this range controlled the presence of occluders. Meanwhile, comparing results in two studies, we observed that visual guidance not only greatly increased the accuracy but also the speed of selection. This is understandable as visual guidance is a strong stimulus to improve participants’ attention. Therefore, future work also includes investigating approaches to improve users’ attention with Solar-Casting.

We used the Huawei VR glass, a system in which the connected phone serves as both the computing and input device. One advantage of this setup is simpler configuration (the smartphone is “at hand”) and our smartphone-based implementation of Depth Ray and Expand is useful as it demonstrates their performance on smartphones (as opposed to a dedicated controller). However, Solar-Casting is independent of the input device or HMD. It can be generalized using any other headsets and input devices with suitable mapping on *Solar*’s scaling and rotation parameters. Exploring optimal mappings for varied hardware controllers is another avenue of future work.

To highlight Solar-Casting’s benefits in virtual environments with occlusion, we conclude by demonstrating Solar-Casting’s use in two scenarios to manage occlusion. In Fig. 10 (a), we created an interior design example, where SemiT is applied to the environment globally such that pillars become translucent so users can easily point at the partially occluded chair. Similarly, in a fully-occluded environment, like a refrigerator in Fig. 10 (b), users can gradually scale *Solar*, filter unwanted food items and then point at the occluded bowl.



(a) Pointing at a partially occluded chair

(b) Acquiring a fully occluded bowl

**Fig. 10.** Two typical scenarios in virtual environment: Solar-Casting can easily be applied in either (a) a partial-occlusion environments or (b) full-occlusion environments to achieve fast and stable target acquisition.

## 8 Conclusion

In this paper, we present Solar-Casting, a dynamic pointing technique for global filtering and selection in occluded virtual environments. Solar-Casting uses a scalable, semi-transparent sphere as the reference for object manipulation and filtering to address occlusion. In a pilot study evaluating three filtering modes – Rotate, Hide and SemiT – we find that Hide and SemiT have better performance, and SemiT improves users’ awareness of the environment. We then evaluate Solar-Casting’s performance in two formal experiments. In the first, we compare Solar-Casting with several techniques with a visual cue pointing to occluded targets. We find that Solar-Casting has competitive performance. However, the advantage of Solar-Casting is that it supports search and selection concurrently, even when target locations are not known *a priori*. To highlight Solar-Casting’s search advantages, in a second experiment we evaluate Solar-Casting without visual guidance. We find that Solar-Casting achieves significantly more accurate selection without any degradation in speed, regardless of occlusion level. Overall, these findings demonstrate that Solar-Casting is an effective tool to support search and target acquisition in cluttered virtual environments.

**Acknowledgements.** We would like to thank Roger Luo and Shino Che Yan for their help in creating Figs. 1–3, Chaoran Chen for his help in organizing and conducting studies remotely, all participants for their help in this difficult time, and reviewers for their valuable feedback. This research received ethics clearance from the Office of Research Ethics, University of Waterloo. This research was funded by a grant from Waterloo-Huawei Joint Innovation Laboratory.

## References

1. Argelaguet, F., Andujar, C.: Efficient 3d pointing selection in cluttered virtual environments. *IEEE Comput. Graph. Appl.* **29**(6), 34–43 (2009)
2. Argelaguet, F., Andujar, C.: A survey of 3d object selection techniques for virtual environments. *Comput. Graph.* **37**(3), 121–136 (2013)
3. Bacim, F., Kopper, R., Bowman, D.A.: Design and evaluation of 3d selection techniques based on progressive refinement. *Int. J. Hum. Comput. Stud.* **71**(7–8), 785–802 (2013)
4. Baloup, M., Pietrzak, T., Casiez, G.: 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)
5. Bowman, D., Kruijff, E., LaViola, J.J., Jr., Poupyrev, I.P.: *3D User Interfaces: Theory and Practice*. Addison-Wesley, CourseSmart eTextbook (2004)
6. Bowman, D., Wingrave, C., Campbell, J., Ly, V.: Using pinch gloves (tm) for both natural and abstract interaction techniques in virtual environments (2001)
7. Bowman, D.A., Hodges, L.F.: 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)
8. Box, G.E.P., Cox, D.R.: An analysis of transformations. *J. Roy. Stat. Soc. Ser. B (Methodol.)* **26**(2), 211–252 (1964)

9. Cashion, J., Wingrave, C., LaViola, J.J., Jr.: Dense and dynamic 3d selection for game-based virtual environments. *IEEE Trans. Vis. Comput. Graph.* **18**(4), 634–642 (2012)
10. Casiez, G., Roussel, N., Vogel, D.: 1€filter: a simple speed-based low-pass filter for noisy input in interactive systems. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2527–2530 (2012)
11. Chen, Y., Katsuragawa, K., Lank, E.: Understanding viewport-and world-based pointing with everyday smart devices in immersive augmented reality. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13 (2020)
12. Chen, Y., Sun, J., Xu, Q., Lank, E., Irani, P., Li, W.: Empirical evaluation of moving target selection in virtual reality using egocentric metaphors. In: *IFIP Conference on Human-Computer Interaction*. Springer (2021)
13. Coffin, C., Hollerer, T.: Interactive perspective cut-away views for general 3d scenes. In: *3D User Interfaces (3DUI 2006)*, pp. 25–28. IEEE (2006)
14. Elkin, L.A., Kay, M., Higgins, J.J., Wobbrock, J.O.: An aligned rank transform procedure for multifactor contrast tests (2021)
15. Elmqvist, N.: BalloonProbe: reducing occlusion in 3d using interactive space distortion. In: *Proceedings of the ACM symposium on Virtual Reality Software and Technology*, pp. 134–137 (2005)
16. Elmqvist, N., Assarsson, U., Tsigas, P.: Employing dynamic transparency for 3d occlusion management: design issues and evaluation. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) *INTERACT 2007*. LNCS, vol. 4662, pp. 532–545. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-74796-3\\_54](https://doi.org/10.1007/978-3-540-74796-3_54)
17. Elmqvist, N., Tsigas, P.: A taxonomy of 3d occlusion management for visualization. *IEEE Trans. Vis. Comput. Graph.* **14**(5), 1095–1109 (2008)
18. Feiner, A.O.S.: The flexible pointer: an interaction technique for selection in augmented and virtual reality. *Proc. UIST.* **3**, 81–82 (2003)
19. Grossman, T., Balakrishnan, R.: The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor’s activation area. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 281–290 (2005)
20. Grossman, T., Balakrishnan, R.: 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)
21. Katzakis, N., Kiyokawa, K., Takemura, H.: Plane-casting: 3d cursor control with a smartphone. In: *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, pp. 199–200 (2013)
22. Katzakis, N., Teather, R.J., Kiyokawa, K., Takemura, H.: INSPECT: extending plane-casting for 6-DOF control. *HCIS* **5**(1), 22 (2015)
23. Kopper, R., Bacim, F., Bowman, D.A.: Rapid and accurate 3d selection by progressive refinement. In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 67–74. IEEE (2011)
24. Liang, J., Green, M.: Geometric modeling using six degrees of freedom input devices. In: *3rd Int’l Conference on CAD and Computer Graphics*, pp. 217–222. Citeseer (1993)
25. Lu, Y., Yu, C., Shi, Y.: 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)*. IEEE (2020)

26. MacKenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.* **7**(1), 91–139 (1992). <https://doi.org/10.1207/s15327051hci0701.3>
27. Mine, M.R.: Virtual environment interaction techniques. UNC Chapel Hill CS Dept. (1995)
28. Montano, R., Nguyen, C., Kazi, R.H., Subramanian, S., DiVerdi, S., Martinez-Plasencia, D.: Slicing volume: hybrid 3d/2d multi target selection technique for dense virtual environments. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE (2020)
29. Moore, A., Kodeih, M., Singhania, A., Wu, A., Bashir, T., McMahan, R.: The importance of intersection disambiguation for virtual hand techniques. In: 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 310–317. IEEE (2019)
30. Mossel, A., Koessler, C.: Large scale cut plane: An occlusion management technique for immersive dense 3d reconstructions. In: Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, pp. 201–210 (2016)
31. Ouramdane, N., Otmane, S., Davesne, F., Malle, M.: FOLLOW-ME: a new 3d interaction technique based on virtual guides and granularity of interaction. In: Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications, p. 137–144. VRCIA 2006, Association for Computing Machinery, New York, NY, USA (2006). <https://doi.org/10.1145/1128923.1128945>
32. Pierce, J.S., Forsberg, A.S., Conway, M.J., Hong, S., Zeleznik, R.C., Mine, M.R.: Image plane interaction techniques in 3d immersive environments. In: Proceedings of the 1997 Symposium on Interactive 3D Graphics, pp. 39–ff (1997)
33. Pietroszek, K., Wallace, J.R., Lank, E.: Tiltcasting: 3D interaction on large displays using a mobile device. In: Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology, pp. 57–62 (2015)
34. Pindat, C., Pietriga, E., Chapuis, O., Puech, C.: Drilling into complex 3D models with gimlenses. In: Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology, p. 223–230. VRST 2013, Association for Computing Machinery, New York, NY, USA (2013). <https://doi.org/10.1145/2503713.2503714>
35. Poupyrev, I., Billingham, M., Weghorst, S., Ichikawa, T.: 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)
36. Poupyrev, I., Ichikawa, T., Weghorst, S., Billingham, M.: Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In: *Computer Graphics Forum*, vol. 17, pp. 41–52. Wiley Online Library (1998)
37. Ro, H., et al.: A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2873–2878. IEEE (2017)
38. Shoemaker, K.: ARCBALL: a user interface for specifying three-dimensional orientation using a mouse. *Graph. Interface* **92**, 151–156 (1992)
39. Siddhpuria, S., Malacria, S., Nancel, M., Lank, E.: Pointing at a distance with everyday smart devices. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–11 (2018)
40. Sidenmark, L., Clarke, C., Zhang, X., Phu, J., Gellersen, H.: Outline pursuits: gaze-assisted selection of occluded objects in virtual reality (2020)
41. Sun, C.: 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, pp. 729–740 (2013)

42. Sun, J., Stuerzlinger, W.: Selecting invisible objects. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 697–698. IEEE (2018)
43. Sun, J., Stuerzlinger, W.: Extended sliding in virtual reality. In: 25th ACM Symposium on Virtual Reality Software and Technology, pp. 1–5 (2019)
44. Sun, J., Stuerzlinger, W.: Selecting and sliding hidden objects in 3D desktop environments. In: Proceedings of the 45th Graphics Interface Conference on Proceedings of Graphics Interface 2019, pp. 1–8. Canadian Human-Computer Communications Society (2019)
45. Sun, J., Stuerzlinger, W., Shuralyov, D.: Shift-sliding and depth-pop for 3d positioning. In: Proceedings of the 2016 Symposium on Spatial User Interaction, pp. 69–78 (2016)
46. Sun, M., Cao, M., Wang, L., Qian, Q.: PhoneCursor: improving 3d selection performance with mobile device in AR. *IEEE Access* **8**, 70616–70626 (2020)
47. Vanacken, L., Grossman, T., Coninx, K.: Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In: 2007 IEEE Symposium on 3D User Interfaces. IEEE (2007)
48. Wacker, P., Nowak, O., Voelker, S., Borchers, J.: Arpen: Mid-air object manipulation techniques for a bimanual AR system with pen and smartphone. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–12 (2019)
49. Williams, E.: Experimental designs balanced for the estimation of residual effects of treatments (1949). <https://doi.org/10.1071/CH9490149>
50. Wobbrock, J.O., Findlater, L., Gergle, D., Higgins, J.J.: The aligned rank transform for nonparametric factorial analyses using only anova procedures. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 143–146 (2011)
51. Wu, S., Chellali, A., Otmane, S., Moreau, G.: TouchSketch: a touch-based interface for 3d object manipulation and editing. In: Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology. p. 59–68. VRST 2015, Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2821592.2821606>
52. Yu, D., Zhou, Q., Newn, J., Dingler, T., Velloso, E., Goncalves, J.: Fully-occluded target selection in virtual reality. *IEEE Trans. Vis. Comput. Graph.* **26**(12), 3402–3413 (2020). <https://doi.org/10.1109/TVCG.2020.3023606>