Disocclusion Headlight for Selection Assistance in VR

Lili Wang *
Beihang University
Peng Cheng Laboratory

Jianjun Chen
Beihang University

Qixiang Ma
Beihang University

Voicu Popescu
Purdue University

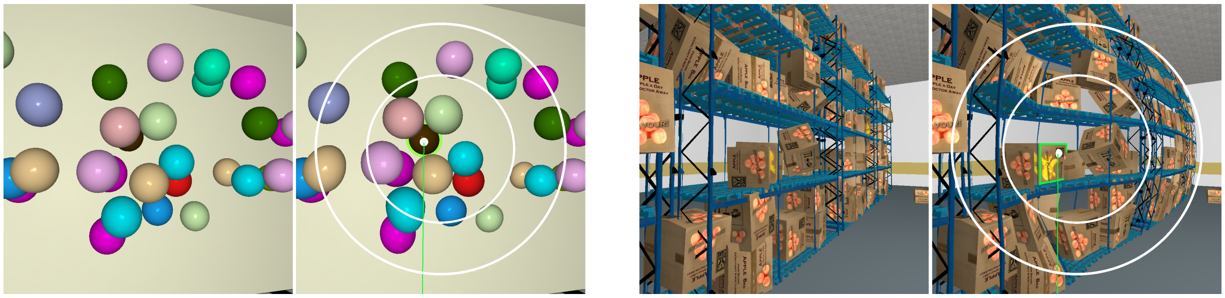

Figure 1: Two pairs of conventional (left of pair) and disocclusion headlight (right of pair) frames. The disocclusion headlight adapts the viewpoint at the center of the frame to maximize object visibility, which facilitates the selection of the brown sphere and of the banana box. The periphery of the disocclusion headlight frame is rendered conventionally, from the user viewpoint, which anchors the user; the transition from the periphery to the center is continuous.

2021 IEEE Virtual Reality and 3D User Interfaces (VR) | 978-1-6654-1838-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/VR50410.2021.00043

Abstract

We introduce the *disocclusion headlight*, a method for VR selection assistance based on alleviating occlusions at the center of the user's field of view. The user's visualization of the VE is modified to reduce overlap between objects. This way, selection candidate objects have larger image footprints, which facilitates selection. The modification is confined to the center of the frame, with continuity to the periphery of the frame which is rendered conventionally. The selection assistance is provided automatically, without any interaction from the user. Furthermore, our method disoccludes without destroying the local spatial relationships between selection candidates, which allows solving complex selection queries based on the relative position of objects. We have tested our method on three selection tasks, where we compared it to two state-of-the-art VR selection techniques, i.e., the alpha cursor and the flower cone. Our method showed significant advantages in terms of shorter task completion times, and of fewer selection errors.

Index Terms: Virtual reality—Pointing and selection—Disocclusion—Multiperspective rendering;

1 Introduction

Object selection is one of the fundamental tasks that a 3D user interface has to enable. In virtual reality (VR), object selection is often the first step in the user's interaction with the virtual environment (VE). Selection in VR is a complicated task, and poorly designed selection techniques often have a significant negative impact on the overall user performance. First, the user hands and fingers are not tracked with high fidelity; from the tens of degrees of freedom of the human hand, a typical VR application only tracks the user's fist. Second, most VR applications do not convey haptics, so grabbing an object lacks the guidance of the feel of the real objects provides. Third, the user cannot always easily change position in the VE to get close to the object they would like to select. Due to these challenges, VR selection methods focus on allowing the user to pick an object from afar. However, selection from a distance has its own challenges. First, accurate selection of distant objects require fine grain hand motions, as, for example, a small angular change of the orientation of a virtual laser pointer can imply a large translation of the laser dot, which can inadvertently jump from one object to another. Second, in complex VEs, occlusions can partially or completely hide the object to be selected.

In this paper we propose the *disocclusion headlight*, a method for VR selection assistance based on alleviating occlusions at the center of the user's field of view. The disocclusion headlight modifies the user's visualization of the VE to avoid that two or more objects be on the same user line of sight. This gives the selection candidate objects disjoint user image footprints, alleviating occlusions, and facilitating selection. The modified, multiperspective visualization is confined to the center of the user image, and it is connected seamlessly to the periphery of the frame, which is rendered conventionally. Fig. 1 shows the selection assistance provided by our method. Whereas in the conventional view the selection target is barely visible due to occlusions, the disocclusion headlight visualization maximizes the visible footprint of all objects at the center of the user view, including that of the selection target. At its essence, recognizing the fact that selection is easy to do in 2D when the visualization is unambiguous, our method optimizes the projection of the VE onto the user's view, obtaining a visualization suitable for selection.

We have tested our selection assistance method in a user study with three tasks. In the first task the user had to select spheres of a given color (Fig. 1, left pair). In the second task the user had to select the two spheres closest to each of several cubes. In the third task the user had to select cardboard boxes with specific labels in a virtual warehouse (Fig. 1, right pair). Our method was compared to two state of the art selection techniques: the alpha cursor and the flower cone [42]. Our method showed significant advantages in terms of fewer selection errors, and of shorter task completion times. Unlike a method such as the alpha cursor, our method provides selection assistance automatically, in the sense that it disoccludes without any interaction from the user. Unlike a method like the flower cone, our method disoccludes without destroying the local spatial relationships between selection candidates, which allows the user to solve more complex selection queries, such as those based on the relative position of multiple objects.

*e-mail: wanglily@buaa.edu.cn

2642-5254/21/$31.00 ©2021 IEEE
DOI 10.1109/VR50410.2021.00043

2 Related Work

Selection in VR is an important task as it is prerequisite to most interactions with the VE. Selection in VR is challenging, and many methods have been developed [3, 37, 27, 17, 24, 2, 4, 32].

Some selection methods let the user aim a ray at the selection target. Inspired by real world laser pointers, one approach is for the user to aim the selection ray with their hand with six degrees of freedom [25]. There are some other ways to aim the selection ray. For example, the ray can be defined with the help of two hands [1], of the view direction [12], of the eye position [8, 7, 5], or of the nose position [33]. In the quest to better control the ray, one approach uses the head position as the ray starting point and the ray direction is defined with a second point confined to a plane in front of the user [28]. The improved control over the selection ray comes at a cost of an increased interface complexity. Another approach is to use multiple rays in tandem to complete the selection task [41, 7, 23].Whereas multiple rays aid with selection target disambiguation and support selection in parallel, the approach brings added interface and tracking complexity.

Selection is challenging in cluttered scenes with many small objects and complex occlusion patters. One approach for assisting with selection in cluttered scenes is to rely on heuristics to guess the intended selection target [11]. Another approach is to control the cursor that slides up and down the ray to indicate which of the multiple objects intersected by the ray should actually be selected [16]. The ray-cursor paradigm can be enhanced with ray stabilization algorithms that reduce the selection error rate caused by unwanted input device jitter [6]. The challenge is that selection proceeds in two steps, i.e., aiming the ray followed by sliding the cursor, which reduces selection efficiency. Another enhancement is to replace the cursor with a bubble, whose radius provides an additional degree of freedom that can be set based on Voronoi diagrams to further disambiguate the selection target [15, 34, 22].

Another approach for selection in cluttered scenes is to expand the selection ray to a selection volume [14, 25], such as a cone, a frustum, a cube, or a sphere. The multiple selection candidates picked up by the volumetric selection tool require disambiguation mechanism [34]. One option is to rely on the user to narrow down the selection in a second step [21, 9, 38]. Another option is to use heuristics that sort the objects based on their position, orientation, and visibility [11, 31, 30].

Several selection assistance methods target specifically the problem of occlusions, which hides most if not all of the selection target [13], as does our disocclusion headlight method. The traditional occlusion management in VR is to ask the user to translate the viewpoint to establish line of sight to the selection target. However, this can be inefficient or even impractical in large or complex VE's. One option for letting the user overcome occlusions from their current position is to abandon the rigidity of the selection ray for a curved ray that can bend to reach around occluders [1]. Since the method does not remove the occlusion from the image, the user might have to select blindly, behind the occluder. Another method for handling occlusions is to provide the user the ability to cut away occluding VE geometry, for example with an alpha cursor [42]. The user slides the cursor to define the radius of a clipping sphere centered at the user position, which disoccludes the selection target.

A different idea is to rearrange the objects in a way that alleviates or removes occlusions. For example, the user can be given a selection sphere and then the multiple objects inside the sphere are shown to the user in a four-way hierarchical menu from which the user selects the target process[21]. Subsequent work replaced the four-way menu with displaying all selection objects simultaneously over the entire image [9]. The flower cone approach [42] uses a cone as the volumetric pre-selection tool, and the candidates are selected with a virtual laser pointer from a matrix pattern. In addition to the less efficient two step approach, rearrangement also brings the challenge of breaking the local spatial relationships between selection candidates, which might be essential for disambiguation.

Like a depth cursor method, our disocclusion headlight method circumvents occluders to reveal the target, but it does so automatically, without user intervention; like a rearrangement method, our method modifies the user visualization of the VE to obtain better separation between the selection targets, but it does so in situ, to preserve the local spatial relationships between the selection targets.

Our method relies on multiperspective visualization to manage occlusions. Our multiperpsective visualization is constructed with a graph camera [26], which is a flexible compound camera model constructed recursively from a root frustum that undergoes a series of bending, splitting, and merging operations. The graph camera has previously been extended to achieve a multiperspective focus+context effect [39]. The benefits of multiperspective visualization have been previously noted in the VR/AR context where it improves navigation efficiency [40, 36], and collaboration [35].

We compare our method to two state-of-the-art VR selection methods: the alpha cursor [42] and the flower cone [42]. We chose these two techniques because they are representative for two important, frequently used selection facilitation approaches, i.e., complexity reduction by elimination and by spatial layout change.

3 Disocclusion Headlight

We have developed the *disocclusion headlight*, a VR selection technique designed to address the following concerns. (1) The overall goal is to improve the user's view of the VE such that the image footprint of objects is maximized by alleviating occlusions. Such an image will serve as a 2D map which helps the user select the desired object with a virtual laser pointer more easily, as the object's larger image footprint increases the range of pointing angles adequate for selection. (2) The method should be automatic–the method should not add degrees of freedom to the interface that the user has to adjust. (3) Last and not least, the image modification should not confuse the user's spatial perception and it should not induce user cybersickness.

(1) Our technique achieves disocclusion based on a multiperspective visualization that changes the viewpoint over the central part of the user field of view to maximize the image footprint of the objects. The multiperspective visualization is constructed with a two-bend graph camera [26] that augments the primary (user) viewpoint with auxiliary viewpoints to improve object visibility. (2) The construction of the two-bend camera depends on the placement of the auxiliary viewpoints. Our construction algorithm selects the best auxiliary viewpoints automatically, by maximizing object visibility, without any user interaction. (3) In order to avoid user disorientation and cybersickness, the change of perspective is confined to a circular region at the center of the image, whereas the periphery of the image is rendered conventionally, from the user perspective. Visualization continuity is maintained with a transition region over which the visualization changes viewpoint gradually.

We construct the disocclusion headlight visualization in real time, for the current user view, according to the pipeline shown in Fig. 2. A set of candidate auxiliary viewpoints is generated and then each candidate is evaluated in turn (Sec. 3.1). A candidate is evaluated by constructing a two-bend graph camera, by rendering the VE multiperspectively with the two-bend graph camera (Sec. 3.2), and then by measuring the disocclusion achieved by the candidate viewpoint in the multiperspective rendering (Sec. 3.3). The candidate auxiliary viewpoint with the highest disocclusion score is selected and used to warp the VE (Sec. 3.4). When rendered conventionally, the warped VE generates an image that shows the disocclusion effect at the center, continuously connected to the periphery of the image which is rendered conventionally from the user viewpoint. Finally, the warped VE is rendered conventionally for the user's left and right eyes. Given the disoccluded view of the VE, the user selects the desired object with a conventional virtual laser pointer cast with a

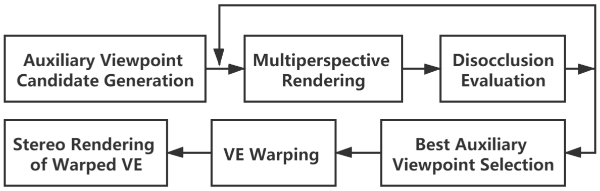
Auxiliary Viewpoint Candidate Generation
Multiperspective Rendering
Disocclusion Evaluation
Stereo Rendering of Warped VE
VE Warping
Best Auxiliary Viewpoint Selection

Figure 2: Pipeline for disocclusion headlight visualization construction.

tracked handheld controller.

3.1 Auxiliary viewpoint candidate generation

The disocclusion effect is based on an auxiliary viewpoint that maximizes the image footprint of the objects at the center of the user image. We find a good auxiliary viewpoint by trying several candidates. The candidates are generated on a vertical plane through the user viewpoint V_0 (Fig. 3, left). The candidate auxiliary viewpoints sample the vertical plane with equal angle increments to the 3D focal point F. F is the average VE geometry samples captured by the user image over a central circular region.

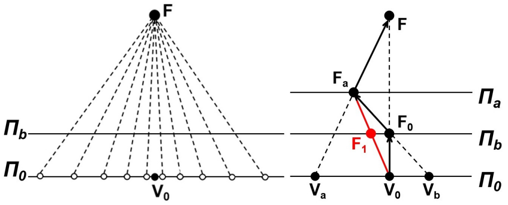

Figure 3: *Left.* Candidate auxiliary viewpoints plane Π_0 (top view). *Right.* Two-bend graph camera need and construction. A one-bend graph camera with viewpoints V_0 and V_a would translate the projection of F from F_0 to F_1. We use a two-bend graph camera, with viewpoints V_0, V_b, and V_a, which achieves the desired disocclusion while keeping the projection of F at the original location F_0.

3.2 Multiperpsective rendering

The auxiliary viewpoint V_a improves the image footprint of the objects at the center of the frame. What is needed, is a method for connecting V_a to the user viewpoint V_0, such that nearby geometry is imaged from V_0 and geometry farther away is imaged from V_a. The simple approach of resorting to a one-bend graph camera is not suitable because it would translate the look-at point projection to the left. In Fig. 3, right, a one-bend graph camera with viewpoints V_0 and V_a and bending plane Π_b would project F to F_a instead of the original location F_0. What is needed is an additional viewpoint V_b, which generates a two-bend graph camera that achieves the disocclusion afforded by V_a, while keeping the projection of F in place, at F_0. The two-bend graph camera rays are piecewise linear, i.e., they are three segments concatenated with C^0 continuity. The rays bend at vertical planes Π_b and Π_a, which are at predefined distances from V_0. In practice Π_b and Π_a split the distance V_0F into three equal parts. V_b is defined as $F_0F_a \bigcap \Pi_0$, where F_a is $V_aF \bigcap \Pi_a$, and Π_0 is a plane parallel to Π_b through V_0.

Once the two-bend graph camera is constructed, the VE is prerendered with it in order to quantify the disocclusion effect achieved by the current auxiliary viewpoint. Graph camera rendering is fast as it provides a projection operation that takes a given VE vertex directly to the multiperspective image plane [26], using a concatenation of the projection matrices along the path from the root viewpoint V_0 to the auxiliary viewpoint V_a. Fig. 4 shows the two-bend graph camera renderings for two candidate auxiliary viewpoints, one that improves the separation of the three spheres and one that makes the occlusion worse. The disocclusion achieved by an individual candidate auxiliary viewpoint and its corresponding two-bend graph camera image is quantified as follows.

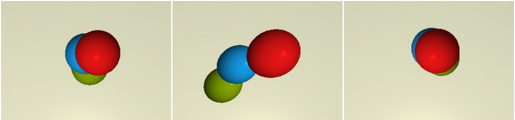

Figure 4: Conventional image from user viewpoint (left), two-bend graph camera image from a candidate auxiliary viewpoint that increases the image footprint of the three spheres (middle), and two-bend camera image from a candidate auxiliary viewpoint not suitable for disocclusion (right).

3.3 Disocclusion Evaluation

Given a candidate auxiliary viewpoint, the disocclusion effect achieved is quantified with Algorithm 1. The input to the algorithm is the candidate auxiliary viewpoint V_a; the user viewpoint V_0; the 3D focus point F; a circular central focus region D where the disocclusion effect occurs; the geometric model VE of the virtual environment. The output of the algorithm is a measure d of the disocclusion achieved by the candidate auxiliary viewpoint V_a.

Algorithm 1 Auxiliary viewpoint disocclusion quantification.

```
Input: candidate auxiliary viewpoint $V_a$, user viewpoint $V_0$, 3D
    focus point $F$, focus circle $D$, VE geometry $VE$
Output: Disocclusion value $d$ achieved by $V_a$
 1: $G$ = TwoBendGraphCamera($V_0$, $V_a$, $F$)
 2: MPI$(x,y,z,id,dc)$ = MultiperspectiveRendering($G$, $VE$)
 3: $L$ = ObjectsInsideRegion($G$, $VE$, $D$)
 4: $n$ = Length($L$); $V[1..n]$ = 0; $O[1..n]$ = 0
 5: for each pixel $p$ in $D$ do
 6:     $V[p.id] = V[p.id] + 1$
 7:     $O[p.id] = O[p.id] + p.dc - 1$
 8: end for
 9: $(V', O')$ = SortDescending$(V, O)$
10: for $i = 1$, $\psi_1 = 0$, $\psi_2 = 0$; $i \leq n$; $i$++ do
11:     if $V'[i] == 0$ then $m = i - 1$, break
12:     $\psi_1 = \max\{\psi_1, \frac{i}{n}\sqrt[i]{\prod_{j=1}^{i} V'[j]}\}$
13:     $\psi_2 = \psi_2 + V'[i]/(V'[i] + O'[i])$
14: end for
15: $d = w\ \psi_1/(\Sigma V'/m + \Sigma O')$ + (1-$w$) $(\psi_2/m)\ \Sigma V'$ / $(\Sigma V' + \Sigma O')$
```

At line 1, a two-bend graph camera G is constructed for V_a as described in Sect. 3.2.

At line 2, G is used to render VE to obtain a multiperspective image MPI with the following pixel channels: VE 3D position (x,y,z); object identifier id; object depth complexity dc. dc records the number of objects that cover the pixel, and it is computed with an accumulation buffer.

At line 3, the algorithm builds the list L of objects that project with G inside the focus circle D. L can also contain objects that are completely hidden in MPI.

At lines 4-8, the algorithm iterates over the focus circle D of MPI to quantify the visibility and the occlusion contribution of each object in L. There are n objects in L, and initially the visibility V and occlusion contribution O of each object is set to 0. The visibility $V[p.id]$ is the number of pixels that show object id. The occlusion contribution $O[p.id]$ measures the area, in pixels, where object id occludes another object, taking into account all occlusion layers. This allows penalizing an auxiliary viewpoint with a high occlusion depth complexity more than a viewpoint that has the same occlusion area, but with a smaller depth complexity.

At line 9, the algorithm sorts the object visibility array V from the most to the least visible object. The occlusion contribution array O is shuffled along during sorting in order to keep O in sync with V.

At lines 10-14, the algorithm counts the number m of visible objects in L, and computes quantities ψ_1 and ψ_2. ψ_1 quantifies the aggregate visibility of a prefix sequence of visible objects of increasing length. The aggregate visibility is computed based on the geometric mean of the first i visible objects, and ψ_1 is updated if the extension of the sequence to the next object increases the value of ψ_1 (line 12). A geometric mean is used as it is more robust to outliers, compared to the arithmetic mean, i.e., it does not give an excessive importance to objects with large visible footprints. ψ_2 takes into account an object's occlusion contribution, which reduces its visibility (line 13). This way an object that is visible over many pixels but also hides many other objects will have a smaller contribution to ψ_2 compared to an object that does not occlude other objects.

The disocclusion value d achieved by candidate auxiliary viewpoint V_a is computed as a linear interpolation between the normalized values of ψ_1 and ψ_2 (line 15). The linear interpolation uses a constant weight w (0.7 in our implementation) that modulates the relative weight of visibility (0.7) versus occlusion contribution (0.3). In Fig. 4, the disocclusion values for the three images are 0.61, 0.90, and 0.54, which clearly favors the middle image.

3.4 Virtual Environment Warping

Once the best candidate auxiliary viewpoint is found, the corresponding two-bend graph camera is used to warp the VE such that when the warped VE is rendered conventionally the resulting visualization improves the visibility of the objects in the focus circle. VE warping proceeds by VE vertex displacing. In order to achieve a smooth transition from the focus to the peripheral context region, the vertex displacement is tapered off gradually over a circular transition region. Fig. 6 shows complete frames corresponding to Fig. 4. Fig. 5 illustrates the displacement conceptually.

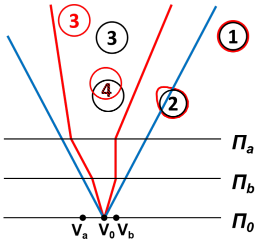

Figure 5: Virtual environment with four spheres before (black) and after (red) warping. Sphere 1 is unaffected since it is outside the transition region. Sphere 2 is partially warped as it crosses the outer boundary (blue line) of the transition region. Spheres 3 and 4 are inside the focus region (red lines) so they are displaced, but not distorted, to remove the occlusion of 3 by 4.

A given VE geometry vertex P is displaced based on its location, according to Algorithm 2. If P projects conventionally beyond the transition circle T, P is not displaced (lines 1 and 2). Otherwise, if P projects with the two-bend graph camera inside the focus circle at P_b (line 3), then the displaced vertex P' is computed by unprojecting P_b with the conventional user camera C_0, along ray V_0P_b, at a depth equal to the sum of the piecewise linear segments of the generalized graph camera ray that passes through P (line 4). The graph camera ray inflection points P_a and P_b are shown in Fig. 7. For illustration clarity, in Fig. 7, Π_b also serves as the image plane.

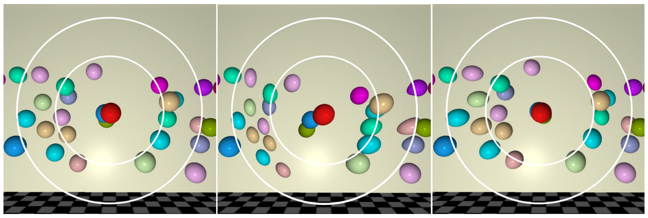

Figure 6: Image rendered conventionally from the user viewpoint (left), and with the disocclusion headlight built for the best (middle) and worst (right) auxiliary viewpoints from Fig. 4. VE warping tapers off the transition region (big white circle) that encompasses the focus region (small white circle)

Algorithm 2 Vertex displacement for VE warping.

```
Input: VE 3D vertex $P$; conventional user camera $C_0$ with user
    viewpoint $V_0$; two-bend graph camera $G$ constructed for $V_0$,
    with best auxiliary viewpoint $V_a$, an additional viewpoint $V_b$,
    and focal point $F$; focus circle $D$; and transition circle $T$.
Output: displaced 3D vertex $P'$.
 1: $P_0 = C_0$.Project($P$)
 2: if $P_0 \notin T$ then $P' = P$ // inside periphery
 3: else if ($P_b = G$.Project($P$)) $\in D$ // inside focus ($P$ in Fig. 7)
 4:         $P' = C_0$.Unproject($V_0$, $\|V_0P_b\| + \|P_bP_a\| + \|P_aP\|$)
 5: else if ($P = Q$) $\in [\Pi_a, \infty]$ // inside transition far ($Q$ in Fig. 7)
 6:         $C_q = C_aF \cap \Pi_Q$
 7:         $F_q = C_q + (Q\text{-}C_q)/\|Q\text{-}C_q\|*r_q$, $r_q = D.r*\Pi_q.z/\Pi_b.z$
 8:         $T_q = F_qQ \cap \mathrm{Cone}(V_0, T)$
 9:         $F_a = V_aF_q \cap \Pi_a$, $T_a = V_0T_q \cap \Pi_a$
10:         $t = \|F_qQ\|/\|F_qT_q\|$
11:         $Q_a = F_a + (T_a$ - $F_a)*t$
12:         $F_b = V_bF_a \cap \Pi_b$, $T_b = V_0T_a \cap \Pi_b$, $Q_b = F_b + (T_b$ - $F_b)*t$
13:         $P' = Q' = C_0$.Unproject($Q_b$, $\|Q_bQ_a\| + \|Q_aQ\|$)
14: else if $P \in [\Pi_b, \Pi_a]$ // inside transition near ($R$ in Fig. 7)
15:         // projection $Q_b$ is computed like above
16:         $P' = R' = C_0$.Unproject($Q_b$, $\|Q_bR\|$)
17: return $P'$
```

Otherwise, P is located in the transition region, and P is displaced by first projecting it on the image plane and then by unprojection in 3D space with the conventional user camera C_0. The projection of P has to interpolate between conventional and graph camera projection to ensure continuity between the context and the focus regions. The algorithm distinguishes between two cases, based on whether P is beyond the second graph camera plane Π_a (lines 5-13), or in between the two graph camera planes Π_b and Π_a (lines 14-16).

The case when P is beyond P_a is illustrated in Fig. 7 with point Q. The projection of Q onto the image plane at Q_b is computed in two steps. First, Q is projected onto Π_a at Q_a (lines 6-11), and then Q_a is projected onto Π_b at Q_b (line 12).

Point Q_a is computed as follows. Point C_q is computed as the intersection between the central ray C_aF of the graph camera, which passes through F, and the plane Π_Q, which is parallel to the image plane and passes through Q (line 6). Point F_q is computed as the intersection of the focus region boundary and line QC_q (line 7). The intersection of the focus region boundary and Π_Q is a circle of radius r_q; r_q is the radius of the focus region D_r, scaled up by the ratio between the depths $\Pi_q.z$ and $\Pi_b.z$ of planes Π_q and Π_b. Point T_q is computed as the intersection between F_qQ and the transition region boundary, which is the cone defined by the transition circle T and user viewpoint V_0 (line 8). Points F_a and T_a are computed by intersecting Π_a with rays from V_a and V_0 through F_q and T_q, respectively (line 9). Finally, Q_a is computed as the point that splits segment F_aT_a with the same ratio t as Q splits segment F_qT_q (lines 10-11). The projection Q_b of vertex Q onto the image plane is then computed as the point that splits segment F_bT_b with the same ratio t. The endpoints of F_bT_b are computed by intersecting plane Π_b with

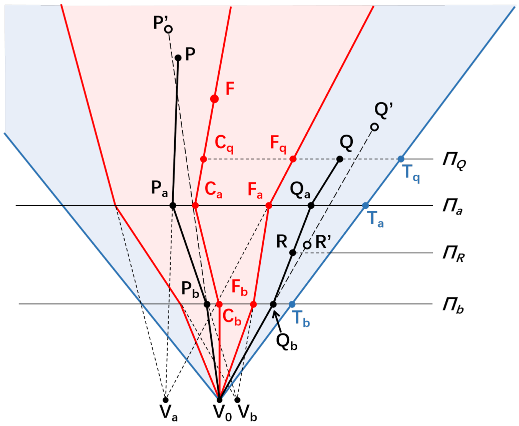

Figure 7: VE vertex displacement. A VE vertex P inside the focus region is displaced by projection with the two-bend graph camera $G(V_0, V_a, V_b, \Pi_a, \Pi_b)$ to P_b followed by unprojection with the conventional user camera to P'. A vertex Q inside the transition region and beyond Π_a is displaced by projection to Q_a and then to Q_b, followed by unprojection. A vertex R inside the transition region and between Π_a and Π_b is displaced by projection to Q_b, followed by unprojection. Vertices beyond the transition region are not displaced.

rays from V_b and V_0 through F_a and T_a, respectively (line 12). Q_b is unprojected back to 3D space with the conventional user camera, at a depth equal to the length of the piecewise linear *interpolated* graph camera ray (line 13).

When P is between planes Π_b and Π_a, the displacement is computed similarly. This case is illustrated in Fig. 7 by vertex R, which was chosen to have the same projection Q_b as Q, for clarity. Unlike for Q, R is projected directly to Q_b, in a single step. Q_b is then unprojected back to 3D with C_0 at depth $\|Q_bR\|$ (line 16).

The warped VE is rendered conventionally for the user's left and right eyes, producing the headlight disocclusion visualization, with a focus region that optimizes object visibility, connected seamlessly to the peripheral region (Fig. 6). Since the warped VE is a 3D scene rendered in stereo, the user benefits of depth perception, as in any conventional VR visualization. Instead of warping the VE and then rendering it conventionally, one could render two multiperspective images of the original VE, once for each eye. However, this would face the challenges of a higher computational cost, of an unstable disocclusion effect as the user moves their head slightly during the selection operation, and of an unstable depth perception, due to the changing aggregate length of the piecewise linear rays.

4 User Study: Results and Discussion

We have evaluated our disocclusion headlight VR selection method in a controlled between-subject user study with three tasks, where we compared our method to two state of the art methods (Sect. 4.1). The results show significant advantages for our method (Sect. 4.2). We have also investigated spatial awareness in additional perceptual experiments (Sect. 4.3).

4.1 VR Selection Study Design

Participants. We have recruited 24 participants, 19 male and 5 female, between 20 and 40 years old. 16 of our participants had used immersive VR applications before. Participants had normal or corrected vision, and none reported vision or balance disorders. The participants were randomly assigned to two control groups and one experimental group, each with eight participants. The first control group (CG_1) used the alpha cursor prior art VR selection method [42]. With the alpha cursor the user can slide a cursor on a virtual laser beam to define the radius of a cutoff sphere. In Fig. 8, left, the user pushes the cursor beyond object 2 to see what it might hide; this increases the radius of the cutoff sphere which clips 2 to reveal 3. The second control group (CG_2) used the flower cone prior method [42] (Fig. 8, right). The user defines the angle of a selection cone aimed like a flashlight. All objects whose centers are inside the cone are shown to the user in a selection candidates matrix pattern. The experimental group (EG) used our disocclusion headlight method. When, during browsing, the user stops moving their head, detected as a drop of frame to frame view changes below a threshold (0.01cm for translation and 0.5deg for rotation), the disocclusion effect is computed and deployed automatically. During deployment, the system continues to update the user image by rendering the warped VE for the current user view. The disocclusion effect is retracted automatically once the user changes the view considerably.

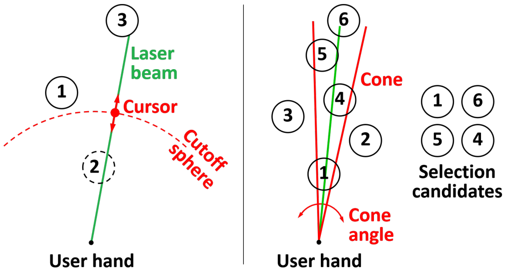

Figure 8: Alpha cursor (left) and flower cone (right) methods [42] used as control conditions.

Hardware and Software Implementation. We used an HTC Vive system which has a tracked HMD and a wireless hand-held controller. The HMD is tethered to a desktop PC (Intel i7 processor, 16GB RAM, and NVIDIA 1080 graphics card). The virtual environments were rendered at 90fps for each eye. The auxiliary viewpoint selection requires rendering the VE multiple times. In practice, we test 100 auxiliary viewpoints, which takes between 18ms and 25ms for our three VE's described below, which have between 200k at 390k triangles. The auxiliary viewpoint optimization proceeds every time the user stops moving their head, which could indicate the intention to deploy the disocclusion effect. The optimization time is short enough to be hidden in the time period between the user stopping and engaging the disocclusion effect. The disocclusion effect itself is deployed over 300ms for a gradual transition. The tracked physical space hosting the VR applications is 4m × 4m, which was sufficient for the participant to perform the VR selection tasks without the need of redirection or teleportation.

Task 1. For the first task (T1), the participant had to select 17 brown spheres, one at the time, from a VE of 206 multicolored spheres (Fig. 1). Three of the brown spheres are completely occluded and the other 14 are at least partially visible. The VE is 5m x 5m, all spheres have a 32cm diameter, the user is at the center of the VE, and the spheres surround the user. The task is complete when all brown spheres are found. Task 1 is illustrated in Fig. 1 (EG) and Fig. 9 ($CG1$ and $CG2$).

Task 2. For the second task (T2), the VE had 200 32cm pink and blue spheres, as well as 12 32cm green cubes, and the participant had to select the two spheres closest to each cube, one cube at the time (Fig. 10). All cubes are readily visible from the user default position, so no searching is involved. The task investigates whether the disocclusion effect of the VR interface preserves local spatial relationships. The task is complete when the correct spheres are selected for all cubes.

Task 3. For the third task (T3), the VE is a warehouse with 480 cardboard boxes on shelves (Fig. 1, right, and Fig. 11). The VE is 20m x 10m, it is 6m tall, and the boxes are cubes of size chosen randomly between 0.4m and 0.6m. 12 of the boxes contain bananas and the other ones contain apples. The content of the box is reflected by the textures applied to its faces. The boxes were placed on the shelves randomly, using a physics engine that randomized the initial

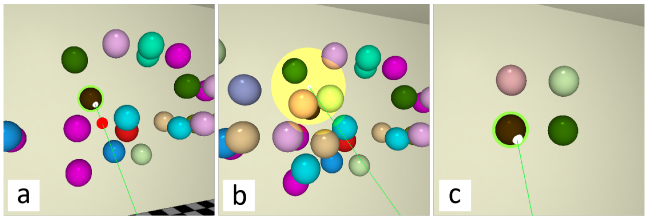

Figure 9: Frames illustrating the control conditions for task 1, corresponding to the conventional and experimental condition frames shown in Fig. 1, left. The brown sphere is not visible in the conventional frame. The user manipulates the alpha cursor to cut away the occluding spheres (a). With the flower cone, all four spheres in the yellow cone (b) are shown in a matrix pattern (c).

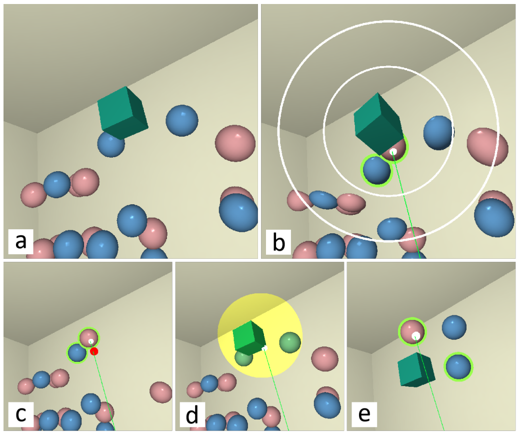

Figure 10: Frames illustrating task 2 where the participant has to select the two spheres closest to the cube. The conventional frame (a) does not show the pink sphere which is close to and hidden by the cube. The disocclusion headlight frame (b) reveals the hidden sphere. The alpha cursor (c) cuts away the cube to reveal the sphere. For the flower cone the cube and three spheres are in the cone (d), but it is impossible to judge relative distances in the matrix pattern (e).

position of the box and then used gravity, friction, and collision detection to compute the final position of each box. The user had to select each banana box, as well as the apple box closest to it.

Procedure All participants performed the three VR selection tasks. Since the control and experimental groups change from task to task, a participant might perform, for example, T1 in the EG condition, T2 in the CG1 condition, and T3 in the CG2 condition. The tasks are performed in the same order T1, T2, and T3, with at least one day, and at most three days of rest in between tasks. For each task, the participant practices for one minute. For all tasks all participants found all selection targets, so no one timed out. When the correct object is selected, the system confirms by placing a green highlight around it. The highlight persists until the end of the task, so a participant never selects the same object multiple times. An incorrect selection is indicated with a temporary red highlight. Participants can select the same incorrect object multiple times, which counts as multiple errors. Participants were told that the task completion time is recorded, which was an indirect encouragement for them to complete the task as quickly as possible.

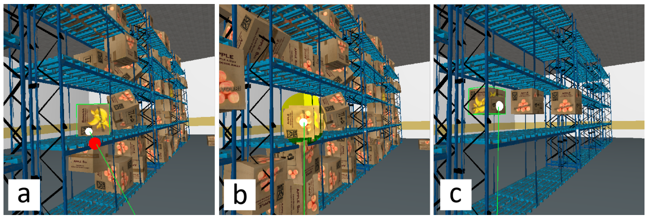

Figure 11: Frames illustrating the control conditions for task 3, corresponding to the conventional and experimental condition frames shown in Fig. 1, right: alpha cursor (a), flower cone (b and c).

Table 1: Correct selection rate.

Task	Condi -tion	Avg ± std. dev.	(EG-CG_i) / CG_i	p	Cohen's d	Effect size
T1	EG	0.96 ± 0.03				
	CG_1	0.80 ± 0.07	19.9%	$< 0.001^*$	3.12	Huge
	CG_2	0.87 ± 0.06	10.8%	$< 0.001^*$	2.11	Huge
T2	EG	0.86 ± 0.03				
	CG_1	0.73 ± 0.02	17.9%	$< 0.001^*$	5.97	Huge
	CG_2	0.55 ± 0.10	56.3%	$< 0.001^*$	4.36	Huge
T3	EG	0.88 ± 0.05				
	CG_1	0.62 ± 0.11	41.8%	$< 0.001^*$	3.09	Huge
	CG_2	0.31 ± 0.09	100%	$< 0.001^*$	7.75	Huge

Metrics. Task performance was measured with the following objective metrics: (1) correct selection rate, defined as the number of correct selections over the total number of selections; (2) task completion time, in seconds; (3) total viewpoint translation, in meters, defined as the sum of all frame to frame head translations. Since viewpoint translation is the conventional, intuitive solution for disocclusion, the viewpoint translation metric offers an indirect measure of the power of the disocclusion technique used. If the disocclusion technique does not reveal the desired object from the current viewpoint, the user will apply the selection technique from a different viewpoint, which is reflected in the viewpoint translation metric. We also evaluated the VR experience with two subjective metrics: user task load, measured with the standard NASA TLX questionnaire [18, 19], and user cyber sickness, measured with the standard SSQ questionnaire [20].

Statistical analysis. For each metric, the EG values were compared to CG_1 and to CG_2. The comparison was performed using a Wilcoxon signed-rank test [29]. In addition to the p value of the statistical test, we also estimated the size of the effect using Cohen's d [10]. The d values were translated to qualitative effect size estimates of *Huge* ($d > 2.0$), *Very Large* ($2.0 > d > 1.2$), *Large* ($1.2 > d > 0.8$), *Medium* ($0.8 > d > 0.5$), *Small* ($0.5 > d > 0.2$), and *Very Small* ($0.2 > d > 0.01$).

4.2 Results and Discussion

Correct selection rate. Table 1 gives the correct selection rate for the three methods and for the three tasks. Column three gives the average value and the standard deviation, column four gives the relative increase of the correct selection rate from CG to EG, and columns five to seven give the statistical analysis of the EG versus CG differences. Statistical significance is indicated with an asterisk.

EG had significantly higher correct selection rates for all tasks ($p \leq 0.01$), and the effect was *Huge*. For task 1, CG_1 has the highest error rate as participants stopped adjusting the alpha cursor as soon as a sliver of the selection target became visible, which resulted in more frequent errors due to the difficulty of aiming the virtual laser beam at the target sliver. CG_2 objects commit errors when they engage the flower cone repeatedly without finding the brown sphere and they end up selecting the darkest colored sphere in the matrix lineup, even if it is not brown. EG objects selected targets more easily using the increased image footprint of candidate selection objects afforded by the disocclusion headlight. EG participants found on average 16.32 spheres, and missed at most one sphere, which indicates that our method works even for the three fully occluded spheres. The CG_1 and CG_2 groups missed as many as 6 spheres. For task 2, CG_2 participants committed frequent errors as the matrix lineup provided

Table 2: Task completion time, in seconds.

Task	Condi -tion	Avg ± std. dev.	$(CG_i\text{-}EG)$ / CG_i	p	Cohen's d	Effect size
T1	EG	39 ± 2.7				
	CG_1	75 ± 9.1	47.7%	$< 0.001^*$	5.29	Huge
	CG_2	111 ± 14	64.8%	$< 0.001^*$	7.16	Huge
T2	EG	64 ± 14				
	CG_1	101 ± 31	36.2%	$< 0.001^*$	1.51	Very Large
	CG_2	134 ± 27	51.9%	$< 0.001^*$	3.25	Huge
T3	EG	83 ± 12				
	CG_1	137 ± 8.2	39.7%	$< 0.001^*$	5.23	Huge
	CG_2	176 ± 47	53.2%	$< 0.001^*$	2.74	Huge

Table 3: Viewpoint translation, in meters.

Task	Condi -tion	Avg ± std. dev.	$(CG_i\text{-}EG)$ / CG_i	p	Cohen's d	Effect size
T1	EG	2.7 ± 0.9				
	CG_1	6.1 ± 2.5	55.6%	$< 0.001^*$	1.83	Very Large
	CG_2	6.3 ± 0.8	57.1%	$< 0.001^*$	4.28	Huge
T2	EG	5.0 ± 1.7				
	CG_1	5.3 ± 1.9	5.8%	0.01^*	0.17	Very Small
	CG_2	8.8 ± 3.9	43.8%	0.002^*	1.27	Very Large
T3	EG	5.0 ± 0.8				
	CG_1	4.8 ± 0.4	-3.8%	0.067	0.30	Small
	CG_2	6.5 ± 3.2	23.4%	0.17	0.64	Medium

by the flower cone breaks local spatial relationships, and correct selection was only possible when the flower cone happened to only contain the two closest spheres. CG_1 and EG participants had to judge distances in a perspective projection, so performance was not perfect. EG participants performed better because the view direction chosen by the disocclusion headlight to improve the visibility of objects implicitly improves the illustration of the distance between objects. Task 3 shows that the advantage of our method increases as the complexity and realism of the VE increases.

Task completion time. Table 2 gives the task completion times. EG participants complete all tasks significantly faster. The automatic deployment of the headlight disocclusion effect allows the participants to find the selection targets quickly. The two-step selection process of the flower cone is slowest (CG_2).

Viewpoint translation. Table 3 gives the user viewpoint translation. For tasks T1 and T2, EG participants translate significantly less, whereas for T3, EG participants translate more than the CG participants. For T1 and T2 the VE is small, with occluding spheres close to the participant, who instinctively translates their head to disocclude and only engage the alpha cursor or the flower cone when the translation is not sufficient; with our disocclusion headlight, the occlusions are removed automatically so the participant can proceed directly to selection. For T3, the VE is larger, with more occluding layers, and our disocclusion headlight cannot separate all cardboard boxes aligned with the participant view direction, so the participant has to translate the viewpoint; both the alpha cursor and the flower cone have the ability to separate any number of occluding layers, albeit sequentially, or by perturbing the local spatial relationships; this leads to translation differences that are not statistically significant.

We measured user task load using the NASA TLX questionnaire, with the results given in Table 4. Overall, EG had lower user load scores. The advantage was the least significant for T3 over CG_1, where CG_1 participants stayed in one place and played with the alpha cursor, whereas EG participants had to walk more to overcome the higher depth complexity. The advantage was significant for T1 over both CG_1 and CG_2, where the task was straightforward for EG, and for T2 over CG_2, where the perturbation of the local spatial relationships by the flower cone made the users struggle with finding the two spheres closest to each cube.

Participants completed the SSQ before and after each task (Table 5). No significant increase between the pre and post SSQ scores were recorded for any of the tasks, and for any of the conditions, including for our disocclusion headlight method. No participant reported visual fatigue through the relevant SSQ questions.

Table 4: NASA Task Load Index data.

Task	Condition	Avg ± std. dev.	p	Cohen's d	Effect size
T1	EG	23 ± 2.7			
	CG_1	25 ± 3.9	0.029^*	0.85	Large
	CG_2	24 ± 1.1	0.021^*	0.61	Medium
T2	EG	25 ± 2.8			
	CG_1	26 ± 3.5	0.405	0.46	Small
	CG_2	30 ± 4.2	0.003^*	1.34	Very Large
T3	EG	26 ± 3.5			
	CG_1	26 ± 6.8	0.673	0.04	Very Small
	CG_2	33 ± 10	0.122	0.88	Large

Table 5: Simulator Sickness Questionnaire data.

Task	Condition	preAvg ± std. dev.	postAvg ± std. dev.	p
T1	EG	5.3 ± 4.9	5.5 ± 5.6	0.59
	CG_1	6.3 ± 5.9	8.5 ± 5.8	0.27
	CG_2	7.8 ± 5.6	8.1 ± 6.9	0.41
T2	EG	5.6 ± 3.8	6.3 ± 4.1	0.47
	CG_1	7.4 ± 5.2	7.9 ± 6.7	0.31
	CG_2	9.0 ± 2.5	9.3 ± 3.7	0.46
T3	EG	7.2 ± 8.8	7.4 ± 8.5	0.97
	CG_1	8.2 ± 7.0	22 ± 13	0.16
	CG_2	11 ± 2.9	12 ± 4.0	0.63

4.3 Perceptual Experiments

We have also conducted two additional perceptual experiments to compare our method to the alpha cursor and the flower cone in terms of depth estimation and layout perception.

In the depth estimation experiment (Fig. 12, a-b), the participant is shown the conventional view (left), the disocclusion headlight effect is gradually deployed (right), two spheres are highlighted, and the participant is asked to decide which one is farther. The task is repeated four times. Table 6 shows the depth perception error rate, which is similar for EG and CG_1 (difference not significant), and which is almost significantly lower for EG versus CG_2. As expected, the flower cone approach precludes accurate depth estimation as it perturbs local spatial relationships.

Table 6: Depth estimation and layout perception error rates.

	Depth estimation		Layout perception	
Condition	Avg ± std. dev.	p	Avg ± std. dev.	p
EG	0.08 ± 0.14	–	0.33 ± 0.14	–
CG_1	0.00 ± 0.00	0.37	0.33 ± 0.14	1.00
CG_2	0.50 ± 0.25	0.07	0.42 ± 0.14	0.52

The spatial layout perception experiment uses the VE of T2 (Fig. 12, c-d). The participant is first shown a conventional view (left) which then changes to the disocclusion headlight view (right).

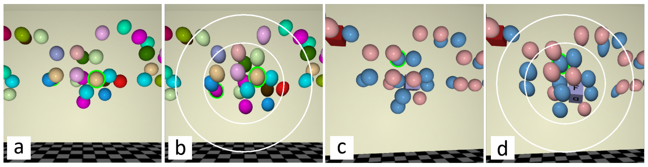

Figure 12: *Depth estimation experiment: conventional (a) and disocclusion headlight (b).* The participant has to indicate which of the two highlighted spheres (green halo) is farther. *Layout perception experiment: conventional (c) and disocclusion headlight (d).* The participant has to indicate the position of the highlighted sphere (green halo) with respect to the cube faces (i.e., front, right, and above here).

A sphere in the proximity of a cube is highlighted. The cube's faces are labeled with F, B, L, R, U, and D for front, back, left, right, up, and down. The participant has to indicate the sphere's relative position to the cube, by answering three questions: (1) whether the sphere is in front or behind the cube, (2) whether it is to the left or to the right, and (3) whether it is above or below. Table 6 shows that the layout perception error rate for EG is not worse than for CG_1 or CG_2, which confirms that the VE warping of the disocclusion headlight does not lower the layout perception of the participant compared to the two state-of-the-art methods.

5 Conclusions, Limitations, and Future Work

We have described the disocclusion headlight, a method that facilitates object selection in VR by maximizing the projection footprint of objects at the center of image. In our experiments, our method has proven to be more accurate and faster than two state-of-the-art VR selection methods. The advantages stem from the fact that our method is automatic, i.e., it does not rely on the user to manipulate the interface to achieve disocclusion, and that our method preserves local spatial relationships.

Our studies investigate selection as part of a continuous VE browsing operation, and do not study selection in isolation, by jumping abruptly from one selection sub-task to the next. Although this leaves some residual searching effort, the searching effort is minimal, as most brown spheres are partially visible for task 1, and as all cubes are visible in task 2. The benefit of investigating selection with a sequential, continuous browsing of the VE is that selection performance is measured more accurately, for example by including the times needed for the deployment and retracting of the multiperspective disocclusion effect.

Our method relies on finding an auxiliary viewpoint from where the selection target is at least partially visible, and it does so by searching a dense but none-the-less discrete set of candidate viewpoints. This task is an example of a next-best-view visibility problem, for which no efficient and exact solutions are known. Our method facilitates selection by replacing the conventional image with a multiperspective image that shows more than what is visible from the current viewpoint, reducing the amount of viewpoint translation the user has to perform to establish line of sight to the target.

Limitations. Our method has several limitations. Since our method disoccludes by routing generalized, piecewise linear rays on a path from the user to the selection target, our method can only succeed if such a path exists (e.g., it cannot disocclude an object enclosed in a box). Our method uses a two-bend construction which cannot reach the back of objects; for example, to see the back face of a cube aligned with the image plane, the user has to translate the viewpoint first to see the cube at an angle, before the disocclusion camera can reveal the back face. Another limitation is that our method optimizes the footprint of all objects, and it can happen that the footprint of the object of interest, which is of course not known to the system, is sacrificed in the interest of a higher average object footprint. Our method is designed to preserve local spatial relationships, which comes at the cost of not supporting a large number of occluding layers of a highly cluttered scene. Our image changes the viewpoint on the focus region and maintains the original view on the peripheral region; both of these regions are free of distortions, but continuity between them is maintained with a transition region where the VE is warped, which could distract the user.

Our method shows the user an image that is different from what they would see with the naked eye. We have chosen to warp the VE by unprojection at a depth commensurate with the length of the piecewise linear rays, which increases the depth perception separation between occluder and occludee, as the non-linear rays are always longer. Our users did not comment on the disparity between occluders and occludees. Another approach is to keep the occluder at its original depth, i.e., in Fig. 7, to place P' to the left of P, at the same depth. We estimate that the depth perception difference between the two approaches is small, and it is probably more salient in VE's with attenuation with distance effects, such as fog.

Our method optimizes the placement of the auxiliary viewpoint by rendering the VE multiple times. For our examples, the VE's are sufficiently simple such that this does not constitute an impediment to interactivity. Future work could handle more complex VE's by limiting the rendering complexity with aggressive culling; in addition to the traditional view frustum and occlusion culling approaches, the optimization could also aggressively bring the far plane in, as selection targets are typically close to the user; furthermore, the rendering passes for the optimization require no shading, and a lower resolution should be sufficient.

Future work. Some of the limitations could be addressed in future work by increasing the disocclusion capability of our method. One approach is to allow for additional bends, which can disocclude the back face of an object, and can support a higher number of occluding layers, at the cost of a greater perturbation of spatial relationships compared to a conventional image. Another approach is to add a degree of freedom to the disocclusion headlight interface; for example, the user could press a button to choose another local maximum of the footprint optimization, which might reveal the object of interest. This has the potential to extend the applicability of our method to cluttered VEs with a high number of occluding layers, such as complex graph networks.

Our studies involved a small number of participants, as conducting experiments has been extremely challenging during the COVID-19 pandemic. However, the effects we have found are sizeable, which indicates that our trials do have at least limited statistical power (e.g., Cohen's $d > 2.0$ for selection correctness, i.e., a huge effect size. Once possible, we will extend the investigation of our selection technique with more participants, with longer exposure times, and with additional prior selection techniques as control.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China through Projects 61932003 and 61772051, by National Key R&D plan 2019YFC1521102, by the Beijing Natural Science Foundation L182016, by the Beijing Program for International S&T Cooperation Project Z191100001619003, by the funding of Shenzhen Research Institute of Big Data (Shenzhen 518000).

References

[1] Steven Feiner Alex Olwal. "The Flexible Pointer: An Interaction Technique for Selection in Augmented and Virtual Reality". In: *Proc. UIST.* Vol. 3. 2003, pp. 81–82.

[2] Carlos Andújar and F. Argelaguet. "Anisomorphic Ray-Casting Manipulation for Interacting with 2D GUIs". In: *Comput. Graph.* 31 (2007), pp. 15–25.

[3] F. Argelaguet and Carlos Andújar. "A Survey of 3D Object Selection Techniques for Virtual Environments". In: vol. 37. 2013, pp. 121–136. DOI: 10.1016/j.cag.2012.12.003.

[4] F. Argelaguet, Carlos Andújar, and Ramón Trueba. “Overcoming Eye-Hand Visibility Mismatch in 3D Pointing Selection”. In: *VRST ’08*. 2008.

[5] Ferran Argelaguet and Carlos Andujar. “Efficient 3D Pointing Selection in Cluttered Virtual Environments”. In: *IEEE Computer Graphics and Applications* 29.6 (2009), pp. 34–43.

[6] Marc Baloup, Thomas Pietrzak, and Géry Casiez. “RayCursor: A 3D Pointing Facilitation Technique Based on Raycasting”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12.

[7] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. “MultiPoint: Comparing Laser and Manual Pointing as Remote Input in Large Display interactions”. In: *International Journal of Human-Computer Studies* 70.10 (2012), pp. 690–702.

[8] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. “Pointable: An In-Air Pointing Technique to Manipulate Out-of-Reach Targets on Tabletops”. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. 2011, pp. 11–20.

[9] Jeffrey Cashion, Chadwick Wingrave, and Joseph J LaViola Jr. “Dense and Dynamic 3D Selection for Game-Based Virtual Environments”. In: *IEEE transactions on visualization and computer graphics* 18.4 (2012), pp. 634–642.

[10] J. Cohen. “Statistical Power Analysis for the Behavioral Sciences”. In: Academic press, 2013.

[11] Gerwin De Haan, Michal Koutek, and Frits H Post. “IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection.” In: *Ipt/egve*. Citeseer. 2005, pp. 201–209.

[12] José Luis Dorado, Pablo Figueroa, Jean-Rémy Chardonnet, Frédéric Merienne, and José Tiberio Hernández. “Comparing VR Environments for Seat Selection in an Opera Theater”. In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2017, pp. 221–222.

[13] N. Elmqvist and P. Tsigas. “A Taxonomy of 3D Occlusion Management for Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 14 (2008), pp. 1095–1109.

[14] A. Forsberg, K. Herndon, and R. Zeleznik. “Aperture Based Selection for Immersive Virtual Environments”. In: *UIST ’96*. 1996.

[15] Tovi Grossman and Ravin Balakrishnan. “The Bubble Cursor: Enhancing Target Acquisition By Dynamic Resizing of the Cursor’s Activation Area”. In: *Proceedings of the 2005 Conference on Human Factors in Computing Systems, CHI 2005, Portland, Oregon, USA, April 2-7, 2005*. 2005. DOI: 10.1145/1054972.1055012.

[16] Tovi Grossman and Ravin 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*. 2006, pp. 3–12.

[17] Xiaoxia Han and Huagen Wan. “A Framework for Virtual Hand Haptic Interaction”. In: *Transactions on edutainment IV*. Springer, 2010, pp. 229–240.

[18] S. Hart. “Nasa-Task Load Index (NASA-TLX); 20 Years Later”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50 (2006), pp. 904–908.

[19] S. Hart and L. Staveland. “Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research”. In: *Advances in psychology* 52 (1988), pp. 139–183.

[20] R. Kennedy, N. E. Lane, K. Berbaum, and Lilienthal Mg. “Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness.” In: *The International Journal of Aviation Psychology* 3 (1993), pp. 203–220.

[21] Regis Kopper, Felipe Bacim, and Doug A Bowman. “Rapid and Accurate 3D Selection by Progressive Refinement”. In: *2011 IEEE symposium on 3D user interfaces (3DUI)*. IEEE. 2011, pp. 67–74.

[22] Yiqin Lu, Chun Yu, and Yuanchun Shi. “Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality”. In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2020. DOI: 10.1109/VR46266.2020.00021.

[23] Fabrice Matulic and Daniel Vogel. “Multiray: Multi-Finger Raycasting for Large Displays”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–13.

[24] Mark R. Mine. “Virtual Environment Interaction Techniques”. In: 1995. DOI: 10.1007/s100550200013.

[25] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, and Mark R. Mine. “Image Plane Interaction Techniques in 3D Immersive Environments.” In: *Symposium on Interactive 3D Graphics*. 1997.

[26] Voicu Popescu, Paul Rosen, and Nicoletta Adamo-Villani. “The Graph Camera”. In: *ACM SIGGRAPH Asia 2009 Papers*. SIGGRAPH Asia ’09. Yokohama, Japan: Association for Computing Machinery, 2009. ISBN: 9781605588582. DOI: 10.1145/1661412.1618504. URL: https://doi.org/10.1145/1661412.1618504.

[27] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. “The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR”. In: *Proceedings of the 9th annual ACM symposium on User interface software and technology*. 1996, pp. 79–80.

[28] Adrian Ramcharitar and Robert J Teather. “EZCursorVR: 2D Selection with Virtual Reality Head-mounted Displays”. In: *Proceedings of the 44th Graphics Interface Conference*. Canadian Human-Computer Communications Society. 2018, pp. 123–130.

[29] D. Rey and M. Neuhäuser. “Wilcoxon-Signed-Rank Test”. In: *International Encyclopedia of Statistical Science*. 2011.

[30] Tobias Rick, Anette von Kapri, and Torsten Kuhlen. “GPU Implementation of 3D Object Selection by Conic Volume Techniques in Virtual Environments”. In: *2010 IEEE Virtual Reality Conference (VR)*. IEEE. 2010, pp. 243–246.

[31] Greg Schmidt, Yohan Baillot, Dennis G Brown, Erik B Tomlin, and JE Swan. “Toward Disambiguating Multiple Selections for Frustum-Based Pointing”. In: *3D User Interfaces (3DUI’06)*. IEEE. 2006, pp. 87–94.

[32] Frank Steinicke, T. Ropinski, and Klaus H. Hinrichs. “Object Selection in Virtual Environments using an Improved Virtual Pointer Metaphor”. In: *ICCVG*. 2004.

[33] Robert J Teather and Wolfgang Stuerzlinger. “Pointing at 3D Targets in a Stereo Head-Tracked Virtual Environment”. In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2011, pp. 87–94.

[34] Lode Vanacken, Tovi Grossman, and Karin Coninx. “Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments”. In: *IEEE Symposium on 3D User Interfaces*. 2007. DOI: 10.1109/3DUI.2007.340783.

[35] L. Wang, Wentao Wu, Zijing Zhou, and Voicu Popescu. “View Splicing for Effective VR Collaboration”. In: 2020.

[36] Lili Wang, Jian Wu, Xuefeng Yang, and Voicu Popescu. “VR exploration assistance through automatic occlusion removal”. In: *IEEE transactions on visualization and computer graphics* 25.5 (2019), pp. 2083–2092.

[37] Colin Ware. “Using Hand Position for Virtual Object Placement”. In: *The Visual Computer* 6.5 (1990), pp. 245–253.

[38] Jonathan Wonner, Jérôme Grosjean, Antonio Capobianco, and Dominique Bechmann. “Starfish: A Selection Technique for Dense Virtual Environments”. In: *Proceedings of the 18th ACM symposium on Virtual reality software and technology*. 2012, pp. 101–104.

[39] M. L. Wu and V. Popescu. “Multiperspective Focus+Context Visualization”. In: *IEEE Transactions on Visualization Computer Graphics* 22.5 (2016), p. 1555.

[40] Meng-Lin Wu and Voicu Popescu. “Efficient VR and AR navigation through multiperspective occlusion management”. In: *IEEE transactions on visualization and computer graphics* 24.12 (2017), pp. 3069–3080.

[41] Hans Peter Wyss, Roland Blach, and Matthias Bues. “ISith-Intersection-Based Spatial Interaction for Two Hands”. In: *3D User Interfaces (3DUI’06)*. IEEE. 2006, pp. 59–61.

[42] Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, and J. Gonçalves. “Fully-Occluded Target Selection in Virtual Reality”. In: *IEEE transaction of computer graphics and visualization* (2020), pp. 62–65.