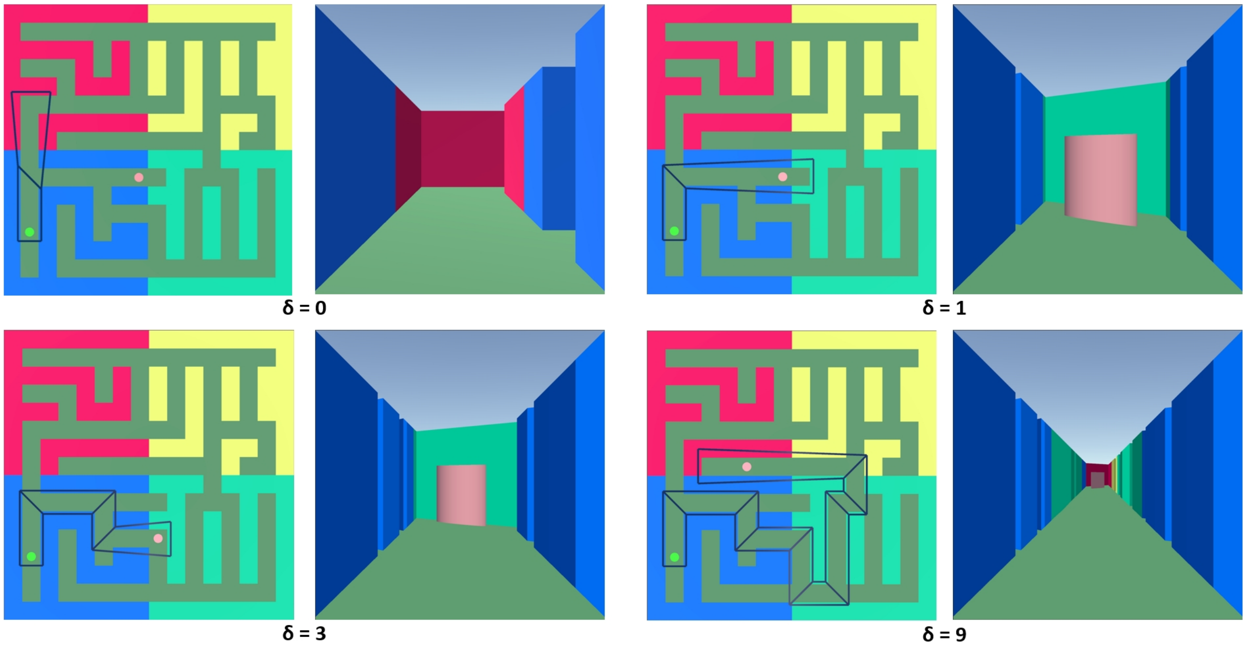
# PreVR: Variable-Distance Previews for Higher-Order Disocclusion in VR

Shuqi Liao, Vetria Byrd, and Voicu Popescu



δ = 0

δ = 1

δ = 3

δ = 9

Fig. 1: *PreVR* illustration in a 3D maze for various maximum disocclusion orders $\delta$. For each pair, the left image is an overhead view illustrating the construction of the disocclusion effect, showing the user viewpoint (green dot), the target (pink dot), and the underlying camera model (black lines); the right image is the user view. $\delta = 0$ corresponds to a conventional visualization that does not reveal the target around the corner. Higher $\delta$ values provide deeper previews revealing the target around 1, 3, or 9 corners.

**Abstract**—The paper introduces *PreVR*, a method for allowing the user of a VR application to preview a virtual environment (VE) around any number of corners. This way the user can gain line of sight to any part of the VE, no matter how distant or how heavily occluded it is. *PreVR* relies on a multiperspective visualization that implements a higher-order disocclusion effect with piecewise linear rays that bend multiple times as needed to reach the visualization target. *PreVR* was evaluated in a user study (N = 88) that investigates four points on the VR interface design continuum defined by the maximum disocclusion order $\delta$. In a first control condition (CC0), $\delta = 0$, corresponds to conventional VR exploration with no preview capability. In a second control condition (CC1), $\delta = 1$, corresponds to the prior art approach of giving the user a preview around the first corner. In a first experimental condition (EC3), $\delta = 3$, so *PreVR* provided up to third-order disocclusion. In a second experimental condition (ECN), $\delta$ was not capped, so *PreVR* could provide a disocclusion effect of any order, as needed to reach any location in the VE. Participants searched for a stationary target, for a dynamic target moving on a random continuous trajectory, and for a transient dynamic target that appeared at random locations in the maze and disappeared 5s later. The study quantified VE exploration efficiency with four metrics: viewpoint translation, view direction rotation, number of teleportations, and task completion time. Results show that the previews afforded by *PreVR* bring a significant VE exploration efficiency advantage. ECN outperforms EC3, CC1, and CC0 for all metrics and all tasks, and EC3 frequently outperforms CC1 and CC0.

**Index Terms**—Disocclusion, visualization, navigation, virtual reality

## 1 Introduction

Virtual reality (VR) provides a powerful visualization of 3D scenes through immersion, depth cues, and a natural user interface for view selection. However, the exploration of complex virtual environments (VEs) can be inefficient. Consider the scenario of a user searching for a target object in VR. Because of occlusions, the user cannot see the entire VE from the current location, so the user has to change view interactively to explore the VE sequentially. If the user reaches a previously hidden part of the VE just to realize that it does not contain the target, the navigation effort is wasted. The cost of the wasted navigation is even higher when the user has to teleport, which can induce user disorientation and cybersickness. Sequential exploration is particularly inefficient in the case of dynamic VEs, as the user has to be in the right place at the right time to find the target. If the target evades the user deliberately or if the target is only present for a short amount of time, the user might never find it.

Improving sequential navigation efficiency can be done by increas-

- *Shuqi Liao is with Purdue University. E-mail: liao201@purdue.edu*
- *Vetria Byrd is with Purdue University. E-mail: vlbyrd@purdue.edu.*
- *Voicu Popescu is with Purdue University. E-mail: popescu@purdue.edu.*

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

ing the power of the visualization provided to the user. The goal is to allow the user to *preview* hidden parts of the VE, *from the current location*, which avoids the unnecessary expense of navigating to hidden parts that prove to be of no interest. Prior work has investigated several approaches for removing occlusions to provide the user with a preview of the VE. One approach is to allow the user to see through occluders, using visualization methods like transparency [3, 44] and cutaway [9]. The approach has the advantage of familiarity to users, but transparency scales poorly with the number of occluding layers, and cutaway completely eliminates occluding layers that might be needed by some applications. Another approach is to rely on an overhead view, to which the user switches temporarily to examine the VE from above [26], or that is permanently displayed as a vignette within the user's field of view [24]. The advantage is that the user can monitor a large section of the VE, sustaining situational awareness. However, this comes at the cost of breaking the sense of presence when abandoning the first-person view for the overhead view, or at the cost of the low resolution visualization provided by the vignette. A third approach is to enhance the user's first-person view with additional perspectives that preview parts of the VE to which the user does not have line of sight. Prior work investigated allowing the user to "see around the corner" by integrating into the user's view a perspective of the VE captured from a secondary viewpoint [41]. An important limitation of this work is that the preview is limited to *first-order* occlusions, i.e., to what can be seen around the *first* corner, with the help of a *single* additional viewpoint.

In this paper we propose *PreVR*, a method to provide a VR user with previews around *any number* of corners, as needed to gain line of sight to any part of the VE, no matter how distant or how heavily occluded it is. *PreVR* relies on a multiperspective visualization that implements a higher-order disocclusion effect with piecewise linear rays that bend multiple times as needed to reach the visualization target. The disocclusion capability of *PreVR* is adjustable, which allows investigating the VR interface design continuum from "no preview and intensive navigation", to "preview of nearby VE parts and less navigation", to "preview of distant VE parts and little navigation", all the way to "preview at any distance and no navigation".

Fig. 1 illustrates our *PreVR* visualization method for various values of the maximum disocclusion order $\delta$. A conventional visualization ($\delta = 0$) does not show the target, and a *PreVR* visualization with $\delta = 1$ is sufficient to allow the user to preview the target around the corner. For $\delta = 3$ the three frustum bends are sufficient to reach the target. *PreVR* supports any number of frustum bends to reveal to the user heavily occluded targets ($\delta = 9$), from the user's current location, without any navigation. We also refer the reader to the accompanying video.

We have conducted a user study (N = 88) that investigates four points on the VR interface design continuum defined by the maximum disocclusion order $\delta$. We used a between-subject design with 22 participants for each of four conditions. In a first control condition CC0, $\delta = 0$, which corresponds to conventional VR exploration with no preview capability. In a second control condition CC1, $\delta = 1$, which corresponds to the prior art approach of giving the user a preview around the first corner. In a first experimental condition EC3, $\delta = 3$, so *PreVR* provided up to third-order disocclusion. In a second experimental condition ECN, $\delta$ was not capped, so *PreVR* provided a disocclusion effect of any order, as needed to reach any location in the VE. Participants performed three types of tasks: searching for a stationary target, for a dynamic target moving on a random but continuous trajectory, and for a transient dynamic target that appeared at random locations in the maze and disappeared 5s later. The study quantified VE exploration efficiency with four metrics: viewpoint translation, view direction rotation, number of teleportations, and time to task completion. The results show that the previews afforded by *PreVR* bring a significant VE exploration efficiency advantage. EC3 frequently outperforms CC1 and CC0, and ECN outperforms EC3, CC1, and CC0 for all metrics and all tasks.

*PreVR* is suitable for VE exploration in a number of scenarios. In one scenario, the target of interest, possibly dynamic, is known to the user and *PreVR* provides the user with sustained line-of-sight to the target. In a second scenario, a set of candidate targets are known to the application and the user can decide to inspect them one at the time, from the current location, with *PreVR* providing line-of-sight to each one of them. In a third scenario there are no known candidate targets and *PreVR* allows the user to browse their neighborhood, for example up to a user chosen distance $r$, or up to a user chosen disocclusion order $\delta$. Please refer to the video which illustrates all three scenarios.

In summary, our paper contributes (1) a method for previewing a virtual environment around any number of corners, and (2) a first systematic investigation of the navigation intensity - visualization power trade-off, which was carried out in a large user study.

## 2 Prior Work

We review prior work on VE exploration, on occlusion management in visualization, and on multiperspective visualization in VR.

We now have all-in-one $300 VR headsets [1] with on board power, rendering, and tracking, allowing for an untethered exploration of large VEs. Exploring the VE with a one-to-one mapping between the virtual and the physical space can provide a powerful immersive experience [32]. However, it is often the case that the VE is far larger than the physical space available to host the VR application. The problem has been addressed from several directions. One approach is to rely on a treadmill to keep the user in place in the physical space while the user navigates the virtual space [34]. Another approach is to redirect the user's walking through translational and rotational gains designed to steer the user away from the boundaries or obstacles of the physical world [20, 29, 30]. The goal is to find gains that are large enough to be effective yet small enough to not be detected by the user, for example by taking advantage of saccades [38]. Large VEs can also be folded to fit into smaller physical spaces, either as a preprocess [39], or as the user navigates the VE [36]. Folding flexibility has been gained by allowing for inconsistencies in the modified VE as long as these were not visible to the user [37].

Researchers have also considered abandoning the design constraint of hiding the VE manipulation from the user, as overt manipulations greatly increase the flexibility of the approach and are acceptable as long as the user can still perform the task at hand comfortably, safely, and efficiently. For example Space Bender [33] is an approach that allows users to navigate room-sized virtual environments by bending the environment geometry as the user approaches a physical boundary. The approach embraces the overt nature of the manipulation, instead of hiding it, with user acceptability and task efficiency confirmed empirically. *PreVR* takes the same approach of overt manipulation, which occurs in visualization (i.e., ray) space and not in the geometry space.

When the VE is substantially larger than the available physical space, the application has to allow the user to move temporarily in the virtual space without moving in the physical space. Flying through the VE maintains visualization continuity, but it increases the chance of cybersickness [10, 21]. The risk of cybersickness is reduced when the user moves abruptly between VE locations through teleportation, for example by pointing at the desired destination with a virtual laser pointer [7]. Teleportation comes at the cost of reducing the user's global spatial awareness [10].

Compared to this body of prior work, *PreVR* increases VE exploration efficiency by allowing the user to preview parts of the VE that are hidden from the user's current location. When the preview reveals a part of the VE that is of no interest, the user does not have to navigate to it, avoiding any unnecessary teleportations that this would entail. The VE is distorted during the preview as needed to establish line of sight, but not during actual navigation. Furthermore, during actual navigation, the user physical world motion is mapped one-to-one to their virtual world motion, without redirection. *PreVR* does not attempt to hide from the user the preview capabilities that go beyond those available in the real world. *PreVR* works by alleviating the occlusions inherent to 3D virtual environments.

Occlusions have long been a concern in the visualization of 3D scenes [14]. One approach for alleviating occlusions is transparency [3], which is limited to one or a few occluding layers. Cutaway or exploded view visualization can remove any number of layers to reach the target [9, 23]. Cutaway fails to show the occluding layers, which some applications need for context, and exploded view obscures the connec-

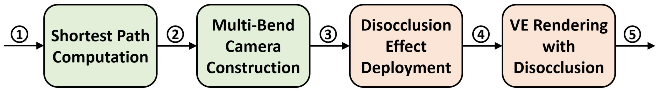

①
Shortest Path Computation
②
Multi-Bend Camera Construction
③
Disocclusion Effect Deployment
④
VE Rendering with Disocclusion
⑤

Fig. 2: Overview of the *PreVR* pipeline. The first two stages (green) are run once per disocclusion effect, and the last two stages (red) are run for each frame.

tion between the parts it visualizes. Another approach is to show the user multiple views of the 3D environment. In picture-in-picture a main perspective is enhanced with an auxiliary perspective [24], with the disadvantage of visualization discontinuity and redundancy between the two perspectives. The user has to switch focus periodically from the main perspective to the auxiliary perspective, and then back, which can be cognitively taxing. Another approach is to suggest to the user the path to take to establish line of sight to the region of interest, for example using arrows and guardrails [2].

Multiperspective visualization improves the integration of multiple perspectives, alleviating visualization discontinuity and redundancy [27, 43]. Multiperspective visualization relies on additional viewpoints that are dispatched through the scene to gather samples not visible from the user viewpoint. The visualization comprehensiveness benefit comes at the cost of output images that are not familiar to the user.

Multiperspective visualization has been used to manage occlusions in VR [41] and AR [22]. When deploying multiperspective visualization to VR, one concern is cybersickness, which is mitigated by limiting the visualization modification to parts of the VE that are not in the user's immediate vicinity [42]. This way, the parts of the VE close to the user are visualized as usual, from the user's perspective, anchoring the user. Another concern is that the multiperspective visualization perturbs global spatial relationships in the VE, hindering spatial awareness. Prior work on multiperspective visualization in VR [41,42] limits previews to the first corner, so a third concern is that the power of the disocclusion effect is not sufficient to reach the target in a complex VE.

*PreVR* leverages multiperspective visualization to allow the user of a VR application to preview hidden parts of the VE. The disocclusion effect is constructed based on the graph camera [27]. Like prior work, *PreVR* mitigates cybersickness by anchoring the user. Unlike prior work, the user triggers the automatic deployment of the disocclusion effect using a glyph, and can examine the region of interest from the current position. Furthermore, *PreVR* does not limit previews to the first corner, affording previews around any number of corners.

## 3 PreVR Higher-Order Disocclusion

Fig. 2 shows the four stages of the *PreVR* pipeline. *The first stage* takes as input (1) the user position $U$, the target position $T$, and the VE geometry, and computes the shortest path between $U$ and $T$ that does not intersect the VE geometry. We compute a piecewise linear shortest path using the $A^*$ algorithm [17]. The shortest path is fed to the next stage (2). The target position $T$ is defined based on the VE exploration scenario of the current VR application. If the system knows the target that the user is interested in, then $T$ is the centroid of the target object; this is the case, for example, in a scenario where the user has tagged a previously seen object as being of interest, or where the user can formulate a query defining the target. If the system knows a list of candidate objects of interest, then $T$ can identify any object from the list; this is the case, for example, in a scenario where the user might want to see an avatar of another user in the case of a collaborative VR application. When the system has no indication as to what might be of interest to the user, $T$ can be defined as any empty space point; this is the case, for example, in a scenario where the user wants to browse the region of the VE that is close to them.

*The second stage* constructs a multiperspective camera based on the path. We use a graph camera [27] with a linear sequence of frustum bending operations, i.e., a multi-bend camera. The construction is described in Sec. 3.1. If the maximum disocclusion order $\delta$ is smaller than the number of path segments, no multi-bend camera can be constructed and the user has to navigate using a conventional visualization to get closer to the target. If the construction is successful, i.e., if the target is within the disocclusion capability of the *PreVR* visualization, the multi-bend camera is fed to the next stage (3) where the disocclusion effect is deployed.

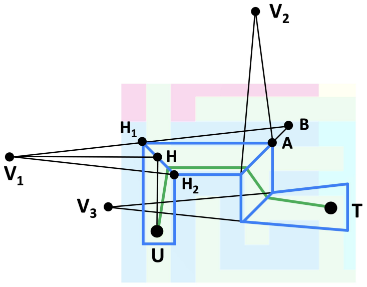


Fig. 3: Construction of multi-bend camera that lets the user $U$ see the target $T$. The camera is constructed from the shortest path between $U$ and $T$ (green) and has three bends, three secondary viewpoints $V_1$, $V_2$ and $V_3$, and four sub-frusta (blue).

*The third stage* deploys the disocclusion effect gradually, by morphing the conventional camera into the multi-bend camera over several frames. The morph proceeds one bend at the time, starting from the user and ending at the target. After the last bend is deployed, the view is zoomed in to increase the visualization footprint of the target. The disocclusion effect deployment is described in Sec. 3.2. Once the multi-bend camera for the current frame is finalized, the camera is fed to the next stage that renders the VE (4).

*The fourth stage* renders the *PreVR* visualization for each user eye using the multi-bend camera, as described in Sec. 3.3. The *PreVR* visualization allows the user to preview the target from the current position, without any navigation.

We describe stages 2, 3, and 4 in detail in the following subsections, using a maze with 90° angles. However, *PreVR* is general and supports disocclusion along any piecewise linear path, as illustrated in Sec. 4, see Figs. 14, 15, and 16.

### 3.1 Multi-Bend Camera Construction

Given a piecewise linear shortest path between the user and the target, the multi-bend camera is constructed automatically starting from the user's conventional camera and bending its frustum according to the path to disocclude the target. In Fig. 3 the construction starts at $U$, the frustum is bent to the right, introducing the secondary viewpoint $V_1$, again to the right, introducing $V_2$, and finally left, introducing $V_3$. A bending operation is controlled by a hinge plane, e.g., $H_1H_2$ for the first bend. The hinge plane is vertical and it bisects the angle of the two path segments that define the bend. Each bending operation defines a new conventional planar pinhole camera sub-frustum. The sub-frustum uses the current hinge plane as the near clipping plane and the next hinge plane as the far clipping plane. The secondary viewpoint is placed on the reflection of the previous view direction on the hinge plane, e.g., $V_1$ is placed on the reflection $V_1H$ of $UH$ on $H_1H_2$. The distance from the secondary viewpoint to the hinge plane is determined based on an initial desired field of view, which can be modified interactively as described below. The sub-frusta become wider with each bending operation, the same way the frustum of a conventional camera grows wider with the distance from the user viewpoint. To avoid that sub-frusta intersect, their size is adjusted with four clipping planes. For example, in Fig. 3, the left side wall of the sub-frustum $V_1$ is $H_1A$ and not $H_1B$.

### 3.2 Disocclusion Effect Deployment

Once the multi-bend camera is constructed, the disocclusion effect is deployed gradually by morphing the conventional camera into the multi-bend camera. The morph deploys one sub-frustum at the time, starting from the first sub-frustum at the user and ending with the last sub-frustum at the target. Fig. 4 illustrates the deployment of a sub-frustum by translating its viewpoint from the previous sub-frustum viewpoint to

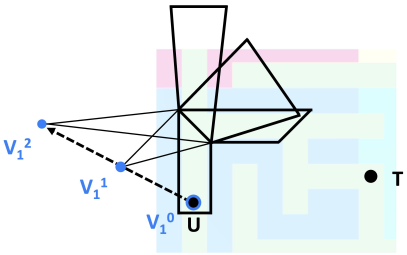


Fig. 4: Gradual sub-frustum deployment by translating the secondary viewpoint $V_1$ from $V_1^0$ ($U$) to its final position $V_1^2$.

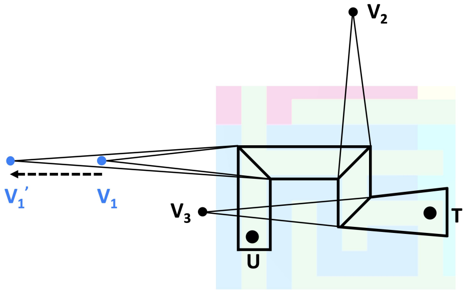


Fig. 6: Zoom-in effect implemented by backward secondary viewpoint translation. The process is applied to all secondary viewpoints simultaneously (not shown here for figure clarity).

the final position of the viewpoint prescribed by the multi-bend camera construction. The first, intermediate, and last frames are shown in Fig. 5. The first sub-frustum is the already deployed conventional user view. The VE geometry in this first sub-frustum continues to be rendered conventionally, based on the user viewpoint, anchoring the user [42]. The second and following sub-frusta are each deployed over several frames. The farther the sub-frustum from the user, the smaller its output frame footprint, and the faster it can be deployed. This whipping effect of the deployment of the "tunnel" from the user to the target attempts to minimize deployment time while avoiding potentially confusing fast changes over a large part of the output frame.

After all sub-frusta are deployed, an optional zoom-in effect magnifies the target for long multi-bend cameras. The zoom-in effect is implemented by pulling back all secondary viewpoints simultaneously (Fig. 6). Since the image frame stays the same, the backward viewpoint translation amounts to a zoom-in effect, as shown in Fig. 7.

### 3.3 VE Rendering

Once the multi-bend camera is finalized for the current frame, the VE is rendered conventionally, except for a modification of the vertex projection. Multi-bend vertex projection starts with finding which sub-frustum contains the given vertex. We search for the sub-frustum with a linear pass over the array of sub-frusta. This is sufficiently fast since the number of sub-frusta is small (typically below 10), and since one can quickly check whether the vertex is inside a sub-frustum by computing the six dot products that evaluate the sidedness of the vertex with respect to the six faces of the sub-frustum. Once the sub-frustum $S_i$ containing the vertex is known, the vertex is projected with all the sub-frusta from $S_i$ to $S_1$. In Fig. 8, $T$ is inside the last sub-frustum and it is projected to $T_3$ with $V_3$, then $T_3$ is projected to $T_2$ with $V_2$, and $T_2$ is projected to $T_1$ with $V_1$. $T_1$ is then pushed back to $T'$ along its ray from $U$, at a distance equal to the length of the piecewise linear ray $UT_1 + T_1T_2 + T_2T_3 + T_3T$. Finally, $T'$ is projected conventionally for the user's left and right eyes. Placing the displaced vertex at a distance commensurate to the distance that the user ray travels to reach the vertex generates appropriate left/right frame disparity, as needed for stereoscopic depth perception. The disocclusion effect is updated every frame, supporting fully dynamic VE's, with the multi-bend camera projection operating on vertices after animation.

## 4 User Study

We have conducted a user study to investigate VE exploration efficiency at four points on the interface design continuum defined by the maximum disocclusion order $\delta$. Our study was conducted with the approval of our Institutional Review Board.

### 4.1 Methods

**Participants.** We have recruited N = 88 participants from the undergraduate and graduate student population of our university. The average age was 20.6 years, and a single participant was over 30 years old. 28 participants self-identified as female, 58 as male, and 2 checked the "no answer" box. 20 participants indicated that they had never used VR before, 29 had used VR once, 33 occasionally, and 6 frequently. The participants were compensated for their time with a $20 (USD).

**Study design.** The study investigated four conditions corresponding to four values of the maximum disocclusion order $\delta$. The difference between the VR interfaces is given by the value of the maximum disocclusion order $\delta$. There are two control conditions and two experimental conditions. For the first control condition (CC0) $\delta = 0$, which corresponds to conventional VR exploration without the possibility of previewing hidden parts of the VE. For the second control condition (CC1) $\delta = 1$, which corresponds to the prior art approach of allowing the user to look around the first corner [42]. The two experimental conditions investigate two variants of *PreVR*: for the first experimental condition (EC3) $\delta = 3$, and for the second experimental condition (ECN) $\delta$ is not limited.

We opted for a between-subjects design where the participants were randomly split into four equal groups, and each group was assigned to one of the four conditions. The between-subjects design requires a larger number of participants but it brings the benefit of avoiding learning effects when the same participant performs tasks in all conditions. We anticipated large effect sizes between conditions, as the preview should considerably reduce the amount of unnecessary navigation. The 22 participants per condition allow for finding large/extra-large effects (i.e., Cohen's $d = 1.0$) with 0.90 power, for $\alpha = 0.05$.

**Tasks.** Each participant performed three tasks in an abstract VE with a 32 m x 32 m 3D maze (Fig. 1). The maze corridors are 2 m wide and the walls are 2 m tall. the first task deals with stationary targets, the second with dynamic targets moving continuously, and the third with dynamic targets jumping between locations.

In *Task 1* the participant had to find a target cylinder with a specific letter. The maze contained three stationary cylinders 1.2 m tall and 0.8 m wide. The cylinders had different colors. One cylinder was marked with letter "A", one with "B", and one with "C" (Fig. 7). The cylinders were placed at random locations in the maze, with one cylinder being

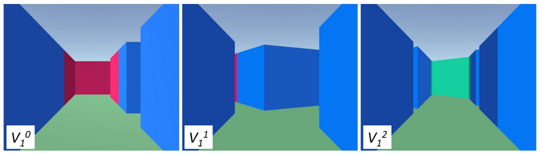


Fig. 5: User frames for the sub-frustum deployment from Fig. 4.

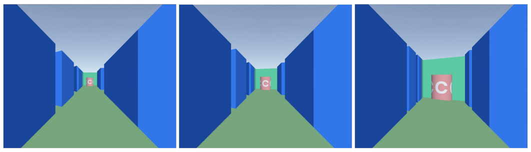


Fig. 7: Deployment of zoom-in effect.

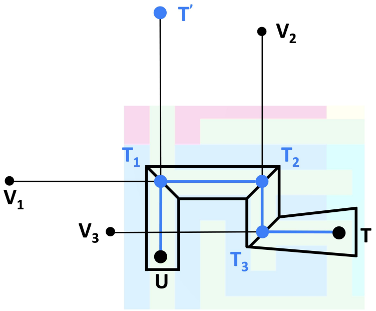


Fig. 8: Multi-bend camera projection of $T$. $T$ is projected successively with $V_3$,$V_2$ and $V_1$ to $T_3$, $T_2$, and $T_1$, $T_1$ is pushed back to $T'$, and $T'$ is projected conventionally from $U$.

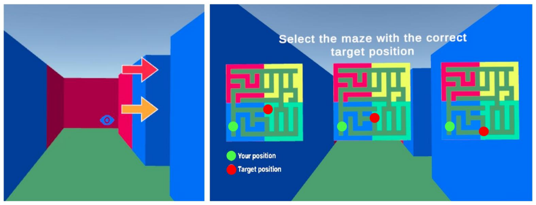


Fig. 9: *Left*: user hints for EC3; the blue cylinder is within the disocclusion capability of the visualization, and the availability of the preview is indicated with the eye icon; the red and orange cylinders are too far to be previewed from the current location and the arrows point to the shortest path towards each cylinder. *Right*: spatial orientation test SOT2; the user is asked to choose the map that correctly shows the position of the target they had just found.

“near” the user’s starting position, i.e., reachable with three turns or fewer, and with two cylinders being “far”, i.e., reachable with three bends or more. For half the *Task 1* trials the target cylinder was “near” and for the other half the target was “far”.

In *Task 2* the participant had to find a dynamic cylinder. There was a single cylinder in the maze, moving on a continuous but random trajectory. The target cylinder had a constant speed and it moved on a straight line in a corridor segment until an intersection. At an intersection, if the target could continue straight it would do so with a 30% probability and turn randomly with a 70% probability. If the target could not continue straight it would make a random turn.

In *Task 3* the participant had to find a transient cylinder. There was a single cylinder in the maze. The target cylinder appeared at one location in the maze, the target stayed visible at that location for 5s, then the target disappeared to reappear at a different location. The locations were chosen randomly within the maze, with the requirement that the target switches maze quarters for each appearance. For example if the target was in the red quarter of the maze (Fig. 1) it had to reappear in the yellow, green, or blue quarters.

**Spatial orientation tests.** After each trial of *Task 1*, the participant was taken back to the starting position and was asked to complete two spatial orientation tests, SOT1 and SOT2. For SOT1, the participant was asked to point in the direction of the target they had just found. For SOT2, the participant was shown three 2D maps of the maze that showed their initial position and the position of the target they had just found (Fig. 9, right). The target position was correct only in one of the three maps, and the participant was asked to pick the correct map. The maze floor is of uniform color. The maze walls are colored based on the quarter of the maze where they are located. We have chosen this color pattern as a trade-off. A more significant color variation within the maze would help participants recognize the location of the task based on color, and without actually building an internal map of the VE. On the other hand, a maze of uniform would make internalizing the VE spatial layout too difficult.

**Implementation.** Our application was developed using the Unity Engine (2021.3.91f) [40] and was run on an Oculus Quest 2 with two handheld controllers. We find the shortest path using Unity’s implementation of the A* path-finding algorithm. The *PreVR* projection is implemented in a geometry shader that discards triangles outside the multi-bend frustum. The average frame rates for maximum disocclusion order $\delta$ values from 1 to 9 are 71.98, 71.85, 72.01, 72.04, 72.05, 71.49, 70.55, 69.90, and 68.76 frames per second. The frame rate hovers around the Quest 2’s maximum frame rate of 72 fps until $\delta = 5$, and then decreases slightly as $\delta$ increases from 6 to 9. The frame rate is 68.76 fps even for a deep preview with $\delta = 9$.

**Experimental procedure.** A participant first filled out an eligibility form, a consent form, and a biographical questionnaire (age, gender, VR experience). Then the participant moved to a 2 m x 2 m designated area of our laboratory and put on the VR headset. The participant picked up a controller in each hand. Attached to the right hand controller was a virtual laser pointer with which the participant collects the target cylinders. Attached to the left hand controller was a text box that reminded the participant what they have to do for the current task, e.g., “Find cylinder A” for *Task 1*, “Capture the moving cylinder” for *Tasks 2*, and “Capture the randomly appearing cylinder” for *Task 3*. The participant first performed three practice trials for *Task 1*, then four trials for *Task 1*, each followed by the two spatial orientation tests, then four trials for *Task 2*, and finally four trials for *Task 3*. Before each trial, the participant was shown instructions in front of them, and the trial started once the participant clicked “Next”. A participant completed the trials in 10 min on average, and in at most 20 min.

Once the tasks were complete, the participant removed the VR headset and filled out a system usability scale questionnaire (SUS) [8], and a simulator sickness questionnaire (SSQ) [18]. There are two approaches for administering the SSQ: pre- and post-exposure, or just post-exposure. Both approaches have advantages and disadvantages [5]. The post-only approach has the advantage of not artificially raising participant awareness of SSQ symptoms before the exposure, and the disadvantage of not establishing a baseline. Whereas the pre- and post-approach is unavoidable in a single-session within-subjects design, we are taking advantage of our between-subjects design and use the post-only approach. The total participation time was 30 min.

Participants navigated through the maze by rotating their head and their body to change view direction, and by teleporting to translate the viewpoint. Participants were given two types of hints placed at maze intersections: arrow and eye icons (Fig. 9, *left*). An arrow icon indicates the first turn on the shortest path to a cylinder. If the participant clicks on the arrow they are teleported to the maze intersection where the arrow is located. An eye icon indicates the availability of a preview to a cylinder. If the participant clicks on the eye icon, the disocclusion effect is deployed showing the participant the cylinder from their current location. For CC0 there were only arrow hints. For CC1 there was an arrow if the cylinder was beyond a single bend, and an eye if the cylinder was reachable with one bend. For CC3 there was an arrow if the cylinder was beyond three bends, and an eye if the cylinder was reachable with three bends. For CCN there were no arrow hints and only eye hints, as all cylinders were always reachable. The hints avoid blind searching, which would make the user study intractable. The lower the $\delta$ value, the more the hints are beneficial, so any advantage of higher $\delta$ values found in the study will be even larger when no hints are used. The hints do preserve the sequential nature of conventional exploration as an arrow only teleports the participant to a location within their line-of-sight, as needed for a valid comparison between lower and higher $\delta$ values.

Participants always started a trial from the same position and with the same orientation (i.e., green dot in Fig. 1, looking towards the red corner of the maze). Once they found a target, the trial ended, and the participant was brought back to the starting position, but not reoriented. Instead, the application asked the participant to turn left or right to assume the initial orientation, preserving the participant’s spatial orientation between trials.

**Data collection.** The application records for each trial the total translation through teleportation, in meters, the total view direction rotation, in degrees, the number of teleportations, and the time the

Table 1: Task 1 analysis of the differences between the four conditions for each of the four navigation efficiency metrics.

| | Kruskall-Wallis $p$ | Dunn's posthoc $p$ | | | | | |
|---|---|---|---|---|---|---|---|
| | CC0 vs CC1 vs EC3 vs ECN | CC0 vs CC1 | CC0 vs EC3 | CC0 vs ECN | CC1 vs EC3 | CC1 vs ECN | EC3 vs ECN |
| Translation | $p$<0.001*, $\chi^2(3) = 232.5$ | 1.0 | <0.001* | <0.001* | <0.001* | <0.001* | <0.001* |
| Rotation | $p$<0.001*, $\chi^2(3) = 194.7$ | 1.0 | <0.001* | <0.001* | <0.001* | <0.001* | <0.001* |
| Teleportations | $p$<0.001*, $\chi^2(3) = 247.8$ | 1.0 | <0.001* | <0.001* | <0.001* | <0.001* | <0.001* |
| Time | $p$<0.001*, $\chi^2(3) = 64.2$ | 0.061 | 1.0 | <0.001* | 0.001* | <0.001* | 0.005* |

participant needs to capture the target, in seconds. For *Task 1* the application also records the pointing direction error in degrees for SOT1 and the correctness of the answer for SOT2. The pointing direction error is measured in the horizontal plane as the angle between the pointing direction and the direction to center of the target. We also collected subjective data from through the SUS and the SSQ questionnaires.

**Data analysis.** We analyzed each of the five continuous dependent variables (translation, rotation, number of teleportations, time, and pointing direction error) for each of the three tasks over the four conditions. We used box plots and statistical tests. We used the non-parametric Kruskal-Wallis H test [19], as our between-subject data was not normally distributed. When the Kruskal-Wallis H test showed a significant difference between the four conditions, we used Dunn's post-hoc test [13] to investigate pairwise differences, with a ×6 Bonferroni correction to account for the six pairs of conditions [6]. We analyzed the dichotomous dependent variable (correct answers for SOT2) using the Chi-squared test [16]. We used a significance level $\alpha = 0.05$ for all statistical tests. We also measured the effect size using Cohen's $d$ [11]. We labeled the effects as "small" ($d > 0.2$), "medium" ($d > 0.5$), "large" ($d > 0.8$), "very large" ($d > 1.2$), and "huge" ($d > 2.0$) [31].

## 4.2 Results and Discussion

We present and discuss results for each of the three tasks, then for the spatial orientation tests, and then for the user study as a whole.

### 4.2.1 Task1: searching for a stationary target

The navigation efficiency results are given in Fig. 10. A boxplot gives the 25% to 75% range (thick bar), the entire range (whiskers), the average (small triangle), the median (white horizontal line), and outliers (small circles). Outliers are defined as data points below Q1 - 1.5IQR and above Q3 + 1.5IQR, where Q1 and Q3 are the first and third interquartiles and IQR = Q3 - Q1. The conditions are compared for each metric in Tab. 1.

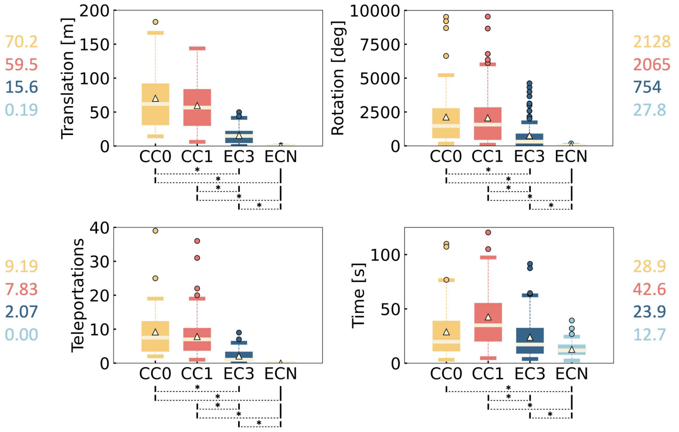


Fig. 10: Task 1 navigation efficiency for four metrics (translation, rotation, teleportations, and time) and four conditions (CC0, CC1, EC3, and ECN). Pairwise significant differences are shown with a horizontal bracket and an asterisk. Mean values are also given numerically.

The experimental conditions (EC3 and ECN) consistently outperform the control conditions (CC0 and CC1). As expected, ECN requires no teleportations, and it hardly requires any viewpoint translation or view direction rotation. All four pairwise comparisons between a control and an experimental condition, i.e., CC0 vs EC3, CC0 vs ECN, CC1 vs EC3, and CC1 vs ECN, reveal a significant navigation efficiency advantage for the experimental condition for all four metrics. Out of the 16 CC vs EC comparisons, the only exception is CC0 vs EC3 for time, where the advantage of EC3 (average time 23.9s) is not significant over CC0 (average time 28.9s). This is due to the fact that participants navigate exclusively via instant teleportation, whereas the disocclusion effect is deployed gradually. Therefore previewing over three corners is not significantly faster than teleporting three times. If participants actually walked through the maze, the time differences would most likely be larger and significant.

Between the two control conditions, CC1 does not have a significant advantage over CC0 for any of the four metrics, which indicates that seeing around a single corner is not sufficient for significant navigation efficiency gains. The CC1 averages are better (lower) than the CC0 averages except for time, where deploying the preview is slower than a single instant teleportation hop. Between the two experimental conditions, ECN always has a significant advantage over EC3, for any of the four metrics. This indicates that deeper previews lead to significant navigation efficiency gains.

Tab. 2 shows that the effect sizes are mostly "very large" and above. This indicates that our 22 participants per condition were sufficient for establishing significance at the $\alpha = 0.05$ level with ample statistical power, i.e., over 95%.

### 4.2.2 Task 2: searching for a moving target

The task 2 navigation efficiency results are given in Fig. 11, and the comparison between conditions is given in Tab. 3. EC3 and ECN have a significant advantage over CC0 and CC1 in terms of translation and teleportation. This is the case because both *PreVR* methods replace the viewpoint translation through teleportation with previews from the current location. Whereas ECN also has a significant advantage over CC0 and CC1 in terms of rotation and time, EC3 does not. In terms of time, EC3 *is not better* than CC0 or CC1. This is due to the fact that the unbounded preview depth of ECN allows the visualization to track the dynamic target, while with the limited preview depth of EC3, the dynamic target can move out of reach. When that happens, the disocclusion effect has to be retracted, the participant has to navigate closer using teleportation, and then redeploy an updated disocclusion effect when the target is again within three bends of the participant. In terms of rotation, EC3 has a smaller average rotation (759°) than CC0 (814°) or CC1 (919°), but the advantage over CC0 is not significant.

Table 2: Task 1 Cohen's $d$ effect sizes.

| | CC0 vs ECN | | CC1 vs ECN | |
|---|---|---|---|---|
| Trans | 1.50 | *very large* | 1.99 | *huge* |
| Rot | 1.27 | *very large* | 1.34 | *very large* |
| Tel | 1.46 | *very large* | 1.87 | *very large* |
| Time | 0.79 | *large* | 1.18 | *very large* |

Table 3: Task 2 analysis of the differences between the four conditions for each of the four navigation efficiency metrics.

| | Kruskall-Wallis $p$ | Dunn's posthoc $p$ | | | | | |
|---|---|---|---|---|---|---|---|
| | CC0 vs CC1 vs EC3 vs ECN | CC0 vs CC1 | CC0 vs EC3 | CC0 vs ECN | CC1 vs EC3 | CC1 vs ECN | EC3 vs ECN |
| Translation | $p$<0.001*, $\chi^2(3)$=223.7 | 1.0 | <0.001* | <0.001* | <0.001* | <0.001* | <0.001* |
| Rotation | $p$<0.001*, $\chi^2(3)$=155.3 | 1.0 | 0.302 | <0.001* | 0.046* | <0.001* | <0.001* |
| Teleportations | $p$<0.001*, $\chi^2(3)$=205.9 | 1.0 | <0.001* | <0.001* | <0.001* | <0.001* | <0.001* |
| Time | $p$<0.001*, $\chi^2(3)$=108.8 | 0.628 | 1.0 | <0.001* | 0.726 | <0.001* | <0.001* |

This is due to the fact that when a preview misses the target because the target moves out of reach, the participant has to rotate the view in search of the arrow icon to move towards finding the evading target, which could amount to repeated view direction rotations from the same location. With the control conditions, selecting an arrow always moves the participant closer to the target, and no view rotation ever goes to waste. Like for task 1, there are no differences between CC0 and CC1, and ECN has a significant advantage over EC3 for all four metrics.

### 4.2.3 Task 3: searching for a transient target

The task 3 navigation efficiency results are given in Fig. 12, and the comparison between conditions is given in Tab. 4. Like for the other two tasks, EC3 and ECN have a significant advantage over CC0 and CC1 in terms of translation and teleportation. Like for the other tasks, ECN also has a significant advantage over CC0 and CC1 in terms of rotation and time. Like for task 2, EC3 has no advantage over CC0 or CC1 in terms of time or rotation. However, unlike for task 2, EC3 does not have an advantage over CC0 and CC1 in terms of rotation. The difference is that for task 2 the target moves on a continuous trajectory, whereas for task 3 the target jumps abruptly between distant locations. For task 2 the icon of an unsuccessful preview is likely to change in place into an arrow, which does not incur the cost of additional view rotation. For task 3 however, due to the large change in target location, the arrow icon is likely to appear at a different location which requires the participant to rotate the view. Like for the previous two tasks, there are no differences between CC0 and CC1, and ECN is significantly better than EC3 for all metrics.

### 4.2.4 SOT1 and SOT2: spatial orientation tests

*SOT1.* Fig. 13 gives the results of the spatial orientation tests. When asked to point in the direction of the target they had just found, participants do so with average errors of 37° for CC0, 45° for CC1, 48° for EC3, and 46° for ECN. The differences between conditions are not significant (Kruskal-Wallis test $\chi^2(3) = 5.97, p = 0.11$). The average errors are lower for CC0, and similar for CC1, EC3, and ECN. These errors are large in an absolute sense, especially since the target was always in the north-east quadrant (12 o'clock to 3 o'clock) with respect to the user's initial position (the green dot at the bottom left of the square maze in Fig. 1). Consistently pointing in the north-east direction would have resulted in smaller average errors. In conventional navigation, the many turns the user has to take, including to retrace their path, amount to a loss of global spatial orientation. Similarly, the turns the visualization rays take to disocclude also lead to a loss of global spatial orientation. The participants did not know the direction of the target and often pointed forward to move on to the next trial.

*SOT2.* When asked to identify the map with the correct target location, participants did so with an accuracy between 61% and 48% for the four conditions. Since three maps were shown, chance performance corresponds to 33%, which is exceeded by all conditions. We explain the overall better performance for SOT2 than for SOT1 by the fact that the correct answer for SOT2 can be inferred by counting the number of turns taken to reach the target. Indeed, we have noticed participants counting out aloud the turns during the exploration of the maze. Then, when presented with the three maps, participants count the number of turns required for each map to choose the correct answer. The control conditions have an advantage over the experimental conditions, but the difference is not significant (Chi-squared test $\chi^2(3) = 4.293, p = 0.232$). This is due to the fact that it is harder to count the number of turns when a deep disocclusion effect is deployed, as the turns are taken quickly. The counting of turns bypasses the need of building an accurate mental map of the virtual environment, so SOT2 is not as accurate of a spatial orientation test as SOT1.

### 4.2.5 SSQ and SUS: cybersickness and usability

**Cybersickness.** The average total SSQ scores (TS) are 23.6 for CC0, 17.7 for CC1, 17.3 for EC3, and 14.4 for ECN. There is no significant difference between the four conditions (Kruskall-Wallis $\chi^2(3) = 3.124, p = 0.37$). The TS values decrease in the order CC0 > CC1 > EC3 > ECN, and correlate with the number of teleportations. Interpreting the absolute TS values is difficult. For the military pilot population TS scores in the [10, 15] range point to "significant symptoms", and TS scores in the [15, 20] range point to "symptoms [that] are a concern" [35]. However, VR research frequently reports scores above 20 [12, 15, 25, 28]. Furthermore, the baseline TS score

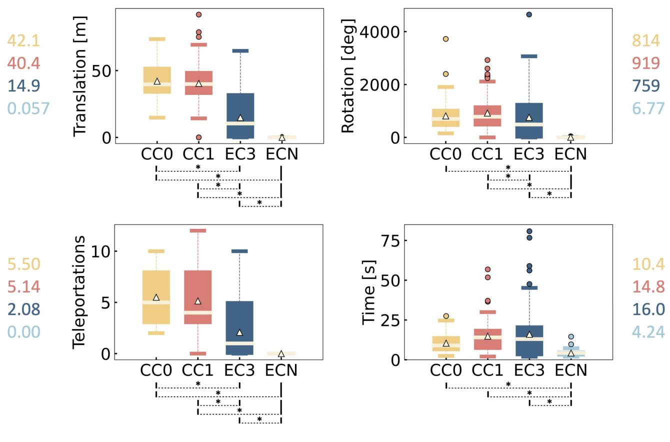


Fig. 11: Task 2 navigation efficiency.

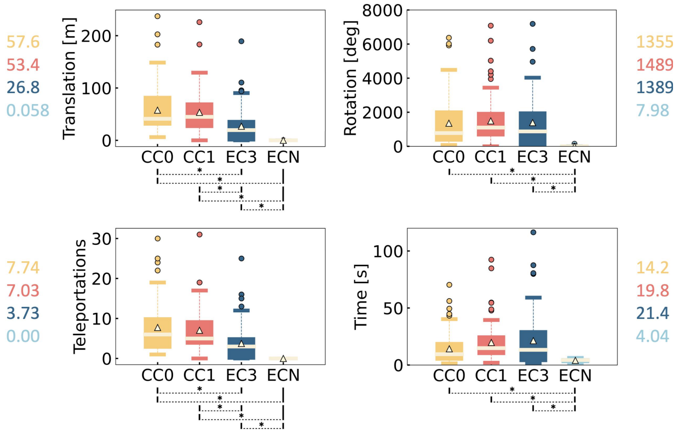


Fig. 12: Task 3 navigation efficiency.

Table 4: Task 3 analysis of the differences between the four conditions for each of the four navigation efficiency metrics.

| | Kruskall-Wallis $p$ | Dunn's posthoc $p$ | | | | | |
|---|---|---|---|---|---|---|---|
| | CC0 vs CC1 vs EC3 vs ECN | CC0 vs CC1 | CC0 vs EC3 | CC0 vs ECN | CC1 vs EC3 | CC1 vs ECN | EC3 vs ECN |
| Translation | $p$<0.001*, $\chi^2(3)$=191.0 | 1.0 | <0.001* | <0.001* | <0.001* | <0.001* | <0.001* |
| Rotation | $p$<0.001*, $\chi^2(3)$=169.6 | 1.0 | 3.039 | <0.001* | 0.301 | <0.001* | <0.001* |
| Teleportations | $p$<0.001*, $\chi^2(3)$=193.1 | 1.0 | <0.001* | <0.001* | <0.001* | <0.001* | <0.001* |
| Time | $p$<0.001*, $\chi^2(3) = 90.4$ | 0.069 | 1.0 | <0.001* | 0.275 | <0.001* | <0.001* |

is never 0. Our participants were instructed to stop the study at the onset of cybersickness symptoms, and they were assured that they will be fully compensated even if they do not complete the study due to cybersickness. None of our 88 participants had to stop the study early due to cybersickness. In terms of SSQ sub-scores, the same pattern is observed: for CC0, CC1, EC3 and ECN, the average nausea sub-scores N are 13.9, 10.8, 10.4 and 9.54, the average oculomotor sub-scores O are 19.6, 15.8, 17.2, 11.9, and the average Disorientation sub-scores D are 31.6, 20.9, 17.7 and 17.9. The lower SSQ scores for *PreVR* confirm that *PreVR*'s conventional visualization of the part of the VE close to the user, which changes as the user expects it in response to head motions, confining the disocclusion effect to the frame region corresponding to the corridor entrance, and reducing the number of teleportations contribute to anchoring the user, thereby alleviating cybersickness compared to conventional VE navigation.

**Usability.** The overall SUS scores for CC0, CC1, EC3, and ECN are 81.9, 71.2, 66.7, and 73.6, corresponding to "Excellent", "Good", "OK", and "Good", respectively [4]. There is a significant difference between the four conditions (Kruskall-Wallis $\chi^2(3) = 12.76, p = 0.005$). The posthoc analysis (Dunn's test with ×6 Bonferroni correction) finds one pairwise significant difference, i.e., CC0 has a significant usability advantage over EC3 ($p = 0.029$). Our participants have indicated two reasons for their preference of the conventional approach: familiarity with the technique, and frustration with EC3 when the deployment of the disocclusion effect gives the target enough time to escape. The trade-off between a slower and a faster deployment of the disocclusion effect can be further adjusted. A slower deployment can give participants a sense of where the target is located within the VE based on the turns the visualization takes to reach it. However, a slower deployment can also give a dynamic target sufficient time to evade the visualization. A faster deployment is more likely to reveal fast targets, but it can also confuse the user through the abrupt visualization change.

### 4.2.6 User study conclusions

The study results support the following conclusions.

1. Either of the *PreVR* conditions is better than either of the control conditions in terms of viewpoint translation and number of teleportations. The deeper the preview, the larger the savings.

2. ECN is also better than either of the control conditions in terms of view direction rotation and task completion time.

3. EC3 is not better than the control conditions in terms of view direction rotation and task completion time. Since the participant can move quickly through the VE by clicking the arrow icons, previewing over three corners with a gradual deployment of the disocclusion effect is not faster than teleporting around those three corners.

4. ECN is better than EC3 in terms of each of the four metrics. The unbounded disocclusion order allows reaching targets no matter how deeply occluded, bringing VE exploration efficiency gains.

5. The harder the task, the more an unbounded disocclusion order helps. EC3 loses its advantage over the control conditions from task 1 to task 2 and then to task 3, and ECN does not.

6. Spatial orientation is difficult for all conditions. The *PreVR* visualization approach is to be used as a method for helping the user find the interesting parts of the VE more efficiently, through previews that avoid unnecessary navigation. Once the user finds an interesting part of the VE, the system can guide the user to the interesting part, e.g., with arrow icons. In other words, the user doesn't have to find their way to the interesting part from memory.

7. Regarding usability, the conventional approach has an advantage over *PreVR*. Between the conditions that offer preview, the higher the disocclusion order, the higher the usability score.

8. Regarding cybersickness, a conservative conclusion is that *PreVR* does not worsen cybersickness compared to conventional VE exploration. The SSQ score for ECN is 61% lower than that for CC0.

Our user study relied on an abstract VE where the target occlusion level could be controlled as needed to contrast the four conditions. The abstract VE also eliminated confounding factors such as visual clutter that could make a user miss a visible target. *PreVR* is ready to be integrated in VR applications. Figs. 14 and 15 show *PreVR* used in the context of a library and of a house interior VE, for various values of the maximum disocclusion order $\delta$.

## 5 Limitations

One limitation of our approach is that the disocclusion effect requires access to the disocclusion target. For example, *PreVR* cannot offer previews of a room from outside the room if all room doors are closed,

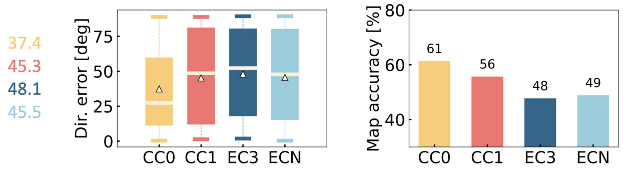


Fig. 13: Pointing direction error and map identification accuracy.

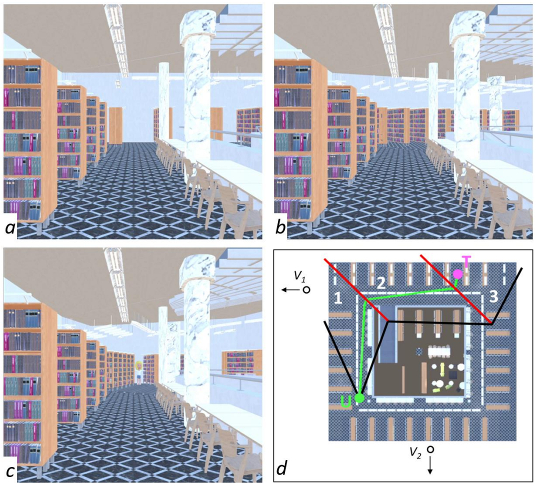


Fig. 14: Library example: output frames (*a*-*c*) and overhead illustration of disocclusion effect construction (*d*). The disocclusion effect is deployed gradually from *a* to *c* to reveal the distant person. In *d*, the shortest path (green) connects the user $U$ to the person $T$, the view frustum bends at two hinge planes (red), resulting in three sub-frusta (1-3), using two additional viewpoints ($V_1$ and $V_2$ not shown at their actual location).

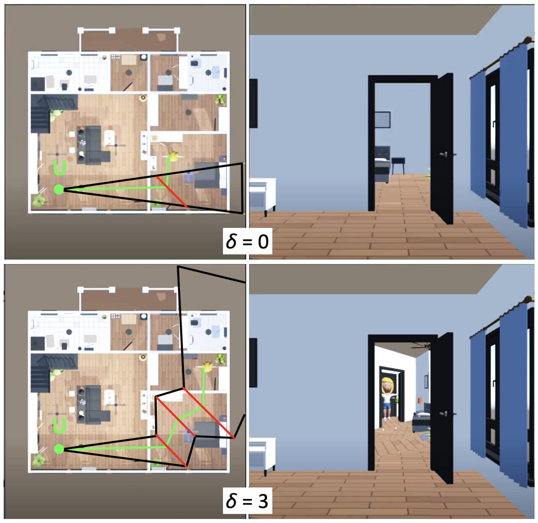


Fig. 15: House interior example: illustration of the disocclusion effect construction (left), and output frame (right).

because there is no space to route the rays to the room of interest. Such scenarios require the preliminary step of creating access, for example by opening a door, or by cutting away or rendering transparently a part of the wall. One of the strengths of our *PreVR* approach is its suitability for use in combination with other occlusion management approaches, allowing the application to benefit from their strengths and to avoid their weaknesses. For example, *PreVR* can break the alignment of two regions of interest, avoiding that they blend together in a transparent visualization, while transparency can remove a wall of no interest, preventing unnecessary complexity in a *PreVR* visualization.

One limitation of our user study is that it only tests the known target or known set of candidate target scenarios, and it does not test the browsing scenario where there are no known candidate targets. We have opted for this study design to limit the time it took a participant to complete the task. Browsing through the entire maze takes a considerable amount of time, especially in the control condition with conventional visualization. In the case of dynamic or transient targets a participant is likely to not find the target in a practical amount of time. *PreVR* does support browsing around any number of corners, by visualizing the user's neighborhood one disocclusion effect at the time.

Another limitation of our current implementation is that the projection of a vertex proceeds sequentially through the list of sub-frusta to find the sub-frustum that contains the vertex, if any. For a small number of sub-frusta, i.e., fewer than 10, like was the case for our examples, this is the most efficient approach. For a larger number of sub-frusta, the sub-frusta will have to be organized based on a hierarchical space partitioning scheme, e.g., a binary space partitioning tree, to accelerate the search for the sub-frustum containing a given vertex.

We construct the disocclusion effect with an abrupt transition from one viewpoint to the next, as rays cross the hinge plane. In other words, our visualization is continuous, but only $C^0$ continuous. The abruptness of the transition is especially visible for textured surfaces, like the floors in Figs. 14 and 15. Future work could examine implementing a gradual transition, e.g., by using shorter piecewise linear segments, or even curved rays, at the transition region. In other words, transitioning from viewpoint $V_i$ to $V_{i+1}$ could be done with $k$ intermediate viewpoints on the segment $V_iV_{i+1}$, which do not have a disocclusion role, but do have the role of implementing the gradual viewpoint change.

*PreVR* allows adjusting the depth of the preview by supporting a variable, non-bounded, application chosen maximum-disocclusion order. The higher the disocclusion order, the more complex the visualization, and the higher the information content of the image. In other words, given an output image resolution, which is fixed and determined by the headset's native resolution, the number of perspectives, i.e., viewpoints, that *PreVR* can show is limited by the minimum image footprint a perspective should be allotted. Although *PreVR* allows adjusting the resolution on potential regions of interest, this comes at the cost of reducing resolution on the context, and when context has to be seen at high resolution, the depth of the preview is limited.

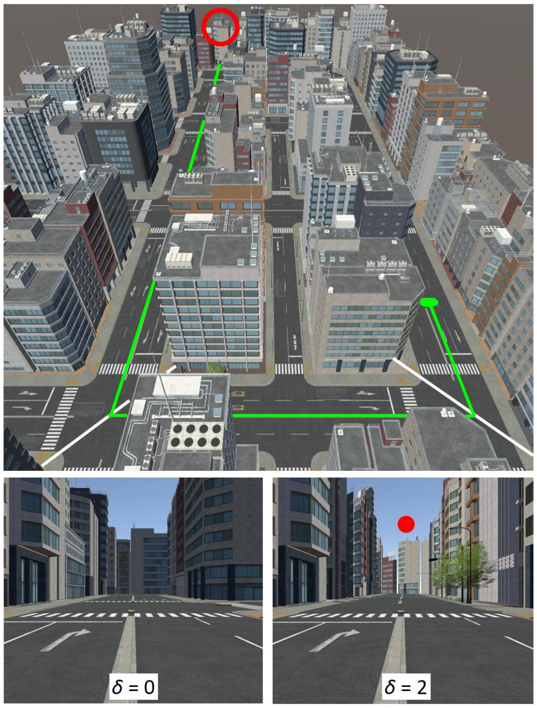


Fig. 16: City example: *PreVR* allows the user (green dot) to see a distant building (red circle and dot) around two corners.

## 6 Conclusions and Future Work

We have presented *PreVR*, an approach for providing variable-depth previews in VR. The more powerful the discocclusion effect, the less navigation is required. We have considered the scenario of a stationary target, as well as the scenario of a dynamic target, including the challenging scenario of a transient target moving discontinuously through the VE. The *PreVR* variant with an unbounded disocclusion order shows VE exploration efficiency gains in all scenarios.

*PreVR* is a flexible and general approach to disocclusion that is ready to be integrated into VR applications, to validate it a variety scenarios, including browsing without a priori knowledge of objects of interest to the user. Our current studies have investigated disocclusion effects constructed based on a planar piecewise linear path. Future work could examine disocclusion with a non-planar path, for example to preview second floor rooms from a ground floor user position, or general disocclusion defined by a graph, leveraging not only frustum bending, but also frustum splitting and merging operations.

Future work could also investigate disocclusion in VEs that have not only dynamic targets, but also dynamic occluders. Based on our fast disocclusion effect construction times, we do not anticipate difficulties in adapting the disocclusion effect in real time to circumvent any dynamic occluder that blocks an earlier constructed piecewise linear connection from the user to the target.

Our work is anchored by the thesis that VR can and should provide the user with visualization capabilities that go beyond those available in the real world. Such capabilities, although inherently of an overt nature, can lead to substantial visualization efficiency gains, without diminishing user comfort, benefiting VR applications.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 2219842 and 2212200.

# References

[1] Quest 2 virtual reality headset. https://www.meta.com/quest2. Accessed: 2023-10-3. 2

[2] 2023 Electronic Arts, Inc. Need for Speed. https://www.ea.com/games/need-for-speed/need-for-speed-unbound, 2023. 3

[3] B. Avery, C. Sandor, and B. H. Thomas. Improving spatial perception for augmented reality x-ray vision. In *2009 IEEE Virtual Reality Conference*, pp. 79–82. IEEE, 2009. 2

[4] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008. 8

[5] P. Bimberg, T. Weissker, and A. Kulik. On the usage of the simulator sickness questionnaire for virtual reality research. In *2020 IEEE conference on virtual reality and 3D user interfaces abstracts and workshops (VRW)*, pp. 464–467. IEEE, 2020. 5

[6] C. E. Bonferroni. Il calcolo delle assicurazioni su gruppi di teste. *Studi in onore del professore salvatore ortu carboni*, pp. 13–60, 1935. 6

[7] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 annual symposium on computer-human interaction in play*, pp. 205–216, 2016. 2

[8] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996. 5

[9] M. Burns and A. Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. *ACM transactions on graphics (TOG)*, 27(5):1–7, 2008. 2

[10] F. Buttussi and L. Chittaro. Locomotion in place in virtual reality: A comparative evaluation of joystick, teleport, and leaning. *IEEE transactions on visualization and computer graphics*, 27(1):125–136, 2019. 2

[11] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013. 6

[12] M. Dennison and M. D'Zmura. Effects of unexpected visual motion on postural sway and motion sickness. *Applied ergonomics*, 71:9–16, 2018. 7

[13] O. J. Dunn. Multiple comparisons using rank sums. *Technometrics*, 6(3):241–252, 1964. 6

[14] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE transactions on visualization and computer graphics*, 14(5):1095–1109, 2008. 2

[15] A. S. Fernandes and S. K. Feiner. Combating vr sickness through subtle dynamic field-of-view modification. In *2016 IEEE symposium on 3D user interfaces (3DUI)*, pp. 201–210. IEEE, 2016. 7

[16] R. A. Fisher. On the interpretation of $\chi$ 2 from contingency tables, and the calculation of p. *Journal of the royal statistical society*, 85(1):87–94, 1922. 6

[17] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 3

[18] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993. 5

[19] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952. 6

[20] E. Langbehn, P. Lubos, G. Bruder, and F. Steinicke. Application of redirected walking in room-scale vr. In *2017 IEEE Virtual Reality (VR)*, pp. 449–450. IEEE, 2017. 2

[21] E. Langbehn, P. Lubos, and F. Steinicke. Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking. In *Proceedings of the Virtual Reality International Conference-Laval Virtual*, pp. 1–9, 2018. 2

[22] S. Liao, Y. Zhou, and V. Popescu. Ar interfaces for disocclusion—a comparative study. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 530–540. IEEE, 2023. 3

[23] E. M. Lidal, H. Hauser, and I. Viola. Design principles for cutaway visualization of geological models. In *Proceedings of the 28th Spring Conference on Computer Graphics*, pp. 47–54, 2012. 2

[24] Y.-T. Lin, Y.-C. Liao, S.-Y. Teng, Y.-J. Chung, L. Chan, and B.-Y. Chen. Outside-in: Visualizing out-of-sight regions-of-interest in a 360 video using spatial picture-in-picture previews. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 255–265, 2017. 2, 3

[25] S. Mirhosseini, I. Gutenko, S. Ojal, J. Marino, and A. E. Kaufman. Automatic speed and direction control along constrained navigation paths. In *2017 IEEE Virtual Reality (VR)*, pp. 29–36. IEEE, 2017. 7

[26] R. Nakatani, D. Kouno, K. Shimada, and T. Endo. A person identification method using a top-view head image from an overhead camera. *J. Adv. Comput. Intell. Intell. Informatics*, 16(6):696–703, 2012. 2

[27] V. Popescu, P. Rosen, and N. Adamo-Villani. The graph camera. *ACM Trans. Graph.*, 28(5):1–8, dec 2009. doi: 10.1145/1618452.1618504 3

[28] E. D. Ragan, S. Scerbo, F. Bacim, and D. A. Bowman. Amplified head rotation in virtual reality and the effects on 3d search, training transfer, and spatial orientation. *IEEE transactions on visualization and computer graphics*, 23(8):1880–1895, 2016. 7

[29] S. Razzaque. *Redirected walking*. The University of North Carolina at Chapel Hill, 2005. 2

[30] M. Rietzler, M. Deubzer, T. Dreja, and E. Rukzio. Telewalk: Towards free and endless walking in room-scale virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–9, 2020. 2

[31] S. S. Sawilowsky. New effect size rules of thumb. *Journal of modern applied statistical methods*, 8(2):26, 2009. 6

[32] Z. S. See, D. Santano, M. Sansom, C. H. Fong, and H. Thwaites. Tomb of a sultan: a vr digital heritage approach. In *2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems & Multimedia (VSMM 2018)*, pp. 1–4. IEEE, 2018. 2

[33] A. L. Simeone, N. C. Nilsson, A. Zenner, M. Speicher, and F. Daiber. The space bender: Supporting natural walking via overt manipulation of the virtual environment. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 598–606. IEEE, 2020. 2

[34] J. L. Souman, P. R. Giordano, M. Schwaiger, I. Frissen, T. Thümmel, H. Ulbrich, A. D. Luca, H. H. Bülthoff, and M. O. Ernst. Cyberwalk: Enabling unconstrained omnidirectional walking through virtual environments. *ACM Trans. Appl. Percept.*, 8(4), dec 2011. doi: 10.1145/2043603.2043607 2

[35] K. M. Stanney, R. S. Kennedy, and J. M. Drexler. Cybersickness is not simulator sickness. In *Proceedings of the Human Factors and Ergonomics Society annual meeting*, vol. 41, pp. 1138–1142. SAGE Publications Sage CA: Los Angeles, CA, 1997. 7

[36] E. A. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte. Leveraging change blindness for redirection in virtual environments. In *2011 IEEE Virtual Reality Conference*, pp. 159–166. IEEE, 2011. 2

[37] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum, and M. Bolas. Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):555–564, 2012. 2

[38] Q. Sun, A. Patney, L.-Y. Wei, O. Shapira, J. Lu, P. Asente, S. Zhu, M. McGuire, D. Luebke, and A. Kaufman. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 2

[39] Q. Sun, L.-Y. Wei, and A. Kaufman. Mapping virtual and physical reality. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016. 2

[40] Unity Software, Inc. Unity Real-Time Development Platform (Version 2021.3.91). https://unity.com/, 2023. 5

[41] M.-L. Wu and V. Popescu. Efficient vr and ar navigation through multiperspective occlusion management. *IEEE transactions on visualization and computer graphics*, 24(12):3069–3080, 2017. 2, 3

[42] M.-L. Wu and V. Popescu. Anchored multiperspective visualization for efficient vr navigation. In *Virtual Reality and Augmented Reality: 15th EuroVR International Conference, EuroVR 2018, London, UK, October 22–23, 2018, Proceedings 15*, pp. 240–259. Springer, 2018. 3, 4

[43] J. Yu and L. McMillan. General linear cameras. In *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part II 8*, pp. 14–27. Springer, 2004. 3

[44] S. Zollmann, R. Grasset, G. Reitmayr, and T. Langlotz. Image-based x-ray visualization techniques for spatial understanding in outdoor augmented reality. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, pp. 194–203, 2014. 2