Quantifiable Fine-Grain Occlusion Removal Assistance for Efficient VR Exploration

Jian Wu, Lili Wang, Hui Zhang, and Voicu Popescu

Abstract—This article presents an occlusion management approach that handles fine-grain occlusions, and that quantifies and localizes occlusions as a user explores a virtual environment (VE). Fine-grain occlusions are handled by finding the VE region where they occur, and by constructing a multiperspective visualization that lets the user explore the region from the current location, with intuitive head motions, without first having to walk to the region. VE geometry close to the user is rendered conventionally, from the user's viewpoint, to anchor the user, avoiding disorientation and simulator sickness. Given a viewpoint, residual occlusions are quantified and localized as VE voxels that cannot be seen from the given viewpoint but that can be seen from nearby viewpoints. This residual occlusion quantification and localization helps the user ascertain that a VE region has been explored exhaustively. The occlusion management approach was tested in three controlled studies, which confirmed the exploration efficiency benefit of the approach, and in perceptual experiments, which confirmed that exploration efficiency does not come at the cost of reducing spatial awareness and sense of presence, or of increasing simulator sickness.

Index Terms—VR exploration, fine grain disocclusion, multiperspective visualization, graph camera, occlusion quantification

1 Introduction

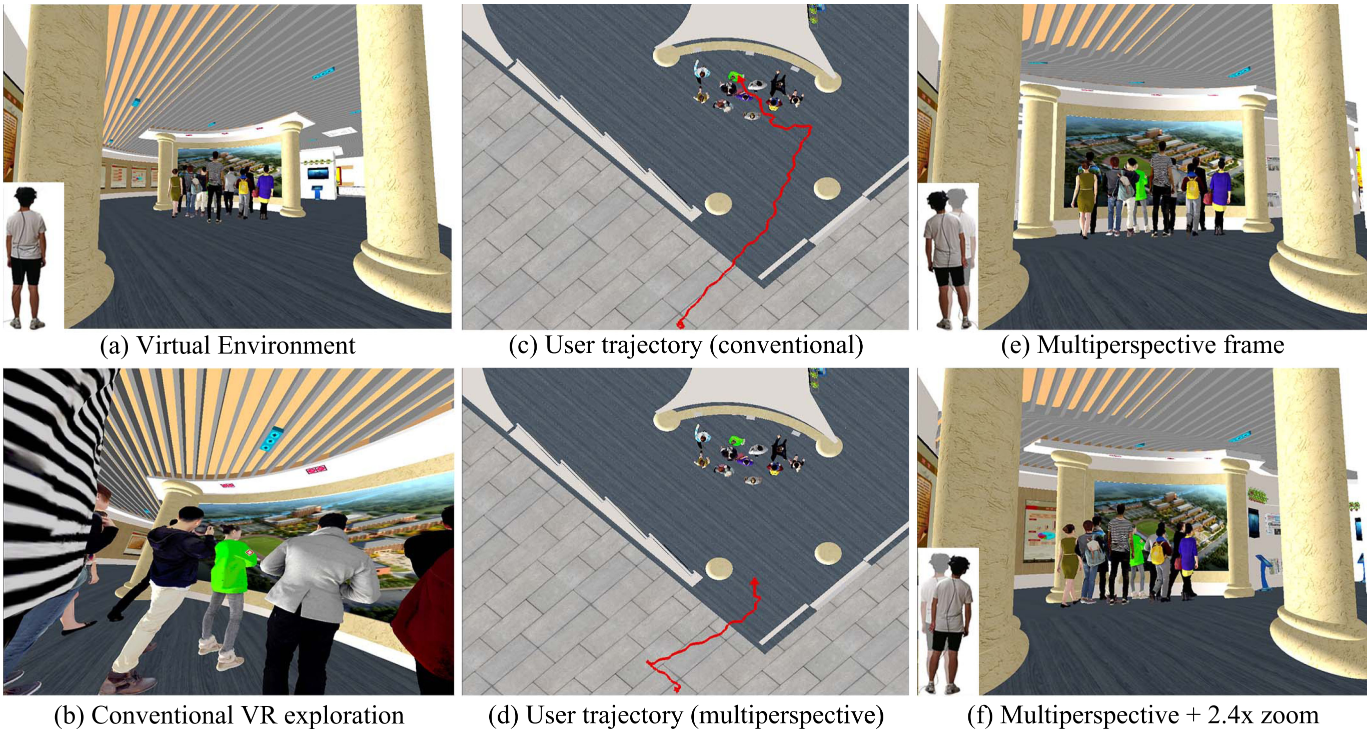
Occlusions pose an important challenge for virtual reality (VR) applications because they limit what the user can see from the current viewpoint. The conventional remedy is for the user to walk around occluders to gain line of sight to the hidden parts of the virtual environment (VE). Such a sequential VE exploration is inefficient, as it takes a long time to circumvent all occluders, one at the time, in search of regions of interest. When the disoccluded part of the VE proves to be of no interest, the navigation effort is wasted. Modern VR systems provide inside-looking-out tracking and do not confine the user to a small outside-looking-in tracked area. However, the user's freedom of motion remains hindered by real world physical constraints, such as furniture and walls. Overcoming these constraints require redirection and/or teleportation, which can be disorienting, making unnecessary navigation even more costly. Finally, another incentive for making navigation more efficient is to improve the ergonomics of the VR experience by reducing the physical effort required from the user.

Several occlusion management techniques have been explored in the context of VR, inspired by techniques originally developed in the general context of visualization. The transparency technique renders occluding layers semi-transparently for the user to see through to a potential region of interest from the current location, without having to walk around occluders [1], [2], [3], [4]. Transparency works well in the case of a single occluding layer but scales poorly with occlusion complexity. Top view methods temporarily fly the user high above the VE to bypass ground level occlusions, such as those cast by buildings in an urban environment [5], [6]. Abandoning the first-person view breaks the sense of presence. Recently, multiperspective visualization methods have been employed in VR to let the user see around occluders from the current location [7], [8], [9], [10], [11], [12]. However, current multiperspective visualization techniques only work for large occluders, such as walls or cars, whose long silhouette edges are needed to route rays around occluders [13], [14]. No current VR disocclusion method can remove fine-grain occlusions, as needed, for example, to find an apple in a leafy tree, or a person in a crowd. Furthermore, no prior method quantifies the amount of occlusion from a current viewpoint, so the user does not know how much of the VE has been explored and how much remains to be disoccluded.

In this paper we make two contributions to VR occlusion management, and we evaluate them in three user studies. The first contribution provides fine-grain disocclusion support. Given the current user view, the method detects a VE region with fine-grain occlusions, and builds a multiperspective visualization that lets the user disocclude the region efficiently, from the current viewpoint, without first having to get close to the region (Fig. 1). Our multiperspective visualization also allows the user to zoom in on the region of interest, which facilitates the exploration task. Furthermore, our method renders nearby geometry from the original, primary user viewpoint, which anchors the user. This first-person visualization of nearby geometry responds to the user head motions as usual and as expected by the user, preventing disorientation and simulator sickness.

- *Jian Wu, Lili Wang, and Hui Zhang are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Peng Cheng Laboratory, Shenzhen, Guangdong 518000, China. E-mail: {lanayawj, wanglily}@buaa.edu.cn, zh_1520@163.com.*
- *Voicu Popescu is with the Purdue University, West Lafayette, IN 47907 USA. E-mail: popescu@purdue.edu.*

Manuscript received 23 June 2020; revised 27 Dec. 2020; accepted 18 Jan. 2021. Date of publication 21 Jan. 2021; date of current version 1 Aug. 2022. (Corresponding author: Lili Wang.)
Recommended for acceptance by M. Marchal.
Digital Object Identifier no. 10.1109/TVCG.2021.3053287

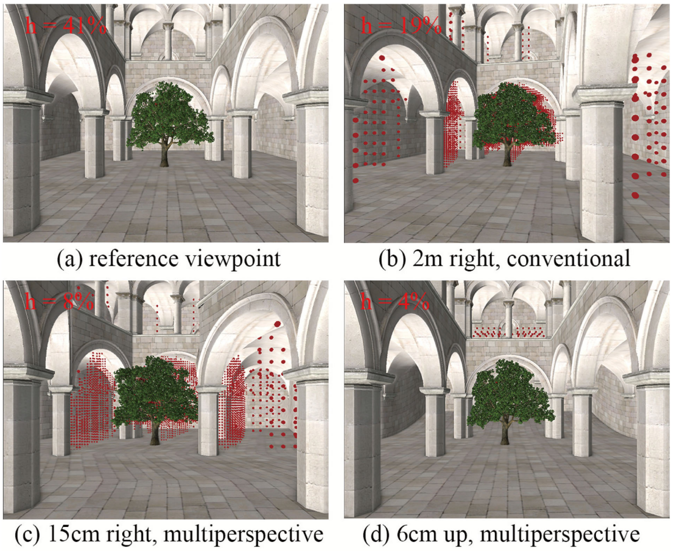

(a) Virtual Environment (c) User trajectory (conventional) (e) Multiperspective frame

(b) Conventional VR exploration (d) User trajectory (multiperspective) (f) Multiperspective + 2.4x zoom

Fig. 1. A user of a VR application is trying to find the person with the green jacket, which is occluded from the current user position (a). With conventional VR exploration, the user has to move closer to the crowd to perform the search (b), which requires significant viewpoint translation (c). With our multiperspective VR exploration technique, the user can search the crowd from the current position (e), with an intuitive lateral head translation, which makes the exploration of the virtual environment more efficient (d). The user can also zoom in on the crowd for greater detail (f). Nearby geometry (e.g., nearby pillars) is rendered from the original user perspective at all times, anchoring the user to prevent disorientation and simulation sickness (e, f).

The second contribution of our paper is a method for quantifying and locating occlusions, which informs the user of how much remains to be seen and where. Our method computes how much of what is visible from the current region is *not* visible from the reference viewpoint. In other words, the method quantifies how much more *could* be seen if the user translated the viewpoint within the current region. In Fig. 2a, the user misses $h = 41\%$ of what is visible from a 2m radius sphere centered at the reference viewpoint. As the user disoccludes the VE interactively by walking the viewpoint conventionally (b), or by using our multiperspective visualization and translating their head (c-d), the occlusion level is reduced. Leveraging the occlusion quantification, the user can continue to reduce occlusion interactively, with guidance from the system, based on the known location of the residual occlusions. Here, after the lateral head translations (c), the system indicates that residual occlusions are located behind the horizontal edges of the second floor (d), and the user can remove them by getting on their tiptoes translating the viewpoint upwards.

We tested the effectiveness of our fine-grain disocclusion method at improving VE exploration efficiency in three controlled between-subject studies. VR exploration efficiency was quantified by measuring task completion time, viewpoint translation, view direction rotation, and number of redirections caused by reaching the boundary of the tracked physical space. Our method showed a significant advantage for each of these metrics. Furthermore, we have quantified spatial awareness through estimation of relative depth, layout recall, pointing direction, and location repeating experiments, which showed that the navigation efficiency brought by our method does *not* come at the cost of a significant loss of spatial awareness, compared to conventional VR exploration. A standard presence questionnaire shows that our method does not incur a significant loss of presence



(a) reference viewpoint (b) 2m right, conventional

(c) 15cm right, multiperspective (d) 6cm up, multiperspective

Fig. 2. Occlusion quantification and localization. The pillars and the tree occlude $h = 41\%$ of the VE (a). If the user walks 2m to the right, occlusion is reduced to 19 percent (b). The disoccluded parts are shown with red dots. Alternatively, with our multiperspective visualization, small head translations efficiently reduce occlusion to 8 percent, and 4 percent (c-d).

compared to conventional VR. Finally, a standard simulation sickness questionnaire revealed that our user anchoring succeeds at avoiding simulator sickness.

2 Previous Work

We briefly discuss related prior work in three areas.

2.1 VR Exploration

The user's sense of presence in the VE is highest when the user is allowed to move freely [15]. However, moving freely is not always possible or desirable, e.g., due to obstacles in the real world or to the large size of the VE.

One approach is to use mechanical devices that keep a walking user in place, such as treadmills [16] or moving platforms [17]. Another option is to ask the user to simulate walking by walking in place [18]. An even farther departure from true walking is to let the user walk by simply tilting their head in the intended direction [19].

Another approach is to let the user actually walk in the VE and then to address problems as they arise. Teleportation is a generic mechanism for temporarily suspending the one-to-one mapping between the user motions and those of their avatar in order to reposition the avatar at a distant location. A popular approach lets users jump to locations to which they point [20]. One challenge is perturbing the user's sense of presence and spatial awareness. A recent teleportation mechanism succeeds at preventing user disorientation by letting the user define the teleportation destination as a panorama, to which the user then transitions on a path inspired by a Moebius surface [21]. The discrepancy between the changing visual cues and the absence of acceleration cues during teleportation can also cause simulator sickness. Simulator sickness can be alleviated by making it clear to the user that the view changes from first to third person during teleportation, for example by having the user select the destination from a set of floating icons [22].

Researchers have also investigated approaches based on manipulating the VE geometry or the user tracking data. In static redirected walking, the VE is folded to fit the physical space. For example, one method aims to minimize the occurrence of geometry anomalies [23], while another aims to make the geometry deformations as inconspicuous as possible [24]. In dynamically redirected walking, the user orientation is manipulated on the fly at the most opportune moments, such as, for example, during user head turns, blinks, an saccades [25], [26]. The user is imperceptibly turned away from the boundaries of the physical space, enabling infinite walking in the VE. These approaches can be used in conjunction with our orthogonal work on occlusion management. With our method, the user can decide from the current position whether a distant VE region is worth a closer look, which can reduce the number of teleportations. Our goal is not to hide the disocclusion effects from the user, but rather to increase VE exploration efficiency without user confusion or simulator sickness.

2.2 Occlusion Management in Visualization

Our method is inspired by occlusion management research in the context of visualization. One approach is to render occlusions transparently [1], [3], [4]. The approach is less effective at dealing with multiple occlusion layers. Cutaway methods remove parts of the occluder to reveal the scene behind [2]. This leads to a contextual information loss, which is costly in the case of head-mounted display (HMD) visualization, where it can lead to user disorientation. To better preserve spatial context, a recent approach enlarges the field of view and melts away occluding layers at the center [27]. The exploded view approach, initially proposed as an illustration technique, has the unique advantage of being able to disocclude objects completely contained by several occluding layers [28], but it perturbs scene topology. Another approach relies on a top-view map that help users find their current position and plan a long route to reach occluded parts of the scene. However, using a top-view map in HMD applications is challenging due to the small display, and due to the loss of presence caused by the switch to the radically different third person visualization.

Multiperspective visualization methods render scenes by integrating samples from multiple viewpoints. The image generalization amounts to a powerful disocclusion mechanism. A method developed specifically for 3D landscape and urban environment visualization composites a scene overview into the conventional view to assist with navigation [9]. Multiperspective visualization has been extended to multi-scale visualization by rendering the secondary perspectives with at a different level of detail [11]. Early research used conventional, linear rays, and focused on relaxing the single center of projection constraint [8], [29]. More recent work generalizes rays to the locus of 3D scene points that project at a given image location, which allows for piecewise linear [10] or even curved rays [30] that are routed around occluders to reach distant regions of interest.

Our approach relies on multiperspective visualization for disocclusion. We build the multiperspective effect based on the graph camera [31], which provides a flexible paradigm for multiperspective visualization. The graph camera is a graph of conventional camera frusta, built recursively through successive bending, splitting, and merging operations. Our disocclusion effect employs a two-bend, three-frusta graph camera. Our method automatically finds a small grain occlusion region in the VE and then constructs a graph camera by selecting the secondary viewpoints and the planes that connect the frusta.

2.3 Occlusion Management in VR

Several occlusion management methods have been developed specifically to increase VE exploration efficiency. One approach is to provide the user with navigational hints, for example through a virtual tape for the user to follow, through a virtual bird companion, or through a magic wand [32]. The approach requires knowing what the user is interested in, i.e., knowing where to send the user. Our method only takes the environment geometry as input, without any information on potential regions of interest. Our method does provide visual cues to the user, but the hints correspond to reference view volume sub-regions which are yet to be explored. This residual occlusion regions might reveal a hidden object, or they might be empty.

A recent study compared the effectiveness of three conventional visualization occlusion management techniques,

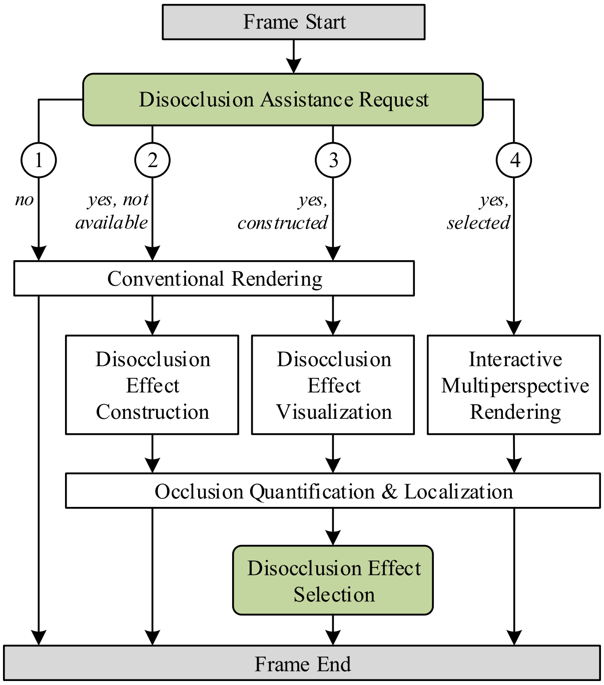

Frame Start
Disocclusion Assistance Request
1
2
3
4
no
yes, not available
yes, constructed
yes, selected
Conventional Rendering
Disocclusion Effect Construction
Disocclusion Effect Visualization
Interactive Multiperspective Rendering
Occlusion Quantification & Localization
Disocclusion Effect Selection
Frame End

Fig. 3. Overview of our method for occlusion removal assistance.

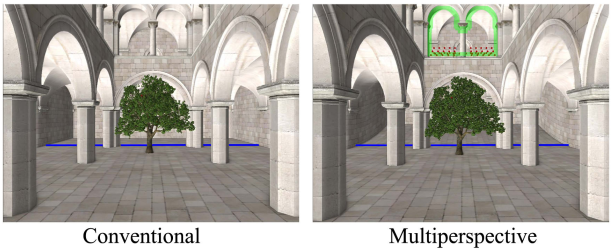


Fig. 4. Coarse grain disocclusion effect based on second floor portal (green). The multiperspective visualization shows the part of the VE beyond the portal plane (blue line added here for illustration) from a secondary viewpoint, high above the user viewpoint.

i.e., top-view, X-ray, and multiperspective visualization in two VR tasks [33]. One task was finding a missing child in a virtual parking garage, and the second task was shopping for specific items in a virtual supermarket. Overall, participants in the multiperspective visualization condition performed the best. X-ray worked for a small number of occluding layers, e.g., a single wall in the garage, and was less effective in the supermarket VE where the products on the shelves created even as few as two occlusion layers. The top-view visualization gives the user a comprehensive view of the VE, but does not allow examining potential regions of interest in sufficient detail for disambiguation. The X-ray approach has the unique advantage of providing disocclusion when the object of interest is completely enclosed, as multiperspective and top-view visualizations require the availability of a path from the user viewpoint to the object.

Multiperspective visualization has shown improved VR exploration efficiency [13]. Porting multiperspective visualization to VR brings two important challenges. One is a user interface challenge. The complex camera implementing the multiperspective visualization has many more degrees of freedom compared to the six degrees of freedom of a conventional camera, and they have to be controlled with an intuitive interface. We take the previous work approach of letting the user control the magnitude of the disocclusion effect with small head translations that are amplified and applied to the secondary viewpoints [14]. The second challenge is to avoid simulator sickness, for which we take the previous approach of always rendering the geometry close to the user with the primary viewpoint, anchoring the user [13], [14]. This prior work justifies our design, which was then thoroughly and successfully tested for simulator sickness and user disorientation in the context of our paper.

However, prior work only handled large occluders [13], [14], whose edges were used to construct the multiperspective visualization. Our work contributes a method for the automatic disocclusion of regions with highly fragmented geometry, which cannot be handled by prior work [13], [14]. Furthermore, our work contributes a method for the quantification and localization of residual occlusions, which assists the user in achieving a systematic and complete VE exploration, whereas in previous work [13], [14], disocclusion proceeded haphazardly.

3 Approach Pipeline

Fig. 3 gives an overview of our interactive method for occlusion removal assistance. The frame provided to the user is computed in one of four major ways. First, if the user has not made a disocclusion assistance request (1), the frame is rendered conventionally.

If the user has made a disocclusion assistance request, but no disocclusion effect is available (2), the system renders the frame conventionally and then computes a set of disocclusion effects by analyzing the resulting z-buffer. Our system constructs two types of disocclusion effects. One type handles coarse grain occlusions, such as those created by doorways, windows, and large objects. We construct coarse grain disocclusion effects with a portal-based prior art method [13]. For the example in Fig. 2d, the coarse grain disocclusion effect was constructed with the portal defined by the second floor opening (Fig. 4). The second type of disocclusion effect targets fine-grain occlusions, and its construction is described in Section 4.

Once the disocclusion effects have been constructed (3), they are visualized to allow the user to select one of them. For the example in Fig. 2 there are 7 disocclusion effects available, and the user cycles through them with the controller to select one. Fig. 5 shows two such disocclusion effects. If the user has selected one of the disocclusion effects (4), a multiperspective frame is rendered with the occlusion effect deployed gradually, under the user control, as described in Section 5. Once the user requests disocclusion assistance, the system estimates visibility from the current region, and informs the user of how much of the VE remains to be discovered and where, using our occlusion quantification and localization method, as described in Section 6.

4 Fine Grain Disocclusion Effect Construction

For many VEs, regions with complex geometry create a complex, fragmented occlusion pattern, with many small occluders. The prior art portal-based disocclusion method

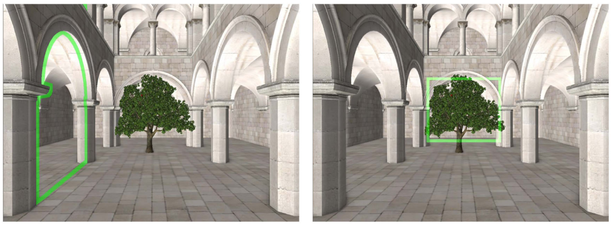


Fig. 5. Visualization of available disocclusion effects: a coarse grain portal disocclusion effect (left), and a fine grain disocclusion effect constructed for the tree (right).

[13] would require constructing a portal disocclusion effect for each of the tens or hundreds of small occluders. Even if such a complex construction succeeds, asking the user to disocclude by selecting and deploying each portal-based disocclusion effect interactively is prohibitively slow. Furthermore, whereas in the case of coarse grain occlusions the large portal offers a clean and intuitive way of partitioning the VE for multiperspective rendering, using a large number of small portal-based disocclusion effects deployed sequentially would perturb the local spatial relationships, which leads to VE distortion and to a cryptic multiperspective visualization. What is needed is a method for handling all fine grain occlusions in a region simultaneously, in parallel, while minimizing local VE distortions. Based on these design concerns, we have developed a disocclusion method that first detects VE regions where fine-grain occlusions occur (Section 4.1) and then constructs a multiperspective visualization (Section 4.2), which lets the user change the viewpoint on the region, disoccluding it efficiently.

4.1 Fine Grain Occlusion Region Detection

The locations of occlusions in a given view is indicated by the presence of discontinuities in the depth buffer of a conventional rendering of the VE. Indeed, depth changing significantly from one pixel to the next can indicate the jump from a front, occluder surface to a back, occluded surface. Even a minimal viewpoint translation moves these two samples apart, revealing new samples not visible in the initial view. In the case of portals, depth discontinuity samples form long edges that can be used to hinge rays. In the case of fine grain occlusions, one must find depth buffer regions with high depth discontinuity density, cluster neighboring depth discontinuity samples in 3D regions, and build disocclusion effects to handle entire regions coherently.

Algorithm 1. Fine Grain Occlusion Region Detection

```
Input: current viewpoint $V$, depth buffer $D$ rendered from $V$
Output: 3D bounding box $R$ of fine grain occlusion region
1: $L$ = Laplacian($D$);   // second order edge detection
2: $T$ = Threshold($L$);   // keep strong edges
3: $F$ = BoxFilter($T$);   // compute edge density
4: $E$ = ErodeDilate($F$);   // remove isolated edge islands
5: $R_{2d}$ = BoundingBox($F$);   // 2D region
6: $(d_{min}, d_{max})$ = MinMax($R_{2d}$, $D$);
7: $R$ = UnProject($V$, $R_{2d}$, $d_{min}$, $d_{max}$);   // 3D region
8: return $R$
```

Algorithm 1 outlines our process for detecting a region with fine grain occlusions, which is also illustrated in Fig. 6.

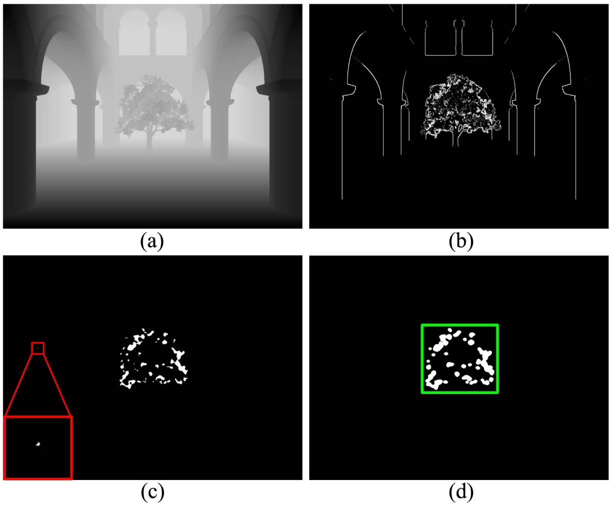


Fig. 6. Detection of fine-grain occlusion region: depth buffer (a), second order depth difference (b), thresholded depth discontinuity density with residual large occluder edge samples (magnification of red rectangle for illustration purposes) (c), and final fine-grain occlusion region (green) after repeated erosion and dilation (d).

The algorithm takes as input the reference viewpoint V and the depth buffer D rendered from V, and produces the 3D bounding box R of the fine grain occlusion region. The conventional depth buffer (Fig. 6a) is processed with a Laplacian operator to detect pixels with a large variation of the second order depth difference (line 1). This is needed for surface orientation invariance, as the Laplacian of the depth of a sample of a planar surface is exactly zero, regardless of the plane orientation. Then low intensity depth pixels are discarded by thresholding (line 2, and Fig. 6b).

At this point, we have identified all strong depth discontinuity pixels, including the depth discontinuities of large occluders, which have to be eliminated in order to obtain an accurate estimate of the region with fine grain occlusions. The large occluder depth discontinuity pixels are eliminated first by computing the local depth discontinuity density with a box filter that counts the number of depth discontinuity pixels in the neighborhood of each pixel (line 3). Depth discontinuity pixels with low density are discarded (line 3, and Fig. 6c). While this discards most large occluder depth discontinuities, a few spurious large occluder depth discontinuity pixels remain at regions where large occluder edges intersect (see magnified red box in Fig. 6c). We remove the remaining large occluder depth discontinuity pixels through a process of repeated erosion and dilation, which eliminates isolated pixels but does not eliminate pixels who are supported by other pixels nearby (line 4). The detected fine grain depth discontinuity pixels are grouped with a 2D axis aligned bounding box (line 7, and Fig. 6d), which is then unprojected to a 3D bounding box by taking into account the smallest and largest depth over the detected depth discontinuity pixels.

4.2 Multiperspective Visualization Construction

Once a fine grain occlusion VE region is detected, we construct a multiperspective disocclusion effect that satisfies the following design concerns. First, the user should be able to change the viewpoint on the region of interest as if they

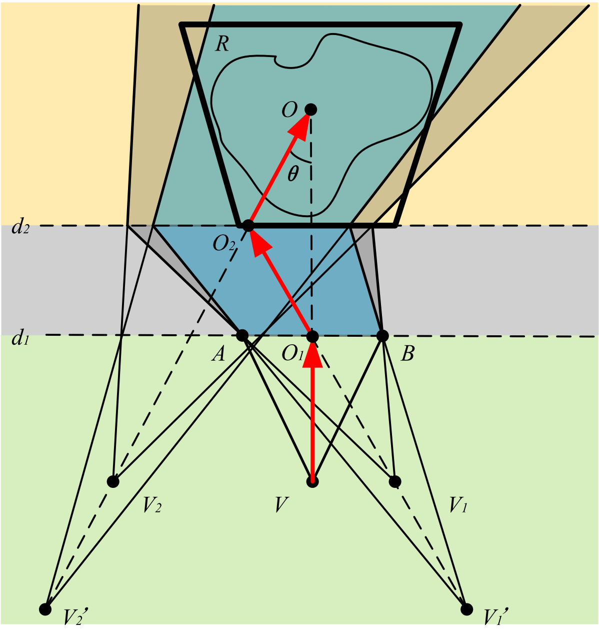
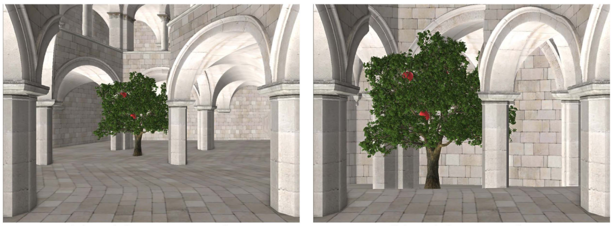


Fig. 7. Two-bend graph camera construction for fine-grain disocclusion of region R. Secondary viewpoints V_1 and V_2 allow changing the view direction on R by angle θ, while keeping the projection of center O of R in place. Nearby geometry is rendered with the original viewpoint V (green), anchoring the user. When the secondary viewpoints V_1 and V_2 are moved back to V_1' and V_2', image region AB is mapped to a smaller VE region (blue), which produces a magnification effect.

orbited around the center of the region of interest. Second, as the region to be disoccluded is typically far away, the user should be able to zoom in on the region to get a more detailed look. Third, the nearby geometry should be rendered conventionally, from the user viewpoint, to anchor the user in order to avoid simulator sickness. If the visualization of the geometry close to the user changes with the user view as expected, and the multiperspective effect is confined to the distant geometry, then the user internalizes the disocclusion effect as a useful animated deformation of the VE, and not as a first person visualization that is incongruent with the user perception.

We construct the fine grain disocclusion effect based on the graph camera, a flexible multiperspective visualization framework that allows introducing additional viewpoints while providing visualization continuity and non-redundancy, as well as interactive rendering performance [31]. Given a user view and a 3D bounding box R of a fine-grain occlusion region, we construct a two-bend graph camera as follows.

The VE is partitioned with two vertical planes at distances d_1 and d_2 (Fig. 7). d_2 is the distance to the front face of R, and d_1 is half of d_2. We place two secondary viewpoints V_1 and V_2 on the same vertical plane as the user viewpoint V. The VE is rendered with V up to d_1 (green), anchoring the user, with V_1 from d_1 to d_2 (gray), and with V_2 beyond d_2 (orange). Here the disocclusion effect is deployed to the let the user see R from the left, so V_2 is to the left of V, and V_1 is to the right of V. The farther V_2 is from V, the more the direction from which R is viewed changes. Two secondary viewpoints are needed in order to achieve the view direction change while keeping the center O of R in place. In other words, in all multiperspective visualizations, O projects where it does in the conventional image. The resulting graph camera has piecewise linear rays with three segments. For example, the ray that samples O has segments VO_1, O_1O_2, and O_2O. No rays intersect, therefore the resulting visualization is non-redundant, and no objects disjoint in 3D can intersect in the visualization.



(a) without zoom in (b) with zoom in

Fig. 8. Zooming effect in multiperspective visualization.

Our two-bend graph camera allows simulating a zoom in operation by translating the secondary viewpoints back on their view direction, i.e., V_1' and V_2' in Fig. 7. Moving the viewpoint back magnifies the visualization of the fine grain disocclusion region, while keeping the center of the region in place (Fig. 8, Fig. 9). As explained in Section 5, the graph camera is not used to show the image to the user directly, but rather to warp the VE, which is then rendered with conventional cameras for the left and right user eyes. This way, the manipulation of the positions of viewpoints V_1' and V_2' have the effect of scaling up a region of the VE, without modifying the user field of view, which, as in any VR application, is fixed, as defined by the HMD.

5 Multiperspective Visualization

Once the graph cameras implementing the disocclusion effects are constructed, the user can select one disocclusion effect and deploy it. The user deploys the disocclusion effect interactively and gradually by progressively moving the secondary viewpoints away from the main user viewpoint. This results in a gradual change of the conventional single perspective image into a multiperspective image, which changes the perspective on the VE region of interest, disoccluding it. In Fig. 7, secondary viewpoints V_1 and V_2 start out at V, and then move away from V increasing the angle θ under which the region of fine grain occlusions is seen. We leverage the tracked VR headset to let the user control the disocclusion effect intuitively, with small head translations, which are amplified to move the secondary viewpoints. In Fig. 1 the user translates their head to the right to see around the right side of the tall person hiding the person with a green jacket. In Fig. 2 the user gets on their tiptoes to see over the second floor ledge.

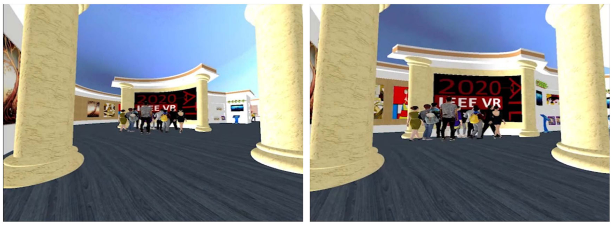


Fig. 9. Conventional visualization (left) and our zoom in effect (right), which scales up the region of interest.

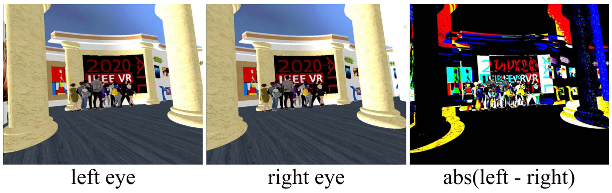


Fig. 10. Stereo user images, and disparity image.

Given a user position, the graph camera routes rays around occluders to achieve the disocclusion effect. However, for the purpose of VR with correct stereo, we have to render two multiperspective visualizations, one for each eye, with appropriate disparity. We achieve this by first warping the VE using the graph camera, and then by rendering the warped VE conventionally, once for each eye.

The VE is warped by displacing its vertices. A vertex is first (1) projected with the graph camera, and then (2) unprojected with a conventional camera located at the user position. Graph camera projection first finds the sub-frustum that contains the vertex [31]. Since our graph cameras have at most three sub-frusta (Fig. 7), and since the sub-frusta are defined by vertical planes, finding the sub-frustum that contains a given vertex is straightforward. Then the vertex is projected directly onto the image plane using a concatenation of projection matrices from the vertex sub-frustum to the root frustum. In Fig. 7, a vertex that belongs to the sub-frustum with viewpoint V_2 (yellow) is projected with matrix $M_2 \times M_1 \times M$, where M_2, M_1, and M are the conventional projection matrices for the sub-frusta with viewpoints V_2, V_1, and V. (2) The projected vertex is pushed back into 3D space along its ray by unprojection with the root frustum camera, i.e., M^{-1}, at a depth equal with the sum of the segments that form the piecewise linear ray. For example, point O is pushed back at depth $VO_1 + O_1O_2 + O_2O$, and point O_2 is closer in the warped VE, at distance $VO_1 + O_1O_2$, which provides correct visibility sorting, perspective foreshortening, and disparity.

Once the VE is warped, we render the warped VE conventionally for each of the user's eyes, generating two multiperspective visualizations with appropriate disparity, enabling stereo viewing. For the correct stereo effect, commensurate to the inverse of the distance from the user to the VE surfaces along the piecewise linear rays, we render the displaced vertices with two root cameras, one for the left, and one for the right eye, as shown in Fig. 10. In order to model the warping of the VE with high fidelity, large triangles are subdivided. For the example in Fig. 1e, the floor has to be subdivided for the visualization to bend it according to the two additional perspectives. Graph camera projection can be decomposed into a vertex displacement followed by a conventional projection with the primary root frustum (Fig. 11).

6 Occlusion Quantification & Localization

So far we have described a method for addressing fine grain occlusions. Like prior art occlusion management methods, our method is heuristic in the sense that it does not guarantee that all of the VE is disoccluded. We have developed a method for informing the user of how much of the VE remains to be disoccluded, and where. Occlusions are quantified and localized with respect to the current region, and not with respect to the entire environment. For example, in the case of exploring an indoor environment, what the user needs is a way of accounting for how much of the current room has been seen, such that they can search it systematically and exhaustively, and not a quantification method that greedily directs the user towards the unexplored rooms before the exploration of the current room is complete.

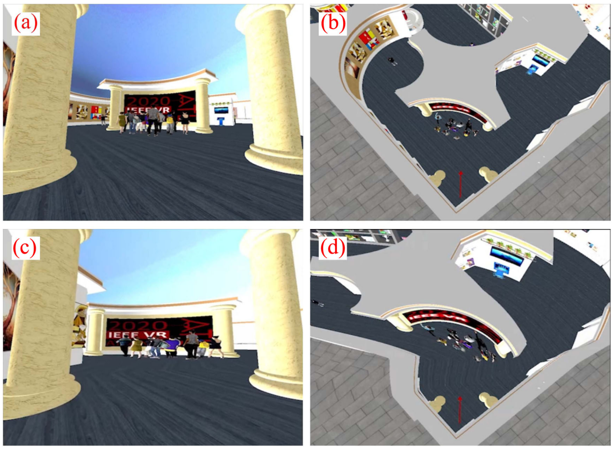


Fig. 11. Illustration of the VE warping induced by our multiperspective visualization: (a) conventional user view, (b) third person view of original VE, (c) multiperspective visualization, and (d) third person view of warped VE, which, when rendered conventionally, produces image (c).

Our method for quantifying and locating occlusions is general, i.e., it does not depend on the disocclusion method chosen by the user. For example, in addition to our multiperspective disocclusion, the method could be used to guide conventional disocclusion by interactive navigation with a conventional, single-perspective visualization. The method also brings the benefit of quantifying disocclusion efficiency, which allows comparing different disocclusion methods.

Our method proceeds in two steps. First, occlusion is initialized by finding the parts of the VE that are not visible from the current viewpoint but that are visible from nearby viewpoints (Section 6). Then, occlusion is updated for every frame, as the user disoccludes the VE interactively (Section 6).

Initial Occlusion Quantification & Localization. The algorithm for initial occlusion quantification and localization (Algorithm 2) takes as input a reference viewpoint V, a VE region G which is used to estimate the parts of the VE not visible from V but visible from nearby viewpoints, and the VE bounding box B. In this paper region G is defined as a sphere centered at V. The output of the algorithm is the set S of VE voxels whose centers are visible from a viewpoint in G but not from V, i.e., the voxels that remain to be disoccluded, and the number n_G of voxels visible from G, which is needed to estimate the relative amount of occlusion.

First, the algorithm defines a set A of auxiliary viewpoints by sampling the current VE region G (line 1). Then the VE is rendered conventionally from the reference viewpoint V and from all auxiliary viewpoints in A to obtain z-buffers which are used later to estimate visibility from each of these viewpoints (lines 2-4). Then the VE geometry is voxelized to estimate to discretize the visibility computations (line 5). A voxel visible from V is counted in n_G (lines 7-9). A voxel not visible from V but visible from an auxiliary viewpoint is added to S, and counted in n_G (lines 10-14).

Algorithm 2. Initial Occlusion Quantification and Localization

```
Input: reference viewpoint $V$, view region $G$, VE bounding
       box $B$
Output:set $S$ of voxels visible from $G$ but occluded from $V$,
number $n_G$ of voxels visible from $G$
 1:  $A$ = Sample($G \cap B$)
 2:  $ZB_V$ = Render($V$)
 3:  for each auxiliary viewpoint $a$ in $A$ do
 4:    $ZB_a$ = Render($a$)
 5:  end for
 6:  $X$ = Voxelize(Frustum($V$) $\cap B$)
 7:  for each voxel $x$ in $X$ do
 8:    if IsVisible($x, V, ZB_V$) then
 9:      $n_G$ = $n_G$+1
10:      continue
11:    end if
12:    for each auxiliary view point $a$ in $A$ do
13:      if IsVisible($x, a, ZB_a$) then
14:        $S = S \cup \{x\}$
15:        $n_G$ = $n_G$+1
16:        break
17:      end if
18:    end for
19:  end for
20:  return $S, n_G$
```

When the VE contains fine grain occlusions, the regions of fine grain occlusions is voxelized at a higher resolution than the rest of the view frustum, for a higher-fidelity quantification of occlusions. In lines 8 and 13, the finer voxels count proportionally to their volume. For example, if a fine grain occlusion region is voxelized at a 2x2x2 higher resolution, one of its voxels counts as 0.125.

Algorithm 3. Residual Occlusion Quantification and Localization

```
Input:view point $f$ of current frame, occluded voxel set $S$,
number of voxels $n_G$ visible from reference region
  Output:updated $S$, current occlusion level $h$
1:  $ZB_f$ = Render($f$)
2:  for each voxel $s$ in $S$ do
3:    if IsVisible($s, f, ZB_f$) then
4:      $S = S - \{s\}$
5:      VisualizeOcclusionRemoval($s$)
6:    end if
7:  end for
8:  $h$ = size($S$) / $n_G$
9:  return $S$, $h$
```

Interactive Occlusion Quantification & Localization. After the initial occlusion quantification and localization (Algorithm 2), our method visualizes and updates the residual occlusion as the user reduces it interactively, according to Algorithm 3. The algorithm takes as input the current viewpoint, the current set of occluded voxels S, and the number of voxels visible from the region around the reference viewpoint. The output are the updated set of occluded voxels S, and the current occlusion level h.

First, the frame is rendered conventionally to obtain a z-buffer ZB_f where to estimate visibility from the current viewpoint (line 1). Then each voxel s that remains to be disoccluded is checked for visibility in ZB_f. If s is visible, it means that the user has succeeded at disoccluding it. The disocclusion of s is visualized as a red sphere that disintegrates over a few frames (line 5), and s is removed from S (line 4). The relative amount of residual occlusion h is computed as the number of voxels in S, i.e., the number of voxels that remain to be disoccluded, over the total number of voxels visible from the region (line 6).

Since the red spheres disappear shortly after they are uncovered, their role is simply to let the user know that they found a part of the VE that was previously unexplored. Once the red spheres are gone, the user has the opportunity to explore the newly found part of the VE, for example, to search it for a specific object. Like with any glyph method to assist VR navigation, our red spheres add to the visual complexity of the VE, and they are not actually part of the VE, which incurs the risk of decreasing the user's visual exploration bandwidth. However, the transient nature of the red spheres minimize this risk, and the spheres simply indicate to the user that they should pay a particularly close attention to this part of the VE as it is seen for the first time.

7 Results and Discussion: User Studies

We have evaluated our method in three controlled user studies (N = 16 x 3), each in a different VE, with different VR exploration tasks. In addition, we have also conducted perceptual experiments to investigate situational awareness and motion sickness implications of our method.

7.1 Methods

VR Setup. We used an HTC Vive Pro (tracker, HMD, and wireless hand-held controller). The Vive was connected to a workstation with a 3.6 GHz Intel(R) Core(TM) i7-9700K CPU, with 16 GB of RAM, and with an NVIDIA GeForce GTX 2080 graphics card. The VEs were rendered in stereo at 90fps for each eye. The tracked physical space was a 4 m × 4 m empty space in our laboratory. Whenever a participant reaches the tracked area boundary, they use the controller to trigger a redirection operation that rotates the VE by 180°.

Participants and Conditions. For each of our three experiments, the participants were 16 graduate students, 10 male and 6 female, between 20 and 30 years old. 10 of our participants had used HMD VR applications before. Participants had normal and corrected vision, and none reported balance disorders. The participants were randomly divided into two groups of eight, a control group (CG) and an experimental group (EG). The same participants were involved in all three controlled user studies and in the perceptual experiments, which they performed on different days.

Virtual Environments and Tasks. In the first user study (VR1), participants were asked to search for seven persons wearing a specific shoulder badge in a crowded virtual *Art Gallery* (Figs. 12a and 1). The gallery covers a 42 m × 35 m

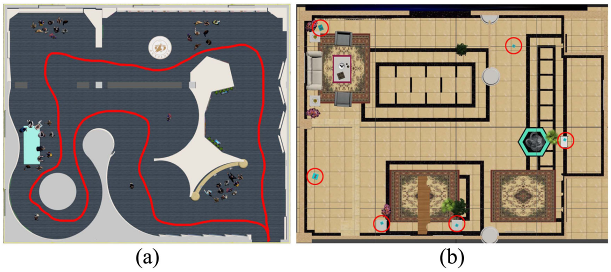


Fig. 12. (a) Top view of *Art Gallery* environment used in first study. The navigable route is a loop, shown as a red line. (b) Top view of *Indoor* environment used in the third study. The six cube locations are indicated in this figure with red circles for illustration purposes.

area. The participants had 15 minutes to find all seven persons, after which the experiment would time out, although no participant timed out in VR1. The system displayed the number of persons found, and would not count the same person more than once.

Before the trial, all participants were given three minutes to see the shoulder badge they had to search for, to learn how to trigger a virtual laser pointer with the controller to tag a person they found, and to familiarize themselves with the redirection mechanism. CG participants relied on conventional VR exploration. EG participants had available both coarse grain and fine grain multiperspective disocclusion effects, to handle the various types of occlusions in the VE. During an additional two minute tutorial, EG participants were shown how to trigger, select, deploy, and zoom the disocclusion effect using the controller and small head translations, and they were told to use the multiperspective effect for an efficient VE exploration as efficient as possible.

In the second user study (VR2), participants were asked to find the three trees in a *Chinese Garden* VE, and to count the apples in each tree (Fig. 13). The three trees had four, four, and five apples, respectively. The Chinese Garden covers a 120 m × 150 m area. CG and EG participants received the same tutorial as for VR1, except that they were instructed to report verbally to the experimenter the number of apples in each tree. The participants were given 15 minutes to complete the task. Participants who timed out were guided to the trees for them to complete the counting task. As for VR1, EG participants had both coarse and fine grain disocclusion effects available, and CG participants used conventional VR exploration.

The third user study (VR3), focuses on our occlusion quantification and localization. Participants were asked to search for small cubes in an *Indoor* VE (Fig. 12b). The *Indoor* environment covers a 40 m × 30 m area, and contained contained six cubes. Each cube is textured with a different two-digit number, which the participant has to report to the experimenter to confirm finding the cube. Both CG and EG participants benefited from multiperspective disocclusion effects, while EG participants also benefited from our occlusion quantification and localization method. The participants did not know the number of the cubes in the VE, and they had to perform an exhaustive search of the VE, keeping track of which parts had already been seen, and which parts were yet to be explored. The task finishes when the participant indicates to the experimenter that they considered the VE to be fully explored.

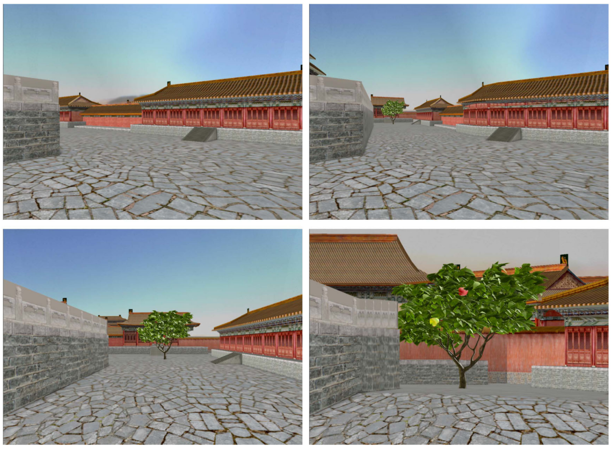


Fig. 13. *Chinese Garden* VE used in second study. Top: conventional visualization (left), and coarse grain disocclusion effect revealing the tree (right). Bottom: conventional visualization (left), and fine grain disocclusion + zoom in effect revealing the apples (right).

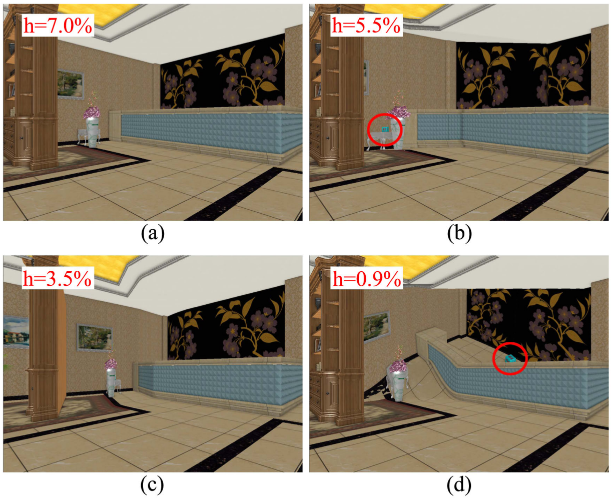


Fig. 14. VE exploration assisted by occlusion quantification for EG in VR3. The residual occlusion in the initial view (a) is reduced from 7.0 to 5.5 percent (b), then to 3.5 percent (c), and finally to 0.9 percent (d), revealing the cube hidden by the vase (b) and by the bar counter (d).

As shown in Fig. 14, EG participants are provided with a measure of the amount of residual occlusion. After finding the first cube (b), participants continue to search for cubes from the current location as the amount of residual occlusion indicates that the search is not yet complete. Once the floor hidden by the bar is disoccluded, the amount of residual occlusion is low and the participant moves on to a different location in the VE. On the other hand, CG participants had to gauge for themselves how much of the environment remains to be disoccluded.

Metrics. The performance of all participants was quantified with the following metrics: *viewpoint translation*, in meters, defined as the sum of the distance over which the user navigates the viewpoint from frame to frame, based on the HMD tracking data; this includes the lateral head translations needed to deploy and retract any disocclusion effect; *view direction rotation*, in multiples of full circle 360° rotations, defined as the sum of all frame to frame view direction rotations; since the user does not tilt or roll the head

TABLE 1
VR Exploration Efficiency Results

Metric	Study	Avg ± std. dev.		Reduction	p	Cohen's d	Effect size
		CG	EG				
ViewPoint translation, in meters	VR1	191 ± 52	88 ± 29	54%	≤ 0.001	2.31	Huge
	VR2	404 ± 121	220 ± 76	46%	0.004	1.71	Very Large
	VR3	c71 ± 11	46 ± 20	35%	0.013	1.43	Very Large
View direction rotation, as a multiple of full 360° rotations	VR1	76 ± 21	30 ± 10	61%	≤ 0.001	2.63	Huge
	VR2	117 ± 28	62 ± 21	47%	0.001	2.08	Huge
	VR3	36 ± 6	24 ± 7	34%	0.0027	1.82	Very Large
Redirections to stay within tracked boundary, in times	VR1	59 ± 10	27 ± 6	55%	≤ 0.001	3.64	Huge
	VR2	122 ± 36	63 ± 19	48%	0.0019	1.90	Very Large
	VR3	13 ± 3	8 ± 2	38%	0.0024	1.84	Very Large
Task completion times, in seconds	VR1	481 ± 144	317 ± 67	34%	0.0016	1.37	Very Large
	VR2	901 ± 203	672 ± 224	25%	0.0065	1.00	Large
	VR3	542 ± 133	361 ± 115	34%	0.0016	1.37	Very Large

much, this metric essentially gives the amount of view direction panning; *number of redirections*, in times, which are triggered by the user when they encounter the boundary of the physical space; *task completion time*, in seconds.

In addition, for VR1, VR2, and VR3, we counted the number of persons, apples, and cubes found by the participants, respectively. For EG participants in VR1 and VR2, and for EG and CG participants in VR3, we also counted the number of times each type of disocclusion effect was invoked. For fine grain disocclusion we also counted the number of times the zoom in operation was used.

7.2 VR Exploration Efficiency Results

We analyzed our data using the p value of a t-test [34], [35], and we quantified effect size using Cohen's d [36], [37]. Table 1 shows that our disocclusion method reduces viewpoint translation substantially. For VR1, viewpoint translation is reduced by 54 percent, which is significant ($p < 0.05$), and which corresponds to a huge effect. Whereas EG participants could search a crowd for the person with the badge from afar, CG participants had to get close to the crowd and then move around it, hence the viewpoint translation savings. Similar results were obtained for VR2 where EG participants could find trees more easily, without wasted navigation, and they could even count apples from afar. For VR3, all participants benefited from disocclusion effects, so the significant reduction in viewpoint translation is due to the residual occlusion quantification, which EG participants relied on to know to leave an area only once it was fully explored.

As in the case of view translation, the disocclusion effect reduces the total rotation significantly for all tasks (Table 1). When a CG participant searches for a person in a crowd or for apples in a tree, they have to pan the view direction a lot, whereas EG participants examine the crowds and the trees from multiple angles by revolving the camera through small head translations. As for translation, VR3 shows that residual occlusion quantification brings a significant reduction in view direction rotation. Table 1 also gives the number of redirections participants had to trigger to stay within the boundary of the tracked physical space. For VR1 and VR2, as expected from the substantial decrease in viewpoint translation, using the disocclusion effect significantly reduces the number of redirections for EG participants. VR3 shows that occlusion quantification brings an additional significant reduction in the number of teleportations. In terms of task completion times (Table 1), the disocclusion effects allowed EG participants to finish VR1 and VR2 significantly faster. For VR3, task completion time was significantly reduced when participants benefited from occlusion quantification in addition to disocclusion assistance, as the small cubes were found more reliably.

For VR1, participants were told there are seven people wearing badges, and the experiment would only stop once all people were found, which all participants did within the allotted 15min. For VR2, the participants were told that there are three trees in which they are supposed to count apples, for a total of 13 apples. If a participant did not find all trees in 15 min, the experimenter would guide them to the trees that were yet to be found. Three of the eight CG participants only found two of the three trees within the 15 min. All eight EG participants found all three trees. CG participants found on average 11.5 apples in the 15 min, without the experimenter's guidance for finding the missing tree, and 12.87 apples with guidance. EG participants found on average 12.75 apples, without ever needing guidance. For VR3, two of the eight CG participants missed one of the six cubes, whereas EG participants always found all six cubes, supporting our hypothesis that residual occlusion quantification enables a more thorough VR exploration. The difference is not significant ($p = 0.07$), but CG participants were taking longer and were exploring the same part of the VE multiple times (Table 1).

These increases in exploration efficiency and task performance were achieved through a frequent use of the disocclusion and zoom in effects. EG participants in VR1 used disocclusion effects 7 times on average, those in VR2 16.6 times, and participants in VR3 7.8 times. The zoom in effect was used, on average, 4.4, 5.3, and 4.8 times, for VR1, VR2, and VR3, respectively. Our method reduces the average navigation speed over VR1 and VR2 from 0.42 to 0.30 m/s, i.e., the participants complete the tasks faster while moving more slowly.

TABLE 2
Spatial Awareness Experiments

Task	Group	Avg ± std. dev.	p
Depth estimation in times	CG	0.00 ± 0.00	0.33
	EG	0.12 ± 0.33	
Layout perception in 30° increments	CG	0.36 ± 0.25	0.37
	EG	0.49 ± 0.25	
Pointing angle error in degrees	CG	2.76 ± 1.44	0.33
	EG	3.04 ± 1.75	
location repeating time in seconds	CG	67.90 ± 7.89	0.34
	EG	69.23 ± 6.79	

7.3 Perceptual Experiments

Our method improves VR exploration efficiency by giving the user the superpower of inspecting in detail a distant VE region from multiple view directions, from the current location, without first having to get close to the region. This disocclusion ability was designed to avoid losing spatial awareness, and to avoid simulator sickness.

7.3.1 Spatial Awareness

Our method shows the user a modified version of the VE, so we investigated whether and how much this VE modification impacts the user's spatial awareness of the VE. The good task performance for VR1, VR2, and VR3 is a first important indication that our disocclusion method does not reduce spatial awareness significantly. We further investigate the effects of our method on spatial awareness with four additional experiments. We compare our method to conventional VR visualization where the VE is explored sequentially, without deformation.

Relative Depth Estimation. In a first spatial awareness experiment, a participant located at P is shown a tree and is asked to judge the relative depth of apples in the tree. The tree has two apples, one yellow, and one red, and none are visible from P. The participant has to find the apples and to judge which one is closer to P. The experiment is repeated five times for each participant, with the two apples at different locations. CG participants (N = 8) walk close to the tree to disocclude the apples. EG participants (N = 8) use our fine grain disocclusion effect to find the apples from P; note that the two apples might not be visible simultaneously. Table 2 gives the average number of relative depth estimation errors for the two groups. The highest possible number of errors is 5. One of the eight EG participants made one error in one of their five tries, and the other seven EG participants made no errors. CG participants made no error, and this small difference between CG and EG is not significant. The gradual deployment of the multiperspective disocclusion effect, implemented as a progressive warp of the VE, allows the user to build a mental map of the original VE, the same such a map is pieced together iteratively through conventional sequential exploration.

Spatial Layout Estimation. In a second spatial awareness experiment (N = 16), participants were asked to estimate the relative position of the yellow and red apples within the tree. The positions were encoded using a clock metaphor Fig. 15. A marker was placed in front of the tree to indicate the six o'clock position (red circle in right image). The clock dial was in the horizontal plane, and the dial center coincided with the tree trunk. For each of five tries, the participant had to report the positions of the two apples with a pair of numbers, e.g., red apple at 3 o'clock and yellow apple at 5 o'clock. Table 2 gives the average spatial layout perception errors, over all participants and all apples, for each group. The maximum error is 6, corresponding to misjudging the position of an apple by 6 hours or 180°. Both CG and EG errors are small, with averages below 0.5 hours or 15°, so our experiment does not find a significant difference between EG and CG.

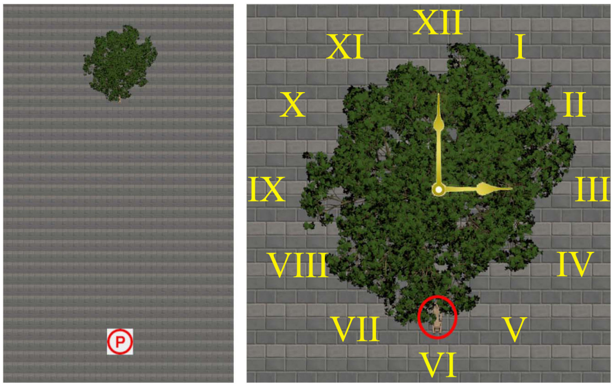


Fig. 15. Spatial layout estimation experiment. The participant is at location P (left) and has to indicate the relative position of two apples using a clock metaphor (right). The clock handles and the roman numerals are not part of the VE and they are shown here for illustration purposes.

Pointing Direction Accuracy. In a third spatial awareness experiment (N = 16), we tested the participant's ability to point correctly to a previously seen element of the VE [21]. For this experiment, the VE is a tree with a single apple. CG participants start out at an initial position P from where the apple is not visible. Then they are asked to walk around to find the apple in the tree, to return to the initial position P, and then to point in the direction of the apple that they had found. EG participants do not move away from the initial position P, but rather use our disocclusion effect to find the apple. Once they find the apple, the visualization reverts to a conventional visualization, and they are asked to point in the direction of the apple which is not visible anymore. For both groups, the pointing direction error is the angle between the direction a participant points to and the actual direction from P to the apple. The pointing errors are given in Table 2. The average error is about a third of a degree larger for EG, a difference that is not statistically significant.

Location Repeating Time. In a fourth spatial awareness experiment (N = 16), we tested the participant's ability to go back to a previous location. Using the *Indoor* VE used for task VR3, we asked the participant to find a cube. Once the cube was found, the participant was teleported to a location outside the room with the cube, and the participant was asked to go back to the location of the cube. The cube was removed after it was found such that the participant cannot search for it again, and the participant has to rely on their mental model of the VE to repeat the previous location. CG participants used conventional VR navigation, whereas EG participants used our multiperspective visualization. Table 2 shows the time CG and EG participants took to go back to

TABLE 3
Total Severity Score of SSQ

Study	Group	pre ± std. dev.	post ± std. dev.	post−pre ±std. dev.	p
VR1	CG	2.44 ± 2.82	12.07 ± 8.18	11.07 ± 5.98	0.06
	EG	6.87 ± 8.83	11.55 ± 8.18	4.68 ± 4.43	
VR2	CG	4.34 ± 8.07	9.14 ± 7.90	4.81 ± 7.49	0.98
	EG	5.65 ± 5.87	9.40 ± 10.59	4.93 ± 9.90	
VR3	CG	2.62 ± 2.64	17.66 ± 10.31	15.04 ± 9.82	0.09
	EG	7.99 ± 6.32	11.11 ± 5.19	3.12 ± 2.74	

the location where they had found the cube, with no statistically significant differences.

7.3.2 Simulator Sickness

We have investigated simulator sickness using a standard simulator sickness questionnaire (SSQ), which was administered before and after each of VR1, VR2, and VR3, and which was analyzed using the Total Severity (TS) score [38]. As shown in Table 3, the differences between the CG and EG pre to post TS increases are not significant. Furthermore, the participants were instructed to stop the experiment at any time should they feel uncomfortable, but all participants completed their experiments. We conclude that, at least in our three studies, the visual anchoring provided by the conventional rendering of nearby geometry succeeds at avoiding simulator sickness. Participants have no trouble understanding that the disocclusion effect is a special effect, akin to VE animation, complementary to first-person immersive exploration, yet integrated seamlessly, with visual continuity, into conventional VR exploration.

7.3.3 Presence

We have investigated the implications of our method on the user's sense of presence. Our work already shows that our disocclusion effect reduces the number of redirections, which are known to break the sense of presence. Furthermore, unlike, for example, static redirection work that permanently alters the geometry of the VE, the distortion of geometry implied by our method is both temporary and under the user's control, which should allow the user to build a correct spatial model of the VE.

We used the iGroup presence questionnaire (IPQ), with its four subscales: general presence (GP), spatial presence (SP), involvement (INV), and realism (REAL) [39], [40], [41]. Fig. 16 graphs the scores for CG and EG participants, averaged over all VR1, VR2, and VR3 tasks. No CG versus EG difference is significant, i.e., p is 0.175, 0.464, 0.480, and 0.479 for GP, SP, INV, and REAL.

8 Conclusions, Limitations, Future Work

We have presented a method for improving VR exploration efficiency through disocclusion assistance. In essence, our method allows the user to conveniently dispatch secondary viewpoints, which collect VE samples to which there is no line of sight, samples that are integrated seamlessly into the main user perspective. Our method detects VE regions with an abundance of fine-grain occlusions, which the user can then explore from afar, from multiple view directions, resulting in a significant increase in VR exploration efficiency, as confirmed in three user studies. The disocclusion effect is designed to avoid user disorientation, and our perceptual experiments together with standard SSQ and IPQ questionnaires confirm that this design requirement has been met. Furthermore, our method measures in real time the amount of residual occlusion from a given reference viewpoint, which helps the user explore a VE exhaustively. To the best of our knowledge, our method is the first to let the user ascertain that a VE has indeed been fully explored, as opposed to the conventional heuristic of trying to explore "as much as possible".

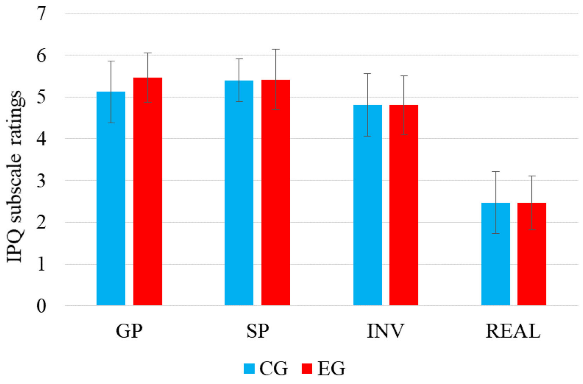


Fig. 16. Presence questionnaire for the four IPQ subscales. Error bars show the SD value to each subscale.

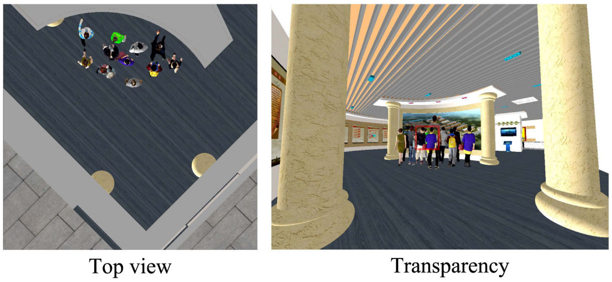


Fig. 17. Prior art top view and transparency occlusion management methods. Top view visualization does not show details of target, such as a specific shoulder badge, and transparency gives a confusing result due to the complex occlusion pattern.

We have compared our method to conventional VR exploration, which achieves disocclusion by relying on the user to translate the viewpoint to gain line of sight to parts of the VE hidden from the current location. Prior work has compared multiperspective visualization to other occlusion management methods such as top view and transparency [13], and it is reasonable to assume that the advantages of multiperpsective visualization found by these studies are likely to apply to our method. Fig. 17 shows that a top view visualization does not allow the user to easily find a person with a specific shoulder badge (or to see the faces of the people in the crowd). Similarly, disoccluding the crowd from afar using transparency results into a cryptic visualization due to the complex occlusion patterns. Of course, there are contexts where top view or transparency do have advantages, such as tracking a person in an empty environment, or seeing through a wall into an adjacent room, and future studies should continue to investigate which disocclusion management approach is to be preferred and when.

One limitation is that our method only addresses first order occlusions, i.e., occlusions created by visible occluders. Since the disocclusion effect is based on the discontinuities in a conventional depth buffer, our method will not address occlusions caused by occluders that become visible once the visible occluders are circumvented. For example our method can disocclude a tree that is visible from the reference viewpoint, but not a tree in a side garden alley that only becomes visible as the side alley is disoccluded. Future work could examine a recursive disocclusion effect construction.

Another limitation, and also a deliberate design choice, is that we abandon the concern of hiding from the user the deviation from the Euclidian geometry of the VE. Our goal is to provide superpowers to the user that are effective, easy to use, and that do not lead to discomfort, and our studies indicate that we reached this goal. Fooling users such that they do not notice their superpowers was not a design concern. Our method shows the location and size of the VE sub-volume that remains to be explored using red dots. Once the red dots are uncovered, they dissipate, allowing the user to see the newly revealed part of the VE. The red dots are similar to the glyphs used by other VR navigation assistance methods, which have been shown to be acceptable, as they do not affect the user's perception of their own position in the VE, and are similar to real world traffic signs. Future work could examine the ideal graphical representation of the space that is yet to be explored, to make sure that the red dots do not turn into a distracting game with false targets and false rewards.

Our method lets the user examine a region from afar, and avoid unnecessary VR navigation when that region proves to be of no interest. If the user does find the region to be of interest, they can navigate to the region and explore it in detail. Of course, our method works with prior art path visualization methods such that once the user finds an object of interest, the system guides the user to the object, for example with a line on the floor.

In the present work, the graph camera that computes the multiperspective visualization samples the region of interest with the leaf frustum. When the geometry behind the region of interest is far, the background of the multiperspective image changes quickly as the user revolves the secondary camera to disocclude the region of interest. As the user focuses on the region of interest, and as the stereo VR visualization does provide accurate depth cues, we did not find this to be of concern. Future work could explore alternative camera model designs that revert to the main user viewpoint beyond the region of interest. The current implementation groups all fine grain occlusions into one region of interest. Future work could explore grouping the fine grain occlusions into several regions of interest, disoccluding one at a time, or to limit the region of interest to the central part of the user view frustum.

Our visualization can be anchored by nearby geometry only when such geometry exists. Typically, our disocclusion method is needed when the region of interest is far from the user, so the ground plane in front of the user is visible and can serve the anchoring purpose. When the region of interest is close, the user can easily disocclude with a conventional visualization as the normal head translations and steps to the side are sufficient. For scenes void of near geometry, such as a planet examined from its outer space, our visualization is not anchored, and it reverts to an interface that couples head translation to spinning the planet. Because of the graph camera construction, our method does not allow viewing the region of interest from a direction that deviates by more than 90°. An alternative approach, such as the *world in miniature*, gives more freedom to the user, at the cost of visual discontinuity between the two perspectives of the compound visualization.

Finally, multiperspective visualization could help solve other VR problems, such as object selection or collaboration.

Acknowledgments

The authors would like to thank Meng-Lin Wu for his inspiring work on occlusion removal for HMD visualization for VR and AR. 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 under Grant Shenzhen 518000.

References

[1] B. Avery, C. Sandor, and B. H. Thomas, "Improving spatial perception for augmented reality X-ray vision," in *Proc. IEEE Conf. Virt. Reality 3D User Interfaces*, 2009, pp. 79–82.

[2] C. Coffin and T. Hollerer, "Interactive perspective cut-away views for general 3D scenes," in *Proc. 3D User Interfaces*, 2006, pp. 25–28.

[3] A. Dey, G. Jarvis, C. Sandor, A. Wibowo, and V. V. Mattila, "An evaluation of augmented reality X-ray vision for outdoor navigation," in *Proc. Int. Conf. Artif. Reality Telexistence*, 2011, pp. 28–32.

[4] O. Erat, W. A. Isop, D. Kalkofen, and D. Schmalstieg, "Drone-augmented human vision: Exocentric control for drones exploring hidden areas," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 4, pp. 1437–1446, Apr. 2018.

[5] T. Teshima, H. Saito, S. Ozawa, K. Yamamoto, and T. Ihara, "Vehicle lateral position estimation method based on matching of top-view images," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, pp. 626–629.

[6] 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. Inform.*, vol. 16, no. 6, pp. 696–703, 2012.

[7] J. Cui, P. Rosen, V. Popescu, and C. Hoffmann, "A curved ray camera for handling occlusions through continuous multiperspective visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1235–1242, Nov./Dec. 2010.

[8] R. Gupta and R. I. Hartley, "Linear pushbroom cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 9, pp. 963–975, Sep. 1997.

[9] H. Lorenz, M. Trapp, J. Döllner, and M. Jobst, "Interactive multiperspective views of virtual 3D landscape and city models," in *The European Information Society*. Berlin, Germany: Springer, 2008, pp. 301–321.

[10] C. Mei, V. Popescu, and E. Sacks, "The occlusion camera," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 335–342, 2005.

[11] S. Pasewaldt, M. Trapp, and J. Döllner, "Multiscale visualization of 3D geovirtual environments using view-dependent multiperspective views," *J. WSCG*, vol. 19, pp. 111–118, 2011.

[12] J. Yu and L. McMillan, "General linear cameras," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 14–27.

[13] L. Wang, J. Wu, X. Yang, and V. Popescu, "VR exploration assistance through automatic occlusion removal," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 5, pp. 2083–2092, May 2019.

[14] M.-L. Wu and V. Popescu, "Anchored multiperspective visualization for efficient VR navigation," in *Proc. Int. Conf. Virt. Reality Augmented Reality*, 2018, pp. 240–259.

[15] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin, "Moving objects in space: Exploiting proprioception in virtual-environment interaction," in *Proc. 24th Annu. Conf. Comput. Graph. Interactive Techn.*, 1997, pp. 19–26.

[16] J. Fung, C. L. Richards, F. Malouin, B. J. McFadyen, and A. Lamontagne, "A treadmill and motion coupled virtual reality system for gait training post-stroke," *CyberPsychol. Behav.*, vol. 9, no. 2, pp. 157–162, 2006.

[17] H. E. Makssoud, C. L. Richards, and F. Comeau, "Dynamic control of a moving platform using the CAREN system to optimize walking invirtual reality environments," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2009, pp. 2384–2387.

[18] E. Guy, P. Punpongsanon, D. Iwai, K. Sato, and T. Boubekeur, "LazyNav: 3D ground navigation with non-critical body parts," in *Proc. IEEE Symp. 3D User Interfaces*, 2015, pp. 43–50.

[19] S. Tregillus, M. Al Zayer, and E. Folmer, "Handsfree omnidirectional VR navigation using head tilt," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 4063–4068.

[20] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, "Point & teleport locomotion technique for virtual reality," in *Proc. Annu. Symp. Comput.-Hum. Interact. Play*, 2016, pp. 205–216.

[21] A. MacQuarrie, and A. Steed, "The effect of transition type in multi-view 360 media," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 4, pp. 1564–1573, Apr. 2018.

[22] M. P. J. Habgood, D. Moore, D. Wilson, and S. Alapont, "Rapid, continuous movement between nodes as an accessible virtual reality locomotion technique," in *Proc. IEEE Conf. Virt. Reality 3D User Interfaces*, 2018, pp. 371–378.

[23] Q. Sun, L.-Y. Wei, and A. Kaufman, "Mapping virtual and physical reality," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 64.

[24] Z.-C. Dong, X.-M. Fu, C. Zhang, K. Wu, and L. Liu, "Smooth assembled mappings for large-scale real walking," *ACM Trans. Graph.*, vol. 36, no. 6, 2017, Art. no. 211.

[25] E. Langbehn, F. Steinicke, M. Lappe, G. F. Welch, and G. Bruder, "In the blink of an eye: Leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 66.

[26] Q. Sun *et al.*, "Towards virtual reality infinite walking: Dynamic saccadic redirection," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 67.

[27] C. Sandor *et al.*, "Egocentric space-distorting visualizations for rapid environment exploration in mobile mixed reality," in *Proc. IEEE Conf. Virt. Reality 3D User Interfaces*, 2010, pp. 47–50.

[28] S. Bruckner and M. E. Groller, "Exploded views for volume data," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 5, pp. 1077–1084, Sep./Oct. 2006.

[29] J. Yu and L. McMillan, "General linear cameras," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 14–27.

[30] J. Cui, P. Rosen, V. Popescu, and C. Hoffmann, "A curved ray camera for handling occlusions through continuous multiperspective visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1235–1242, Nov./Dec. 2010.

[31] V. Popescu, P. Rosen, and N. Adamo-Villani , "The graph camera," *ACM Trans. Graph.*, vol. 28, no. 5, 2009, Art. no. 158.

[32] G. Cirio, P. Vangorp, E. Chapoulie, M. Marchal, A. Lécuyer, and G. Drettakis, "Walking in a cube: Novel metaphors for safely navigating large virtual environments in restricted real workspaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 4, pp. 546–554, Apr. 2012.

[33] L. Wang *et al.*, "Occlusion management in VR: A comparative study," in *Proc. IEEE Conf. Virt. Reality 3D User Interfaces*, 2019, pp. 708–716.

[34] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[35] D. W. Zimmerman, "Teacher's corner: A note on interpretation of the paired-samples T test," *J. Educ. Behav. Statist.*, vol. 22, no. 3, pp. 349–360, 1997.

[36] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Academic press, 2013.

[37] S. S. Sawilowsky, "New effect size rules of thumb," *J. Modern Appl. Statist. Methods*, vol. 8, no. 2, 2009, Art. no. 26.

[38] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *Int. J. Aviation Psychol.*, vol. 3, no. 3, pp. 203–220, 1993.

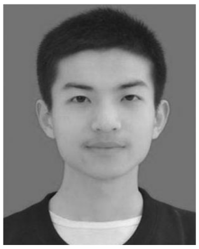
[39] T. Schubert, F. Friedmann, and H. Regenbrecht, "The experience of presence: Factor analytic insights," *Presence, Teleop. Virt. Environ.*, vol. 10, no. 3, pp. 266–281, 2001.

[40] H. Regenbrecht and T. Schubert, "Real and illusory interactions enhance presence in virtual environments," *Presence, Teleop. Virt. Environ.*, vol. 11, no. 4, pp. 425–434, 2002.

[41] T. W. Schubert, "The sense of presence in virtual environments: A three-component scale measuring spatial presence, involvement, and realness," *Zeitschrift für Medienpsychol.*, vol. 15, no. 2, pp. 69–71, 2003.

Jian Wu is currently working toward the PhD degree with the School of Computer Science and Engineering, Beihang University, Beijing, China. His current research focuses on virtual reality, augmented reality, and visualization.

Lili Wang received the PhD degree from the Beihang University, Beijing, China. She is currently a professor with the School of Computer Science and Engineering, Beihang University, and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her research interests include virtual reality, augmented reality, mixed reality, real-time rendering, and realistic rendering.



Hui Zhang is currently working toward the MA degree with the School of Computer Science and Engineering, Beihang University, Beijing, China. His current research focuses on virtual reality and visualization.

Voicu Popescu received the BS degree in computer science from the Technical University of Cluj-Napoca, Cluj-Napoca, Romania, in 1995, and the PhD degree in computer science from the University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, in 2001. He is currently an associate professor with the Computer Science Department, Purdue University. His research interests lie in the areas of computer graphics, computer vision, and visualization. His current projects include camera model design, visibility, augmented reality for surgery telementoring, and the use of computer graphics to advance education.