



# Automatic portals layout for VR navigation

Xiaolong Liu<sup>1</sup> · Lili Wang<sup>1,2,3</sup> · Yi Liu<sup>1</sup> · Jian Wu<sup>1</sup>

Received: 15 September 2022 / Accepted: 17 October 2023 / Published online: 4 January 2024  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

## Abstract

Portals layout in a large virtual scene can help users improve navigation efficiency, but determining the number and the positions of the portals has some challenges. In this paper, we propose two automatic virtual portals layout methods for efficient VR navigation. We first introduced a visibility importance-based method to determine the portals' positions and numbers for a given scene. To improve the walkability of the VR environment, based on the visibility importance-based method, we propose a simulated annealing-based portal layout method to optimize the portals' positions further. To reduce the number of reverse redirections in the navigation, we also proposed a real-time portal orientation determination algorithm to determine the orientations of the portals. We designed a user study to test the two methods we propose. The results showed that our methods made the VR navigation more efficient than the portals random layout and non-portal methods. Our methods achieved a significant reduction of task completion time, total viewpoint translation, and the number of reverse path redirections without increasing the scores of SSQ, IPQ, and task load.

**Keywords** Virtual reality · Navigation · Teleportation · Portal

## 1 Introduction

Navigation and exploration are fundamental interactions in Virtual Reality (VR). Real walking (RW) is the most natural way to navigate the virtual environment (VE) (Usoh et al. 1999; Suma et al. 2007). However, when the physical space that hosts VR applications is much smaller than the virtual environment, it becomes troublesome that users frequently

meet the physical boundary. The reverse path redirection method (Wang et al. 2019) allows the user to rotate the virtual scene 180 degrees by pressing a button. Then the user can continue to navigate the virtual scene by turning around. However, reverse path redirection may break the user's sense of orientation and causes discomfort when applied too many times. Redirected walking (RDW) (Steinicke et al. 2010) uses translation gain, rotation gain, and curvature gain to scale the transformation of the user's viewpoint. However, if the virtual scene is much larger than the physical scene, the existing redirected walking methods cannot solve the problem of walking in a small physical scene to explore a very large-scale virtual scene. Teleportation (Bozgeyikli et al. 2016; Linn 2017; Bhandari et al. 2018; Willich et al. 2020) is a more efficient navigation and exploration method in VR, allowing users to jump to locations pointed by a handheld controller. The disadvantage of teleportation is that users have to teleport multiple times to reach an invisible target region. What's more, it often leads to over-reliance on teleportation, which breaks presence (Liu et al. 2018).

Virtual portal is another way to navigate and explore in VR (Bruder et al. 2009; Steinicke et al. 2009; Freitag et al. 2014, 2017; Valve 2007; Wang et al. 2019). A pair of portals connect two remote locations. The user walks through the frame of the first portal (mirror, gateway, etc.) to reach

Lili Wang, Yi Liu and Jian Wu contributed equally to this work.

✉ Lili Wang  
wanglily@buaa.edu.cn  
Xiaolong Liu  
liuxiaolong186@buaa.edu.cn  
Yi Liu  
563127539@qq.com  
Jian Wu  
Lanayawj@buaa.edu.cn

- <sup>1</sup> State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China
- <sup>2</sup> Peng Cheng Laboratory, Xingke 1st, Shengzhen 518055, Guadong, China
- <sup>3</sup> Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Huayuan road, Beijing 100191, China

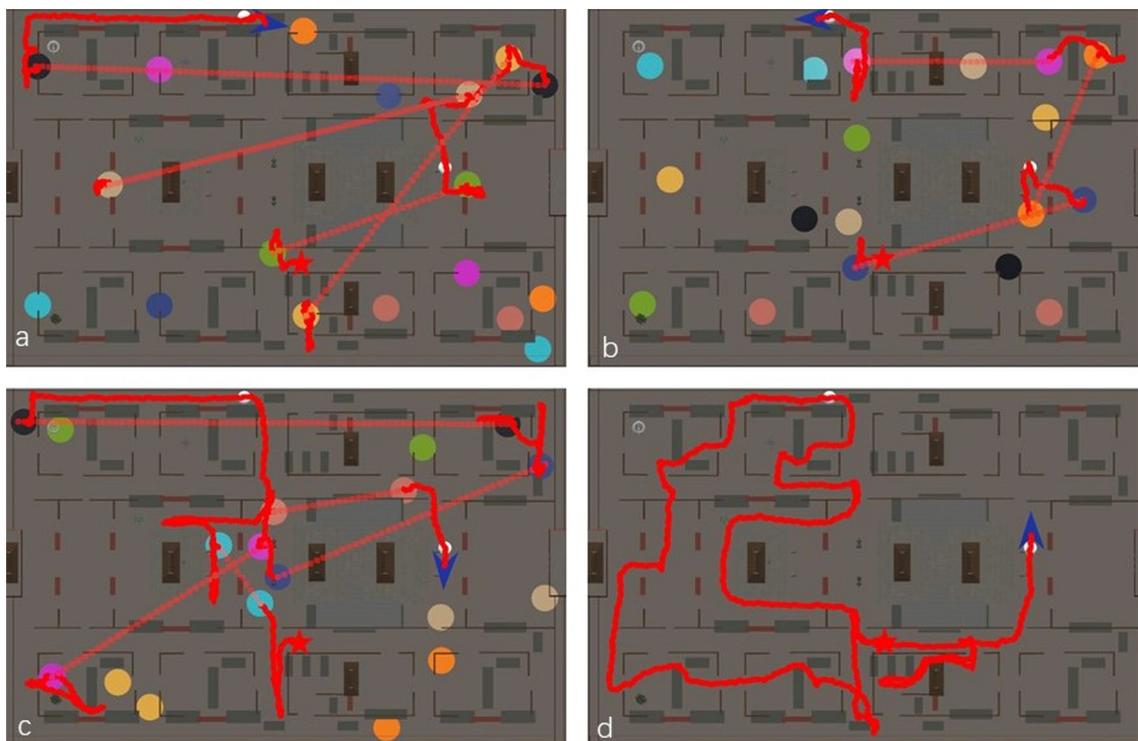
the other end outside the second portal. In large cities in real world, the layout of subway entrances is similar to the layout of portals in virtual scenes. When users take the subway, they only pay attention to the entrance and exit exits. The layout of the subway entrance needs to consider many complex factors, such as population distribution, land, construction technology, and so on. In large-scale virtual scenes, portals with similar functions like subway entrances are also required to provide users with portals that can enter from one entrance and exit quickly from another, thereby improving the efficiency of navigation and exploration. When placing the virtual portals in a large virtual scene, two key factors will affect the efficiency of user navigation and exploration: the first is the number of portals that need to be placed; the second is the positions and orientations of those portals.

In this paper, we propose two automatic virtual portals layout methods for efficient VR navigation and exploration. For a given scene, we first introduced a visibility importance-based portal layout method (VIPP) to determine the positions and the number of portals. Then we propose a simulated annealing optimized visibility importance based

portal layout method (SA-VIPP) to optimize the positions of the portals generated by the first method. Besides, we provided a real-time portal orientation determination algorithm to reduce the number of reverse redirections during navigation. We also designed a user study to test our methods. The results showed that our methods made the VR navigation more efficient than the portals random layout and non-portal methods. Our methods significantly reduced task completion time, total viewpoint translation, and the number of reverse path redirections without increasing the scores of SSQ, IPQ, and task load. Figure 1 shows the comparison of the Forbidden City scene in our user study with the two methods we propose, the random layout method and the non-portal method.

In summary, the contributions of our paper are as follows:

- A visibility importance based method to determine the positions and the number of portals automatically;
- A simulated annealing optimized visibility importance based portal layout method to optimize the positions of the portals;



**Fig. 1** The user trajectory visualization in the top view map of Forbidden City **a** using our visibility importance based portal layout method (VIPP), **b** using our simulated annealing optimized visibility importance based portal layout method (SA-VIPP), **c** using random portals layout method and **d** using no portal. Two white spheres in the Forbidden City indicate the locations the user needs to reach (they are not visible to users during the experiments). Circles of the same color represent a pair of portals. The red star represents user's start

position and the blue arrow represents the end position. The solid red line represents the natural walking trajectory of the user. The red dotted line represents the connection line of the used portal. The total viewpoint translation and the complete time of our VIPP are reduced by 57.9% and 48.8% for the random method, and 39.3% and 36.9% for the non-portal method. The reductions of our SA-VIPP are 68.7% and 70.1% for the random method, and 78.0% and 75.7% for the non-portal method (Color figure online)

- A real-time portal orientation determination algorithm to reduce the number of reverse redirections during navigation.
- We designed user studies to evaluate the efficiency of our methods.

## 2 Related work

Navigation and exploration of scenes are fundamental interactions in VR. Real walking in VR is to track the translation of people walking in physical space to explore virtual scenes. This is the most natural way in VR navigation and exploration. Suma et al. (2007, 2009) show that real walking in VR is beneficial to navigation and way-finding, and real walking can reduce VR sickness more than other methods. However, real walking has a limitation in VR: it does not allow users to explore large virtual environments in a limited physical space. Redirected walking (RDW) allows users to explore larger virtual scenes by manipulating the user's horizontal translation and rotation angles. Mary (2001) et al. first proposed the RDW method that makes the user feel like walking along a straight line while the user is actually walking along an arc unconsciously in VR. Steinicke et al. (2009) found the translation gain threshold ranged from  $-14$  to  $26\%$ , the rotation gain threshold ranged from  $-20$  to  $49\%$ , and the curvature gain radius was  $22$  m. Neth et al. (2012) et al. investigated the influence of walking speed on the sensitivity of curvature gain and found that the curvature gain depends on walking speed, and higher gains are suitable for lower walking speeds. Although redirected walking enables users to explore larger virtual scenes in a small physical space, they may introduce simulator sickness, interfere with spatial learning and memory, and cause a higher cognitive load than walking in the real world. For a more comprehensive understanding of RDW methods in VR, we recommend readers to read the survey (Nilsson et al. 2018).

Teleportation is a popular VR navigation and exploration method that users can use to jump to locations pointed to with a handheld controller. Bozgeyikli et al. (2016) proposed the point & teleport technology, where users point to any location in the virtual world, and the virtual viewpoint will be transmitted to that location. Linn (2017) proposed a method of gaze teleportation, the users can press the button and teleport to the point they are looking at. Bhandari et al. (2018) proposed a teleportation method called dash. This method can quickly but continuously move the user's perspective and retain some optical flow clues, which can better perform path integration. Willich

et al. (2020) proposed a foot-based teleportation method that can free the user's hand in VR. The disadvantage of teleportation is that users have to teleport multiple times to reach an invisible target region, and it often leads to over-reliance on the teleportation, which breaks presence (Liu et al. 2018).

Virtual portal is another VR navigation and exploration method that connects two remote locations. The portal can be placed in a large virtual scene as a door. The user walks through the frame of the portal object (mirror, gateway, etc.) to reach the other end outside the portal. The game Portal (Valve 2007) uses the portal for the first time. In the game, users can use the portal gun to open two interoperable portals, and the locations of the portals are all within the user's field of vision. Bruder et al. (2009) use the portal for the first time in VR, and the location of the transmission is the user selects in the miniature world. The user can reach the selected destination by walking through the portal. Steinicke et al. (2009) used the transitional environment and virtual portal as a means from the transitional environment to the actual virtual world. Freitag et al. (2014) and Liu et al. (2018) used portal to change the user's walking direction in the physical space, enabling users to always explore a large virtual space within a smaller physical space. Husung and Eike (2019) compared the technology in four aspects: presence, continuity, usability, and preference. They found that the orb and portal have the highest ratings. Wang et al. (2019) proposed a method of using a virtual portal to remove the occlusion. The user can see the occluded content through the portal to better navigate the virtual scene. Misztal et al. (2020) proposed a method to prevent twisting of HMD cables using portals.

Previous work mainly focused on the portal's functions, ignoring the relationship between the portal layout and the virtual scene structure. With more and more large virtual scenes in VR, how to layout portals according to the structure of these scenes becomes more and more important for VR navigation and exploration. Our method is the first to propose the fully automated portal layout methods based on virtual scene, so that users can explore and navigate in the virtual scene more effectively through these portals. The difference between our method and the state-of-the-art method is that our method automatically determines the number and layout of portals. This work is necessary to improve the efficiency of navigation and exploration in VR in large virtual scenes. We propose two methods. The first method considers the visual importance of the portal so that users can find the target in the scene faster. The second method uses a cost function to reduce the average

distance between any two points in the scene to place the portal more reasonably, making navigation and exploration more efficient. As far as we know, our method is the first to automatically optimize the number, location and orientation of virtual portals in a virtual scene.

### 3 Automatic portals layout method

In this section, we introduce two automatic methods to place the virtual portals in the virtual environment. The first one is the visibility importance based portal layout method (VIPP), and the second one is the simulated annealing optimized visibility importance based portal layout method (SA-VIPP). After that, we also propose a real-time portal orientation determination algorithm to determine the orientation of the portals, which can reduce the number of necessary reverse redirections.

#### 3.1 Visibility Importance based Portal layout method

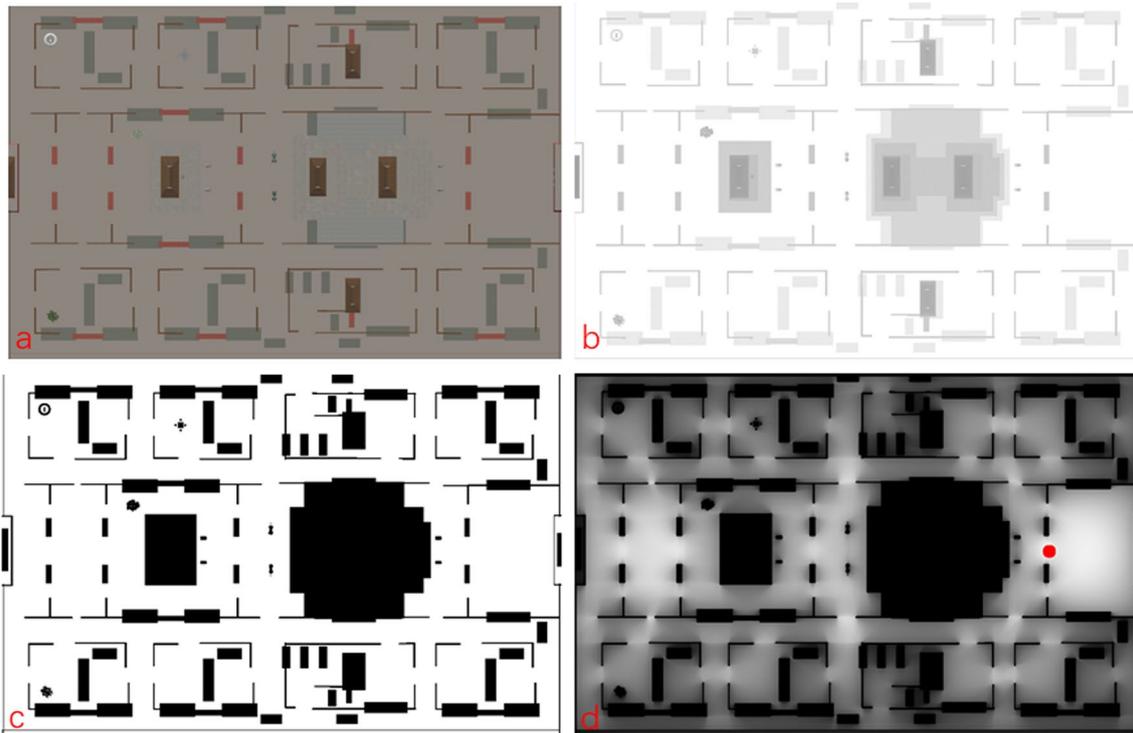
Walking efficiency and visibility are two important factors in virtual navigation and exploration. Walking efficiency can be measured by the length and time of the walking path to reach the target location. A shorter path length and time indicate higher navigation and exploration efficiency for the same starting and ending points. Visibility can be measured by the accumulated area of the scene that users can observe during the navigation and exploration. The larger the area, the more content the user can explore. For any given virtual scene, it is necessary to place each portal in an area of high visibility importance. Additionally, each pair of portals within the teleportation gates should be placed at a significant distance from each other, thereby further enhancing the user's walking efficiency and visibility. Therefore, this method calculates multiple importance maps and portal placement weight maps for the scene, adaptively determining the number and positions of portals. This portal layout aims to ensure that the visible area covers the majority of the scene, allowing users to quickly move to another area through portals in most parts of the scene. Additionally, each portal is placed in an area of good visibility, providing users with a better roaming perspective and enabling them to explore more virtual scene content. Finally, each pair of portals with the same color is placed at a significant distance from each other, ensuring that each teleportation provides a substantial benefit.

We calculate positions and number of portals for the scene according to the visibility importance of the scene. We first render the depth map of the scene from the top view orthographically and extract the walkable path map according to depth. A greedy algorithm is proposed to place the virtual portals based on visibility of the scene and the distance of each portal pair. This method is based on a greedy algorithm to iteratively place each pair of portals. In each iteration, importance maps of various virtual scenes and weight maps for portal placement are constructed and updated. These maps guide the placement positions and quantities of the portals. When the ratio of the visible area of the portals to the walkable region area exceeds a predefined threshold, we stop placing the portal.

The depth map from top view is generated by rendering the scene orthographically, as shown in Fig. 2b. Then we construct a portal layout weight map (*PPWM*) and the value in *PPWM* is from 0 to 1 (Fig. 2c). We use predefined depth threshold to initialize *PPWM* with 0 and 1, where 0 means unwalkable, 1 means walkable. After this, we adopt a greedy algorithm to initialize the positions and the number of the portals.

Two maps are constructed and updated to guide the algorithm. The first map is a scene visibility importance map (*VIM*), and it stores the value of the visibility importance from any walkable position to the entire scene. We place the first portal of a portal pair at the position with the highest visibility importance in *VIM* for each greedy iteration, as shown in Fig. 2d. Then we update the weights in *PPWM* by decreasing the weights of the position *pos* around the first portal *p* with Eq. 1. The visibility importance for each position in *VIM* can be computed using Algorithm 1. For a given initialized portal layout weight map *PPWM* and a predefined moving step *step*, we first initialize all values in *VIM* as 0 (line 1). Then we traverse each walkable position *pos* in *PPWM* (lines 2–3) and cast rays from *pos* to various directions (lines 4–5). For each ray, it moves ahead continually with *step* along *dir* until it arrives at the first unwalkable position (lines 6–7). During the moving process, the visibility contributions are continuously accumulated in *VIM* (lines 8–10). *o* is a predefined adjustment constant to avoid the visibility importance becomes too large. In our implementation, *o* is set to 1.

$$PPWM[pos] = \max(PPWM[pos] - 1/(distance(p, pos) + o), 0) \quad (1)$$



**Fig. 2** **a** is the top view of *ForbiddenCity* scene. **b** is the top view depth map of the scene. **c** is the portal layout weight map (*PPWM*). **d** is the scene visibility importance map (*VIM*). The red circle repre-

sents the first portal position which is placed according to the visibility importance (Color figure online)

### Algorithm 1 Compute Position Visibility Importance

---

**Input:** portal layout weight map *PPWM*, ray moving step *step*  
**Output:** visibility importance map *VIM*

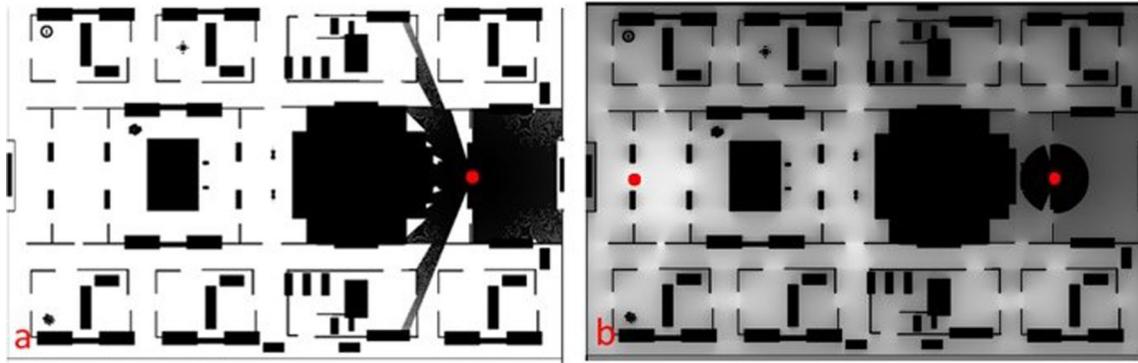
- 1:  $VIM = 0$
- 2: **for each**  $pos \in PPWM$  **do**
- 3:     **if**  $PPWM[pos] \neq 0$  **then**
- 4:          $pos' = pos$
- 5:         **for each**  $dir \in range(0, 2\pi)$  **do**
- 6:              $pos' = pos' + dir \cdot step$
- 7:             **while**  $PPWM[pos'] \neq 0$  **do**
- 8:                  $v = PPWM[pos'] / (distance(pos, pos') + o)$
- 9:                  $VIM[pos] = VIM[pos] + v$
- 10:                  $pos' = pos' + dir \cdot step$
- 11:             **end while**
- 12:         **end for**
- 13:     **end if**
- 14: **end for**
- 15: **return** *VIM*

---

The second map is a visibility and distance importance map (*VDIM*), and it stores the value of the visibility and distance importance from any walkable position to the entire scene. We place the second portal in a portal pair at the position with the highest visibility and distance importance in *VDIM* for each greedy iteration, as shown in Fig. 3. The visibility and distance importance for each position  $pos$  in *VDIM* can be computed with Eq. 2.

$$\begin{cases} VDIM[pos] = \alpha \cdot \bar{d}(pos) + (1 - \alpha) \cdot VIM[pos] \\ \bar{d}(pos) = \frac{1}{n} \cdot \sum_{q \in U(pos)} distance(p, q) \end{cases} \quad (2)$$

where  $p$  is the position of the first portal and  $\alpha$  is the weight of distance importance. In our implementation, we set  $\alpha$  to 0.5, which helps to keep the first portal far away from the second in the pair and keep the portals easy to reach.  $q$  is a position in the visible region of  $pos$ . First, the *VIM* is constructed to place the first portal in a location with the highest visibility importance. Then, Eq. 1 is used to update the weight of portal placement in the surrounding area of the first portal.



**Fig. 3** **a** is the updated *PPWM* after the first portal inserted. **b** is the visibility and distance importance map (*VDIM*)

After the second portal is placed, we remove the visible region of it in *PPWM*, and update *VIM* and *VDIM* in the same way to guide the layout of the next portal pair. This process is repeated until the ratio of the area of the visible region in the current *PPWM* to the area of the region area in the original *PPWM* is less than a predefined threshold.

### 3.2 Simulated annealing optimized visibility importance based portal layout method

In many VR applications, users need to arrive at target positions or pick up target objects in virtual scenes. The visibility importance based portal layout method needs to be improved. Instead of visibility, walkability, defined as the average walking path length between any two locations in the scene, becomes important. The shorter the length, the better the walkability of the scene. We first divide the virtual scene evenly into blocks, and we call each block a sub-region. To improve the walkability between each sub-region of the VR environment, we propose a simulated annealing based portal layout method to optimize the portals' number and positions further (Algorithm 2). The optimization starts from the layout we get with VIPP method. We considered the complex combination relationships between virtual scenes and multiple pairs of portals, and made improvements to the optimization process of the simulated annealing algorithm. In each optimization iteration, we designed a quality metric algorithm (Algorithm 3) for portal layout, which can measure the expected walking distance for users using portals.

#### Algorithm 2 Simulated Annealing Portal layout Optimization

---

**Input:** virtual scene's point to point matrix  $M$ , initialized positions of all portals  $P_{init}$ , rate of temperature drop  $r$ , initialized temperature  $T_{init}$ , minimum temperature  $T_{min}$ , max iteration number  $I_{max}$ , sampling step  $s$ , expected portals visible regions ratio  $R$

**Output:** portal optimized position  $P_{res}$

- 1:  $minCost = INF$
- 2:  $L = FloydShortestPathLength(M)$
- 3:  $P = P_{init}$
- 4:  $C_P = SimulatedAnnealingCost(P, L)$
- 5:  $i = 1$
- 6:  $T = T_{init}$
- 7: **while true do**
- 8:     **while**  $i > I_{max}$  **and**  $T > T_{min}$  **do**
- 9:          $P' = RandomDislayout(P, s)$
- 10:          $C'_P = SimulatedAnnealingCost(P', L)$
- 11:          $C = C'_P - C_P$
- 12:         **if**  $e^{-C/T} > random(0, 1)$  **then**
- 13:              $P = P'$
- 14:              $C_P = C'_P$
- 15:         **end if**
- 16:         **if**  $C'_P < minCost$  **then**
- 17:              $minCost = C'_P$
- 18:              $P_{res} = P'$
- 19:         **end if**
- 20:          $s = max(r \cdot s, 1)$
- 21:          $T = T \cdot r$
- 22:          $i = i + 1$
- 23:     **end while**
- 24:     **if**  $portalVisibilityRegionRatio(P_{res}) \geq R$  **then**
- 25:         **break**
- 26:     **end if**
- 27:      $randomAddTwoPortal(P_{res})$
- 28: **end while**
- 29: **return**  $P_{res}$

---

In Algorithm 2, we first initialize  $minCost$  to a large number (line 1), and calculate the shortest path length array  $L$  of any two points in the VE using the Floyd–Warshall algorithm (line 2), which  $M$  stores the initial path length of point to point in the VE. Then we calculate the cost of the current portal position  $P$  by algorithm 3 (line 4). We set the current temperature  $T$  to  $T_{init}$  (line 6). Next, we optimize the positions of all portals until the visible area ratio of all portals is greater than the predefined threshold  $R$  (lines 7–23). For each optimization procedure, we use a simulated annealing algorithm to optimize the positions of portals  $P_{res}$  (lines 8–20). The details of the simulated annealing algorithm are as follows: for each portal in  $P$ , we generate a new position  $P'$  randomly around the original portal (line 9); secondly, we calculate the cost  $C'_p$  of  $P'$  (line 10) and the changes of cost  $C$  (line 11); Thirdly we determine whether to accept  $P'$  based on the cost change  $C$  and temperature  $T$ . The probability of acceptance is inversely proportional to the change in cost and directly proportional to the current temperature (lines 12–15). After that, we record the current optimal portal position  $P_{res}$  and the minimum walking cost  $minCost$  (line 16–19). Then we update the temperature, step of simulated annealing, and the number of iterations (lines 20–22). Finally, if the ratio of the visible area of the current portal layout  $P_{res}$  to the visible area of the scene is greater than or equal to the expected ratio  $R$ , stop the iteration, otherwise, add two random portals to continue the iteration (lines 24–26).

**Algorithm 3** Simulated Annealing Cost

---

**Input:** portals  $P$ , shortest path length array  $L$   
**Output:** cost  $C$

```

1:  $D = 0$ 
2: for  $k = 1$  to  $size(P)$  do
3:    $p = P[k].x$ 
4:    $p' = P[k].y$ 
5:   for  $i = 1$  to  $size(L)$  do
6:     for  $j = 1$  to  $size(L)$  do
7:       if  $A[i][p] + A[p'][j] < A[i][j]$  then
8:          $A[i][j] = A[i][p] + A[p'][j]$ 
9:       end if
10:    end for
11:  end for
12: end for
13: for  $i = 1$  to  $size(L)$  do
14:   for  $j = 1$  to  $size(L)$  do
15:      $D = D + A[i][j]$ 
16:   end for
17: end for
18:  $C = D / size(L)$ 
19: return  $C$ 

```

---

In order to compute the cost of the simulated annealing algorithm (Algorithm 3), we initialize the shortest distance

between each position pair  $D$  to 0 (line 1). Then we traverse each portal pair of portals, and put the first and second portal in the pair into  $p$  and  $p'$  (lines 2–3). For each position pair in the shortest path length array  $L$ , we update the shortest path length after using the portals (lines 4–6). Then we sum up all the shortest path lengths between each walkable position pair as the cost (lines 13–17). Finally, we compute and return the cost  $C$  of the simulated annealing algorithm (lines 18–19). To gain a better understanding of the algorithm’s process, we recommend watching the video accompanying our paper.

### 3.3 Real-time portal orientation determination

Physical scene sizes are often smaller than virtual scene sizes, which leads to users needing to frequently reset their direction during VR walking. This disrupts the continuity of the roaming experience. To reduce the number of reverse redirections in the navigation after determining the positions and number of the portals, we propose a real-time portal orientation determination algorithm. The method dynamically optimizes the portal orientation based on the virtual scene and the user’s position and motion, thus implicitly guiding the user back to the center of the physical scene through the portal.

The orientation of the portal is set dynamically to face users to help them entering it easily. Then our portal orientation determination method is used to determine the orientation of the exit portal, in order to reduce the number of reverse redirections. Our method has two steps. The first step is a pre-process, which initializes the orientations of all portals based on the scene structure with a navigation possibility based method. The second step is to estimate the orientation offset for the second portal the user leaves in real time when user navigates.

The navigation possibility based method is shown in Algorithm 4. The algorithm takes positions of all portals  $P$ , virtual scene’s point to point map  $M$ , physical scene’s width  $S_w$  and height  $S_h$  as inputs, the output is the initialized orientations for all portals  $Ori$ . We first compute the shortest path length array  $L$  of any two positions in the scene by the Floyd–Warshall algorithm (line 1) and get the shortest path length array  $L'$  by updating  $L$  with portals  $P$  in the scene (line 2). Then diagonal length of the real scene  $R$  is computed based on the physical scene’s width  $S_w$  and height  $S_h$  (line 3). After this, We calculate the probability of navigating in different directions when the user leaves the second portal (lines 4–13). We first take each portal’s position  $p$  as the starting point, then move to the location  $p'$  along the direction  $dir$  and the length  $step$  (lines 4–7). Second, we compute the straight-line distance  $d$  and minimum walking distance  $d'$  between  $p$  and  $p'$  (lines 8–9). Third, we compute the walking complexity  $r$  of  $p$  and  $p'$  (line 10). Finally, we

calculate the probability of the user arriving at  $p'$  after leaving  $p$ , and sum it to the corresponding direction's possibility  $poss[dir]$  (line 11). After the direction possibility estimation, we traverse orientation  $ori$  of each portal  $p$  (line 15). For a given direction  $ori$ , we generate a semicircle with  $p$  as the center and  $R$  as the radius, facing the  $ori$ , and sample the direction  $dir$  on the semicircle (line 17). Then we sum up all directions' probability to  $I$ , which represents the probability that the user direction does not need to change when the user walks out of the second portal (line 18). At last, we record the orientation with the largest  $I$  as the initial orientation of  $p$  (lines 20–23).

**Algorithm 4** Probability based Portal Orientation Initialization

---

```

Input: portals  $P$ , virtual scene's point to point matrix  $M$ , physical scene's width  $S_w$  and height  $S_h$ 
Output: initialized orientation for all portals  $Ori$ .
1:  $L = \text{FlyodShortestPathLength}(M)$ 
2:  $L' = \text{updateShortestPath}(L, P)$ 
3:  $R = \sqrt{S_w^2 + S_h^2}$ 
4: for each  $p \in P$  do
5:   for each  $dir \in \text{range}(0, 2\pi)$  do
6:     for each  $s \in \text{range}(0, R)$  do
7:        $p' = p + dir \cdot s$ 
8:        $d = \text{distance}(p, p')$ 
9:        $d' = \text{shortestWalkDistance}(L', p, p')$ 
10:       $r = d / d'$ 
11:       $poss[dir] = poss[dir] + r / d'$ 
12:    end for
13:  end for
14:   $Ori[p] = I_{max} = 0$ 
15:  for each  $ori \in \text{range}(0, 2\pi)$  do
16:     $I = 0$ 
17:    for each  $dir \in \text{semiCircle}(p, ori, R)$  do
18:       $I = I + poss[dir] \cdot \cos|dir - ori|$ 
19:    end for
20:    if  $I > I_{max}$  then
21:       $I_{max} = I$ 
22:       $Ori[p] = ori$ 
23:    end if
24:  end for
25: end for
26: return  $Ori$ 

```

---

In the second step, we calculate the orientation offset  $\theta$  of the second portal that the user is passing through in real time with Eq. 3.  $D_m$  represents the walking direction of the user in the physical space, and  $D_c$  is the direction from the user to the center of the physical space. Figure 4 shows some examples of our real-time portal orientation determination method.



**Fig. 4** Real-time portal orientation determination. **a** is the top view of *ForbiddenCity*, and the optimization of portal direction is based on the scene structure in the red rectangle. In **b**, the white arrow indicates the initialized orientation of the portal. In **c**, the blue arrow indicates the direction of the user in the virtual scene, and the orientation of the second portal of the portal pair is optimized according to the user's direction in the physical space in **d**. **e**, **f** shows that when the user's position and direction are different in the physical space **f**, the result portal orientation also changes **e** (Color figure online)

$$\theta = \text{atan2}(D_m.y, D_m.x) - \text{atan2}(D_c.y, D_c.x) \tag{3}$$

## 4 User study

We designed a user study in two scenes to evaluate the effectiveness of our methods. The task load, presence, and simulator sickness are also evaluated.

### 4.1 User study design

#### 4.1.1 Participants

We have recruited 36 participants, 32 males and 4 females, between 20 and 30 years old (means 24). 20 of our participants had VR experience before. Participants had normal and corrected vision, and none reported vision or balance disorders. There are 2 control conditions ( $CC_1, CC_2$ ) and 2 experimental condition with our methods ( $EC_1, EC_2$ ). All participants are required to participate in experiments with all conditions.  $CC_1$  is with the no portal method.  $CC_2$  is with the random portal layout method.  $EC_1$  is with our VIPP.  $EC_2$  is with our SA-VIPP (with portal orientation determination).

### 4.1.2 Hardware and software setup

We used a set of HTC Cosmos VR HMDs with two hand-held controllers, allowing the user to reverse redirections at the VE. The HMDs were connected to a workstation with a 3.8 GHz Intel(R) Core(TM) i7-9800X CPU, 32GB of memory, and an NVIDIA GeForce GTX 2080 Ti graphics card. The tracked physical space hosting the VR applications is 4.2 m  $\times$  4.2 m. We developed a VR application for our experiments using Unity. The virtual scene is rendered at 90fps for each eye.

### 4.1.3 Hypotheses

Our method was designed to allow a user to touch all spheres. Thus, we formulate the following hypotheses:

**H1:** Users can touch all spheres faster with our portals layout ( $EC_1$ ) and  $EC_2$  compared to  $CC_1$  and  $CC_2$ .

**H2:** Users can touch all spheres by shorter total viewpoint translation with our portals layout ( $EC_1$ ) and  $EC_2$  compared to  $CC_1$  and  $CC_2$ .

**H3:** Users can touch all spheres by fewer number of reverse path redirection with our portals layout ( $EC_1$ ) and  $EC_2$  compared to  $CC_1$  and  $CC_2$ .

**H4** User task load with  $EC_1$  and  $EC_2$  is lower than with  $CC_1$  and  $CC_2$ .

### 4.1.4 Scene 1

The first scene is a 40 m  $\times$  40 m *Desertvillage* ( $S_1$ ), and 5 white spheres with a radius of 0.6 m are placed on the ground into the scene randomly (Fig. 5). We divide the virtual scene into uniform blocks and set each center of the block as a point. The spheres and portals are generated at these points. We divide the virtual scene into  $103 \times 103$  uniform blocks. In each task, the participants are required to navigate and explore the scene and touches all spheres. The spheres are disappeared after the participant touch them. The participants are also placed into the scene randomly at the beginning of the task. After the participant touched all spheres in the scene, the task is completed.

### 4.1.5 Scene 2

The second scene is a 50.6 m  $\times$  33.2 m *ForbiddenCity* ( $S_2$ ), and 2 spheres are placed into the scene randomly

(Fig. 6).  $S_2$  is divided into  $87 \times 56$  uniform blocks. The task is the same as  $S_1$ .

### 4.1.6 Procedure

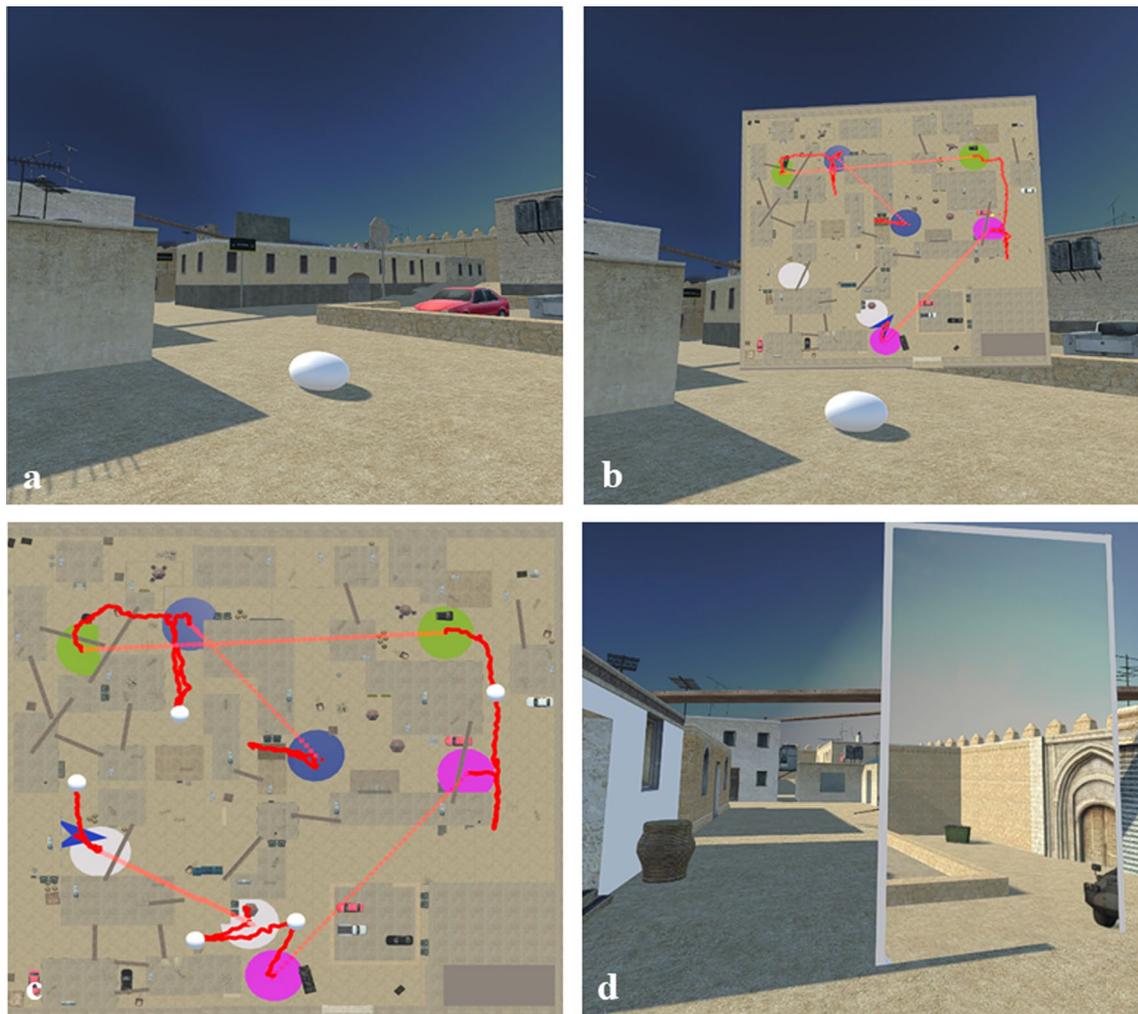
All participants performed the two scenes with all conditions  $CC_1$ ,  $CC_2$ ,  $EC_1$  and  $EC_2$  in random order. The minimum interval between the tasks was one day and the maximum interval was three days. For each scene, participants practiced for 3 min. When they pressed the start button, our system started recording all the objective metrics data. We told the participants that we would record and evaluate the task completion time, which indirectly encouraged them to complete the task as soon as possible. In the task, participants could view the mini-map by pressing the ‘view’ button on the handle. On the mini-map, the user could see the positions of all portals and his/her trajectory, but the user couldn’t see the positions of the spheres. We used the same color to represent a pair of portals. When the participant walked to the border of physical space, the participant could get a reminder and redirect  $180^\circ$  by pressing the ‘redirection’ button on the handle.

### 4.1.7 Metrics

(1) The task completion time, in seconds, which represents the time from the generation of the spheres in the scene to the completion of the task; (2) Total viewpoint translation, in meter, computed as the sum of all frame to frame head translations; (3) Number of reverse redirections the participant uses in the task. We also evaluated the perception with three subjective metrics: user task load, measured with the standard NASA TLX questionnaire (Hart 2006; Hart and Staveland 1988), user sense of presence in the virtual environment, measured with the standard Igroup Presence Questionnaire (IPQ) (Schubert 2003), and user simulator sickness, measured with the standard simulator sickness questionnaire (SSQ) (Kennedy et al. 2993).

### 4.1.8 Statistical analysis

For each metric, the values of  $EC_2$  were compared to those of  $CC_1$ ,  $CC_2$ ,  $EC_1$  respectively using a one-way repeated-measures ANOVA (Gelman 2005). First, the distribution normality assumption was verified using the Shapiro–Wilk test (Shapiro and Wilk 1965). All our data satisfied the normality assumption. Then the sphericity assumption is evaluated using the Mauchly test (Mauchly 1940). When the sphericity assumption is violated, a Greenhouse–Geisser correction is applied to the data. Then an overall ANOVA was conducted to investigate whether one can reject the null hypothesis that there is no statistically significant difference between the three conditions. When the null hypothesis was rejected ( $p < 0.05$ ), the



**Fig. 5** The first scene *DessertVillage* of our user study. **a** is one participant's view frame during navigation. **b** is the view with a mini-map observed when triggered by the participant, on which the portals and trajectory are marked. **c** visualizes the participant's trajectory in

the top view map. The blue arrow indicates participant's current location. **d** shows the participant's current view. The participant has just passed a portal, and he/she looks back at that portal

differences between the three pairs was analyzed with post-hoc tests, with a significance level lowered conservatively using the Bonferroni correction. For the time dependent variable we also quantified the size of the effect using Cohen's  $d$  (Cohen 2013). The  $d$  values were translated to qualitative effect size estimates of *Huge* ( $d > 2.0$ ), *Very Large* ( $2.0 > d > 1.2$ ), *Large* ( $1.2 > d > 0.8$ ), *Medium* ( $0.8 > d > 0.5$ ), *Small* ( $0.5 > d > 0.2$ ), and *Very Small* ( $0.2 > d > 0.01$ ). The statistical analysis was performed using the SPSS software (IBM <https://www.ibm.com/analytics/spss-statistics-software>).

## 5 Results

### 5.1 Task performance

#### 5.1.1 Task completion time

Table 1 gives the task completion time. The third column gives the average and standard deviation, and the fourth column gives the relative time cost reduction from  $*C$  to  $EC_2$ . The fifth to seventh columns provide statistical information about the difference between the  $EC_2$  and  $*C$ . Statistical significance is indicated by an asterisk.



**Fig. 6** The second scene *ForbiddenCity* of our user study. The orientation of the portal is optimized in real time to guide users back to the center of the physical space to avoid reverse redirections. **a** is the participant’s view in virtual scene. The participant is walking into the first portal of the portal pair, and the portal rotated dynamically to face the participant. **b** is the third view of the participant in physical space. The participant is detected walking to the physical boundary (red dotted area). The orientation of the second portal is determined to guide the participant to the center of the physical space marked

with the yellow dot. Arrow 1 represents the walking direction of the participant. In **a**, the view inside the portal shows what they will face after walking through the second portal, which is a wall. In order not to collide with the wall after exiting the second portal, participants turn around intuitively. **c** is the participant’s view in virtual scene after he/she turns around. **d** is the corresponding third view of the participant in physical space of **c**, in which the participant is guided to the center (the yellow dotted line). Arrow 2 shows the participant’s turning in physical space (Color figure online)

**Table 1** Task completion time, in seconds

Scene	Condition	Avg ± std. dev.	$(* C_i - EC_2) / * C_i$	<i>p</i>	Cohen’s <i>d</i>	Effect size
S1	<i>EC</i> <sub>2</sub>	201.56 ± 66.43				
	<i>EC</i> <sub>1</sub>	203.24 ± 59.56	0.8%	0.92	0.03	Very small
	<i>CC</i> <sub>1</sub>	352.56 ± 65.74	42.9%	< 0.001*	2.28	Huge
	<i>CC</i> <sub>2</sub>	299.06 ± 97.00	32.6%	< 0.001*	1.17	Large
S2	<i>EC</i> <sub>2</sub>	105.09 ± 24.05				
	<i>EC</i> <sub>1</sub>	167.97 ± 57.15	37.4%	0.001*	1.43	Very Large
	<i>CC</i> <sub>1</sub>	294.86 ± 96.19	64.4%	< 0.001*	2.71	Huge
	<i>CC</i> <sub>2</sub>	262.32 ± 124.21	59.9%	0.001*	1.76	Very Large

\*Statistically significant at *p* < 0.05

The sphericity assumption is verified: *p* < 0.001(*T*<sub>1</sub>), *p* < 0.001(*T*<sub>2</sub>) and *p* < 0.001(*T*<sub>3</sub>). After applying the Greenhouse–Geisser correction, the overall ANOVA

reveals significant differences between the five conditions: (*F*(2.1, 50.402) = 14.148, *P* < 0.001) for the *Desert-village* scene, (*F*(2.23, 49.03) = 56.24, *P* < 0.001) for the

*ForbiddenCity* scene. Post-hoc analysis reveals that  $EC_2$  were significantly shorter than for  $CC_1, CC_2$  for both scenes. Compared with  $CC_1, CC_2$  all two scenes, our two methods significantly improves the task time performance, and the effect size ranges from "Large" to "Huge". Compared with  $EC_1$  in S1, the SA-VIPP method is faster than the VIPP method, but not significantly. Compared with  $CC_1, CC_2$  and  $EC_1$  of  $S_2$ , the SA-VIPP method significantly improves the task time performance, and the effect size ranges from "Large" to "Huge".

### 5.1.2 Total viewpoint translation

Table 2 gives the total viewpoint translation. The sphericity assumption is verified:  $p < 0.001(T1)$  and  $p < 0.001(T2)$ . After applying the Greenhouse–Geisser correction, the overall ANOVA reveals significant differences between the five conditions: ( $F(1.92, 40.422) = 64.148, P < 0.001$ ) for the *Desertvillage* scene, ( $F(4.23, 39.03) = 36.24, P < 0.001$ ) for the *ForbiddenCity* scene. Post-hoc analysis reveals that  $EC_2$  were significantly shorter than for  $CC_1, CC_2$  for both scenes. Compared with  $CC_1, CC_2$  of all two scenes, our two methods significantly reduces the total viewpoint translation, and the effect size ranges from "Very Large" to

"Huge". Compared with  $EC_1$  in S1, the SA-VIPP method does not reduce the total viewpoint translation than the VIPP method. Compared with  $CC_1, CC_2$  and  $EC_2$  of  $S_2$ , the SA-VIPP method reduces the total viewpoint translation, and the effect size ranges from "Very Large" to "Huge".

### 5.1.3 Number of reverse path redirection

Table 3 gives the number of reverse path redirection. The sphericity assumption is verified:  $p < 0.001(T1)$  and  $p < 0.001(T2)$ . After applying the Greenhouse–Geisser correction, the overall ANOVA reveals significant differences between the five conditions: ( $F(4.4, 40.632) = 44.428, P < 0.001$ ) for the *Desertvillage* scene, ( $F(2.63, 23.13) = 76.32, P < 0.001$ ) for the *ForbiddenCity* scene. Post-hoc analysis reveals that  $EC_2$  were significantly shorter than for  $CC_1, CC_2$  for both scenes. Compared with  $CC_1, CC_2$  of all two scenes, our two method significantly reduces the number of reverse path redirections, and the effect size ranges from "Very Large" to "Huge". Compared with  $EC_1$  in S1, the SA-VIPP method reduce the number of reverse path redirections than the VIPP method, but it is not significant. Compared with  $CC_1,$

**Table 2** Total viewpoint translation, in meters

Scene	Condition	Avg ± std. dev.	(* $C_i - EC_2$ ) / * $C_i$	$p$	Cohen's $d$	Effect size
S1	$EC_2$	105.91 ± 21.70				
	$EC_1$	103.44 ± 25.51	-2.4%	0.70	0.25	Small
	$CC_1$	176.21 ± 41.57	39.9%	< 0.001*	2.55	Huge
	$CC_2$	149.97 ± 35.96	29.4%	< 0.001*	1.93	Very Large
S2	$EC_2$	42.20 ± 5.24				
	$EC_1$	66.68 ± 6.37	36.7%	< 0.001*	2.79	Huge
	$CC_1$	126.86 ± 34.24	64.4%	< 0.001*	3.25	Huge
	$CC_2$	91.13 ± 35.29	53.7%	< 0.001*	1.76	Very Large

\*Statistically significant at  $p < 0.05$

**Table 3** Number of reverse path redirection

Scene	Condition	Avg ± std. dev.	(* $C_i - EC_2$ ) / * $C_i$	$p$	Cohen's $d$	Effect size
S1	$EC_2$	29.86 ± 12.96				
	$EC_1$	31.00 ± 12.27	3.70%	0.74	0.09	Very Small
	$CC_1$	54.71 ± 7.74	45.4%	< 0.001*	2.33	Huge
	$CC_2$	46.29 ± 10.59	35.5%	< 0.001*	1.39	Very Large
S2	$EC_2$	11.00 ± 1.58				
	$EC_1$	22.75 ± 5.12	51.6%	< 0.001*	3.10	Huge
	$CC_1$	35.00 ± 9.98	68.6%	< 0.001*	3.36	Huge
	$CC_2$	27.25 ± 8.84	59.6%	< 0.001*	2.56	Huge

\*Statistically significant at  $p < 0.05$

**Table 4** The task performance of  $EC'_2$

Scene	Metric	Avg $\pm$ std. dev.	$(EC'_2 - EC_2) / EC'_2$	$p$	Cohen's $d$	Effect size
S1	CT	186.47 $\pm$ 33.05	-8.1%	0.30	0.29	Small
	VT	98.37 $\pm$ 11.86	-7.7%	0.12	0.43	Small
	NR	28.71 $\pm$ 8.28	-4.0%	0.70	0.11	Very Small
S2	CT	115.16 $\pm$ 17.51	8.7%	0.20	0.48	Small
	VT	45.97 $\pm$ 8.33	8.2%	0.15	0.54	Medium
	NR	13.0 $\pm$ 2.12	15.4%	< 0.01 *	1.07	Large

\*Statistically significant at  $p < 0.05$

$CC_2$  and  $EC_2$  of  $S_2$ , the SA-VIPP method reduces the number of reverse path redirections, and the effect size both are "Huge".

### 5.1.4 Evaluate the effects of our real time portal orientation optimization

To investigate the effect of our real time portal orientation optimization, we compared the task performance of  $EC_2$  and  $EC'_2$  (SA-VIPP without portal orientation determination). Table 4 gives the results of the task completion time (CT), total viewpoint translation (VT), and the number of reverse path redirection (NR) of  $EC_2$  and  $EC'_2$ . Statistical significance is indicated by an asterisk.

Compared with  $EC'_2$ ,  $EC_2$  has made significant improvements in the reductions of the task completion time, total viewpoint translation and number of reverse path redirection in  $S_2$ , and the effect size ranges from "Very Large" to "Huge". In  $S_1$ , compared with  $EC'_2$ ,  $EC_2$  has made significant improvements in the reductions of total viewpoint translation and the numbers of reverse path redirection, and the effect size ranges from "Medium" to "Very Large". Task completion time of  $EC_2$  are similar to those of  $EC'_2$ .

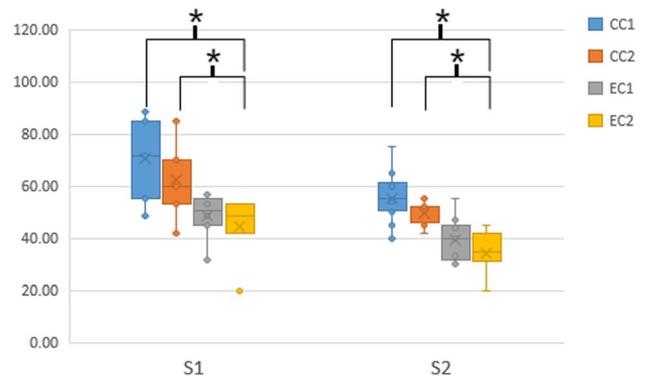
### 5.2 Perception

We have also investigated task load, presence, and simulator sickness using standard questionnaires.

We used Raw TLX (Hart 2006; Hart and Staveland 1988) to measure the task load. We averaged the scores of six Raw TLX task load problems. The sphericity assumption is verified:  $p < 0.001(T1)$  and  $p < 0.001(T2)$ . After applying the Greenhouse-Geisser correction, the overall ANOVA reveals significant differences between the five conditions: ( $F(1.68, 35.330) = 39.903, P < 0.001$ ) for the *Desertvillage* scene, ( $F(4.39, 31.23) = 96.19, P < 0.001$ ) for the *ForbiddenCity* scene. Post-hoc analysis reveals that  $EC_2$  were significantly shorter than for  $CC_1, CC_2$  for both scenes. Figure 7 shows the results of task load. Compared with all control conditions, the task load of our two methods is reduced significantly.

**Table 5** Igroup Presence Questionnaire data

Scene	Condition	GP	SP	INV	REAL
S1	$EC_2$	4.01 $\pm$ 0.24	3.21 $\pm$ 0.21	3.42 $\pm$ 0.32	2.65 $\pm$ 0.49
	$EC_1$	3.52 $\pm$ 0.39	3.37 $\pm$ 0.19	4.02 $\pm$ 0.27	2.97 $\pm$ 0.41
	$CC_1$	4.53 $\pm$ 0.23	3.35 $\pm$ 0.21	4.34 $\pm$ 0.55	2.98 $\pm$ 0.32
	$CC_2$	3.88 $\pm$ 0.45	3.72 $\pm$ 0.26	3.78 $\pm$ 0.29	2.97 $\pm$ 0.52
S2	$EC_2$	4.12 $\pm$ 0.45	3.95 $\pm$ 0.19	3.92 $\pm$ 0.21	2.59 $\pm$ 0.48
	$EC_1$	4.01 $\pm$ 0.27	3.88 $\pm$ 0.43	4.11 $\pm$ 0.12	2.89 $\pm$ 0.66
	$CC_1$	4.39 $\pm$ 0.26	3.91 $\pm$ 0.20	4.09 $\pm$ 0.23	2.62 $\pm$ 0.39
	$CC_2$	3.92 $\pm$ 0.23	3.90 $\pm$ 0.61	3.98 $\pm$ 0.20	3.02 $\pm$ 0.25



**Fig. 7** Participant task load per task and per condition

**Table 6** Simulator Sickness Questionnaire data

Scene	Condition	preAvg $\pm$ std. dev.	postAvg $\pm$ std. dev.	$p$
S1	$EC_2$	5.14 $\pm$ 1.07	5.30 $\pm$ 1.35	0.31
	$EC_1$	5.23 $\pm$ 2.7	5.35 $\pm$ 2.23	0.41
	$CC_1$	6.08 $\pm$ 2.05	6.20 $\pm$ 2.11	0.26
	$CC_2$	6.11 $\pm$ 1.22	6.38 $\pm$ 1.20	0.10
S2	$EC_2$	6.24 $\pm$ 1.29	6.52 $\pm$ 1.35	0.31
	$EC_1$	6.17 $\pm$ 1.17	7.22 $\pm$ 2.27	0.12
	$CC_1$	6.82 $\pm$ 1.25	7.11 $\pm$ 1.23	0.19
	$CC_2$	6.95 $\pm$ 1.16	7.25 $\pm$ 1.27	0.23

We measured users' sense of presence using the standard Igroup Presence Questionnaire (IPQ) (Schubert 2003). Table 5 shows our IPQ measurements broken down into the usual categories of general presence (GP), spatial presence (SP), involvement (INV), and realism (REAL). The experimental condition produces IPQ scores similar to both control conditions, and no difference is significant.

We used the standard simulator sickness questionnaire (SSQ) (Kennedy et al. 2993) (Table 6) to measure simulator sickness. The SSQ was administered before and after the experiment for each task and each condition. These differences are not statistically significant. We concluded that in these experiments, simulator was not significantly related to  $EC_2$  compared to  $CC_1$ ,  $CC_2$  and  $EC_1$ . No participant reported visual fatigue through related SSQ questions.

## 6 Discussions

The results support hypothesis  $H_1$ . The methods of placing with portals are more efficient than the method of placing without portals ( $CC_1$ ) and the method of placing with random portals ( $CC_2$ ). The possible reason is that portals allow the user to efficiently observe and switch between sub-regions of the virtual scene.

The results support hypothesis  $H_2$  and  $H_3$ . Compared to  $CC_1$  and the random portal layout method ( $CC_2$ ), our VIPP ( $EC_1$ ) and SA-VIPP ( $EC_2$ ) method's total viewpoint translation and number of reverse path redirection are significantly reduced. The possible reasons are that portals in  $EC_1$  can cover visible regions in the virtual scene to a great extent. For that, the user can find the target spheres quicker than other methods.  $EC_2$  improves the walkability between each sub-region of the virtual scene, making it much easier to find the target sphere.

The results support hypothesis  $H_4$ . The possible reason for the reduced task load is that with  $EC_1$  and  $EC_2$ , participants need less time, shorter total viewpoint translation, and fewer number of redirections to complete the same task.

In  $S_1$ , participants use the SA-VIPP layout method to find the sphere faster than the VIPP layout method, but it is not significant. In addition, the viewpoint translation and the number of redirections of the participants using the SA-VIPP layout method are not significant compared with the VIPP layout method. The possible reason is that  $S_1$  consists of many alleys and obstacles, making it difficult for participants to see the target spheres. In  $S_1$ , the portals' visible regions area is more important than walkability between each sub-regions of the virtual scene.

In  $S_2$ , compared to VIPP ( $EC_1$ ), SA-VIPP ( $EC_2$ ) allows participants to arrive at the spheres' location faster with less viewpoint translation and perform fewer reverse redirections, which have significant effects. The possible reason is that  $S_2$

consists of fewer obstacles and a wider field of vision. When participants find a sphere, they have to walk longer to catch it. Consequently, walkability plays a more important role in  $S_2$  than visibility. Compared with  $EC_2$  in  $S_2$ , the number of reverse redirects of  $EC_2$  is significantly reduced. However, in  $S_1$ , it is not significantly reduced. The possible reason is that most of the  $S_1$  are criss-cross alleys, which is basically within 4 m. And our physical space is 4.2 m  $\times$  4.2 m, so the number of times the user walks to the boundary of the physical space in  $S_1$  is relatively small than in  $S_2$ .

Based on the above discussion results, we give two suggestions for portal layout. First, select the SA-VIPP method with real-time portal orientation determination for scenes with a broader field of view and less occlusion; Second, select the VIPP method without real-time portal orientation determination for scenes with more occlusion and a narrower field of view.

In the portal layout method, the cognitive processes of users play a significant role in their interaction with the virtual environment and the teleportation experience. The placement of portals can influence users' cognitive processes in several ways:

### 6.1 Spatial awareness

The portal layout method aims to enhance users' spatial awareness by strategically placing portals in visible and easily accessible areas. This helps users develop a mental map of the virtual environment and facilitates their understanding of the spatial relationships between different areas.

### 6.2 Navigation and wayfinding

The placement of portals can influence users' navigation and wayfinding processes. By strategically positioning portals in locations that are easily identifiable and intuitive, users can quickly orient themselves and navigate through the virtual environment with greater ease. This can lead to more efficient and seamless movement between different areas.

### 6.3 Scene understanding

The cognitive process of scene understanding involves comprehending and interpreting the virtual environment and its various elements. The portal layout method can impact scene understanding by providing users with clear visual cues and landmarks associated with the placement of portals. This aids users in forming mental models of the virtual space and understanding the connections between different areas.

In relation to classical teleporting, the portal layout method differs in that it offers users a more immersive and visually guided teleportation experience. Rather than

instantly transporting users from one location to another, the method utilizes portals as intermediate points that users physically navigate through. This allows for a more continuous and interactive experience, promoting better scene understanding and spatial cognition.

## 7 Conclusions, limitations, and future work

We have proposed two automatic virtual portals layout methods for efficient VR navigation. We also introduced a real-time portal direction determination method to naturally guide users to the center of the physical space to avoid unnecessary reverse path redirections. Compared with no portal methods and the random portals layout method, our methods are proved to be more efficient. Our methods significantly reduce the number of task completion times, total viewpoint translation, and the number of reverse redirections without increasing the scores of SSQ, IPQ, and task load.

One limitation is that we use a button to show and close the mini-map. In this way, the user needs to use their short-term memory to decide how to walk in the virtual scene, leading to wrong decisions due to faulty memory after closing the mini-map. Another option is that the mini-map is always semi-transparent, floating in the corner of the user's field of view, but we found that when doing this, the map is too small to see clearly, and the virtual environment still be obscured by the mini-map. Therefore, one future work is to design a better user interface to use the portal. Another limitation is that all portals are currently displayed on the mini-map. If there are too many portals, users will feel confused, and affect their decision-making. Another limitation is that our portal is not integrated with the scene. This is because the outdoor scene has fewer objects and is relatively open. If it is an indoor application with a complex scene, we can fit the portal and the wall to produce a more natural feeling. Thus, one future work is to adaptively select part of the portals to display on the map according to the user's location, making it easier for users to make decisions. Our automatic virtual portal layout method needs to remove the ceiling manually. For multi-layer complex scenarios, such as a three-story shopping mall, removing the ceiling many times before using our method manually is more troublesome. Therefore, our future work is to use the navigation mesh as the basis for placing the portal in a multi-floor scenario and optimize the automatic layout of the portal to improve the efficiency of user navigation. Future work will provide standard metrics for the selection of the *SA – VIPP* method and *VIPP* method in different scenarios, such as  $\alpha$  in Eq. 1. Our method adopts a paired approach to the layout of portals. Future work is that all portals can act as exit portals,

allowing users to freely choose travel locations, thereby improving the efficiency of navigation and exploration in VR. To further explore the automatic placement of portals, we will compare manual portal placement with the automatic portal layout in the future to guide VR exploration and navigation. We will consider implementing the GUESS-18 questionnaire in future evaluations to further explore the gamification elements of the application. Our focus was primarily on investigating and optimizing the effectiveness and user experience of our proposed method. However, a comparison with the self-teleport option would provide a valuable perspective on the advantages and disadvantages of different teleportation methods. We will add including this comparison in future studies to provide a more comprehensive evaluation. Exploring the impact of scene understanding and cognitive aspects of teleportation is indeed an interesting direction for further research. We will incorporate a more detailed discussion on the cognitive processes and their relation to classical teleporting in our future work. Integrating Space Syntax or graph theory measures into our approach and evaluating their impact on human behavior is an intriguing idea. By considering measures that align with human cognition and space use, we may be able to further optimize the layout of portals and improve the overall user experience. We will carefully consider and explore the application of Space Syntax or graph theory measures in conjunction with simulated annealing in our future research.

**Acknowledgements** This work was supported by National Key R & D plan 2019YFC1521102, by the National Natural Science Foundation of China through Projects 61932003 and 61772051, by the Beijing Natural Science Foundation L182016, by the Beijing Program for International S & T Cooperation Project Z191100001619003.

**Data availability** Data is available on request from the authors.

## Declarations

**Conflict of interest** The authors declare no conflict of interest in this work.

**Ethics approval** Our experiments were approved by the Biology and Medical Ethics Committee of Beihang University.

**Consent to participate** All participants of the study agreed to the informed consent document. All data were anonymous to protect the participants.

## References

- Bhandari J, MacNeilage PR, Folmer E (2018) Teleportation without spatial disorientation using optical flow cues. In: Graphics interface, pp 162–167
- Bozgeyikli E, Raji A, Katkooi S, Dubey R (2016) Point teleport; teleport locomotion technique for virtual reality. In: Proceedings of

- the 2016 annual symposium on computer–human interaction in play, CHI PLAY '16. Association for Computing Machinery, New York, pp 205–216
- Bruder G, Steinicke F, Hinrichs KH (2009) Arch-explore: a natural user interface for immersive architectural walkthroughs. In: 2009 IEEE symposium on 3D user interfaces, pp 75–82
- Cohen J (2013) Statistical power analysis for the behavioral sciences. Academic Press, Cambridge
- Freitag S, Rausch D, Kuhlen T (2014) Reorientation in virtual environments using interactive portals. In: 2014 IEEE symposium on 3D user interfaces (3DUI), pp 119–122
- Freitag S, Weyers B, Kuhlen TW (2017) Efficient approximate computation of scene visibility based on navigation meshes and applications for navigation and scene analysis. In: 2017 IEEE symposium on 3D user interfaces (3DUI), pp 134–143
- Gelman A (2005) Analysis of variance. *Qual Control Appl Stat* 20(1):295–300
- Hart S (2006) Nasa-task load index (nasa-tlx); 20 years later. *Proc Hum Factors Ergonom Soc Ann Meet* 50:904–908
- Hart S, Staveland L (1988) Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Adv Psychol* 52:139–183
- Husung M, Langbehn E (2019) Of portals and orbs: an evaluation of scene transition techniques for virtual reality. In: Proceedings of mensch und computer 2019, MuC'19. Association for Computing Machinery, New York, pp 245–254
- IBM. Spss software. <https://www.ibm.com/analytics/spss-statistics-software>
- Kennedy R, Lane NE, Berbaum K, Mg L (1993) Simulator sickness questionnaire. An enhanced method for quantifying simulator sickness. *Int J Aviat Psychol* 3:203–220
- van Linn A (2017) Gaze teleportation in virtual reality. <https://api.semanticscholar.org/CorpusID:45248672>
- Liu J, Parekh H, Al-Zayer M, Folmer E (2018) Increasing walking in VR using redirected teleportation. UIST '18. Association for Computing Machinery, New York, pp 521–529
- Mary C (2001) Whitton Sharif Razzaque. Zachariah Kohn. Redirected walking, EUROGRAPHICS
- Mauchly JW (1940) Significance test for sphericity of a normal  $n$ -variate distribution. *Ann Math Stat* 11(2):204–209
- Misztal S, Carbonell G, Ganther N, Schild J (2020) Portals with a twist: cable twist-free natural walking in room-scaled virtual reality. In: 26th ACM symposium on virtual reality software and technology, VRST '20. Association for Computing Machinery, New York
- Neth CT, Souman JL, Engel D, Kloos U, Bulthoff HH, Mohler BJ (2012) Velocity-dependent dynamic curvature gain for redirected walking. *IEEE Trans Visual Comput Graph* 18(7):1041–1052
- Nilsson NC, Peck T, Bruder G, Hodgson E, Serafin S, Whitton M, Steinicke F, Rosenberg ES (2018) 15 years of research on redirected walking in immersive virtual environments. *IEEE Comput Graph Appl* 38(2):44–56
- Schubert TW (2003) The sense of presence in virtual environments: a three-component scale measuring spatial presence, involvement, and realness. *Z Medienpsychologie* 15(2):69–71
- Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). *Biometrika* 52:591–611
- Steinicke F, Bruder G, Jerald J, Frenz H, Lappe M (2009) Estimation of detection thresholds for redirected walking techniques. *IEEE Trans Visual Comput Graph* 16(1):17–27
- Steinicke F, Bruder G, Jerald J, Frenz H, Lappe M (2010) Estimation of detection thresholds for redirected walking techniques. *IEEE Trans Visual Comput Graph* 16(1):17–27
- Steinicke F, Bruder G, Hinrichs K, Lappe M, Ries B, Interrante V (2009) Transitional environments enhance distance perception in immersive virtual reality systems. APGV '09. Association for Computing Machinery, New York, pp 19–26
- Suma EA, Finkelstein SL, Reid M, Ulinski A, Hodges LF (2009) Real walking increases simulator sickness in navigationally complex virtual environments. In: 2009 IEEE virtual reality conference, pp 245–246
- Suma EA, Babu S, Hodges LF (2007) Comparison of travel techniques in a complex, multi-level 3d environment. In: 2007 IEEE symposium on 3D user interfaces
- Usuh M, Arthur K, Whitton MC, Bastos R, Steed A, Slater M, Brooks FP (1999) Walking gt; walking-in-place gt; flying, in virtual environments. In: Proceedings of the 26th annual conference on computer graphics and interactive techniques, SIGGRAPH '99, USA. ACM Press/Addison-Wesley Publishing Co., pp 359–364
- Valve Portal (2007). <https://store.steampowered.com/app/400/Portal/>
- Wang L, Jian W, Yang X, Popescu V (2019) VR exploration assistance through automatic occlusion removal. *IEEE Trans Visual Comput Graph* 25(5):2083–2092
- von Willich J, Schmitz M, Müller F, Schmitt D, Mühlhäuser M (2020) Podoportation: Foot-based locomotion in virtual reality. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp 1–14

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.