Multi-User Redirected Walking in Separate Physical Spaces for Online VR Scenarios

Sen-Zhe Xu ⬤, Jia-Hong Liu ⬤, Miao Wang ⬤, *Member, IEEE*, Fang-Lue Zhang ⬤, *Member, IEEE*, and Song-Hai Zhang ⬤, *Member, IEEE*

Abstract—With the recent rise of Metaverse, online multiplayer VR applications are becoming increasingly prevalent worldwide. However, as multiple users are located in different physical environments, different reset frequencies and timings can lead to serious fairness issues for online collaborative/competitive VR applications. For the fairness of online VR apps/games, an ideal online RDW strategy must make the locomotion opportunities of different users equal, regardless of different physical environment layouts. The existing RDW methods lack the scheme to coordinate multiple users in different PEs, and thus have the issue of triggering too many resets for all the users under the locomotion fairness constraint. We propose a novel multi-user RDW method that is able to significantly reduce the overall reset number and give users a better immersive experience by providing a fair exploration. Our key idea is to first find out the "bottleneck" user that may cause all users to be reset and estimate the time to reset given the users' next targets, and then redirect all the users to favorable poses during that maximized bottleneck time to ensure the subsequent resets can be postponed as much as possible. More particularly, we develop methods to estimate the time of possibly encountering obstacles and the reachable area for a specific pose to enable the prediction of the next reset caused by any user. Our experiments and user study found that our method outperforms existing RDW methods in online VR applications.

Index Terms—Redirected walking, multi-user, online VR, fairness.

I. INTRODUCTION

FACILITATED by the convergence between Internet technologies and Virtual Reality (VR), multiplayer online VR

applications are becoming a reality. The recent rise of Metaverse [1] has drawn the increasing interest of the Computer Graphics community to such kinds of immersive, interactive, and collaborative VR experiences. In a multiplayer online VR application/game, players are located in a separate physical space (e.g., their own home), but they can interact and play in a shared online virtual environment through the internet connection, as an example shown in Fig. 1.

For offering VR users easy movements in the virtual environment (VE), it is widely accepted that natural walking provides a much higher presence for users [2], [3], [4], [5], compared with using controllers or Omni-directional treadmills for locomotion. However, during natural walking, space limitation of the physical environment (PE) often limits the user's walking in a large VE. Redirected walking (RDW) [3] has been proposed to reduce the number of encounters with physical obstacles and enable users to explore larger VEs in relatively small PEs. The RDW methods usually use *subtle* and *overt* manners together to adjust the user's movement. The *subtle* RDW methods exploit the imprecision of the human motion perception and slightly modify the user's orientation or velocity during walking, enabling users to walk longer without noticing the inconsistency between their virtual and physical movements. Once the user is about to encounter an obstacle, the *overt* RDW technology then will suspend the virtual scene and reset the user's orientation in PE. The previous RDW methods aim to reduce the number of resets since resets cause pauses in the user's immersive experience.

However, the previous RDW algorithms only focused on the situation of a single VR user or multiple VR users in a shared physical space. Although previous RDW methods for multiple users can avoid collisions among users, each user's resets are considered irrelevant to others. Since in online VR apps/games multiple users are located in different physical environments, a fairness issue emerges when applying the previous RDW algorithms to multiplayer online VR apps/games. When one of the users has to be reset due to the physical obstacles, the other users can continue to walk in their PEs, making the users' relative positions in the VE change. As the users' tasks for an online VR application/game are often correlated, the users' relative virtual position changes will cause intolerable unfairness in online VR apps/games. For example, real-time strategy (RTS) games such as "Dota" or "Red Light, Green Light" for online VR will become unplayable due to users' relative position changes. Furthermore, since users' PEs are different, users within smaller PEs will have fewer opportunities
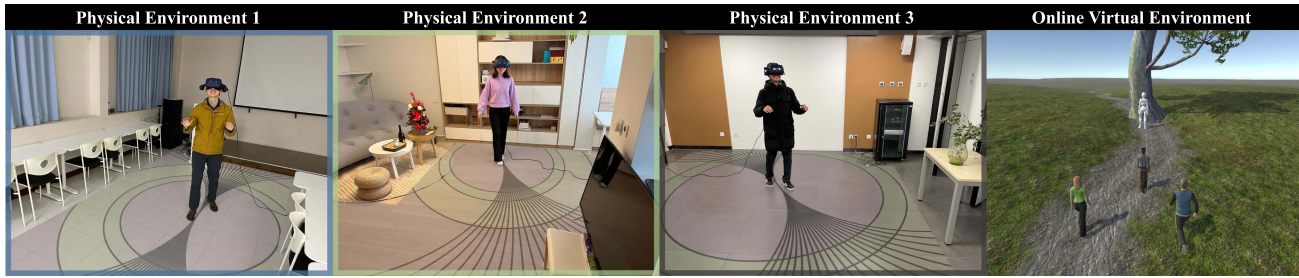
| Physical Environment 1 | Physical Environment 2 | Physical Environment 3 | Online Virtual Environment |

Fig. 1. Demonstration of multiple users in separate physical environments (PE), playing the same online VR game "Red Light, Green Light" in an online virtual environment (VE). For the fairness of locomotion, all users need to be paused and reset when any of the users encounter obstacles. Our redirected walking controller analyses the time of encountering obstacles and spatial extent of reachable area for each user in real-time and significantly reduces the number of resets for all the users by rigorously reasoning and maximizing the "bottleneck" time to the subsequent reset.

for moving during the online game, as they will be frequently suspended and reset their physical poses. For the fairness of the online VR apps/games, a good online RDW strategy must make the locomotion opportunities of different users equal, regardless of different physical environment layouts.

To achieve the locomotion fairness mentioned above in off-site multiplayer online VR apps/games, when a user gets stuck in the virtual space due to resetting, other users should not move in the virtual space either. A reasonable solution is that when some of the users trigger a reset, all users pause and reset to their better orientations synchronously and then resume the online game together. Rather than allowing the users to continue walking in PE while staying still in VE, this manner is less noticeable and more understandable to the users. More importantly, it can avoid wasting physical space. We refer to the manner that all players reset and resume together as the *locomotion fairness constraint*. Obviously, the *locomotion fairness constraint* would still increase the number of resets for every single user since other players may implicate each player to reset. Thus, we aim to reduce the number of resets of all the users for off-site multiplayer online VR apps/games under the *locomotion fairness constraint*.

In this paper, we propose the first practical and reliable RDW solution for off-site multiplayer online VR apps/games, which takes into account the *locomotion fairness constraint* and keeps the relative virtual positions unchanged after resets, as well as reduces the number of resets for all the connected users. Our key idea is to coordinates all the users' walking to maximize the time intervals between resets. Specifically, we propose practical yet effective methods to estimate the time range of encountering obstacles for each user's pose and the spatial extent achievable in a given time after resetting. By analyzing how long and how far users can walk after a reset we find the "bottleneck" user that may cause all users to the next reset. The reset occurrence will be put off by making the bottleneck user walk for the longest possible time. To prolong the time before the further subsequent reset, the users other than the bottleneck user will be reset and redirected to the their reachable and favorable poses at the same time, so all the user can have the longer possible walking distances/times after the next reset caused by the bottleneck user.

We conduct experiments in both simulation and real-world environments. We implement our work on the top of the open-source OpenRDW platform [6], and by our work we have

extended the OpenRDW platform to a "Meta" version and it's also the first RDW platform for the off-site multiplayer online VR applications. Our code will be released. The results of both simulations and real-world experiments demonstrate that our method outperforms all compared techniques.

II. RELATED WORK

A. Redirected Walking Strategies

The formulation of the RDW problem can be traced back to the year 2001 [3]. Many practical and far-reaching heuristic methods have been proposed since then. Razzaque et al. [7] proposed three general and widely used redirecting methods, namely, steer-to-center (S2C), steer-to-orbit (S2O), and steer-to-multiple-targets (S2MT), which respectively redirects the user to the physical center, the circular orbit around the physical center and the predefined waypoints in physical space. Hodgson et al. [8] improved S2C by setting temporary steering targets to make the user back to the tracking space center faster, which involves a new strategy named steer-to-multiple+center that combines S2C and S2MT by including physical centers to the waypoints in S2MT. Azmandian et al. [9] compared various RDW methods in different physical environment sizes and aspect ratios and confirmed that S2C outperforms most other methods, but S2O performs better when the user walks along a long straight virtual path.

The recent efforts on RDW are seeking more efficient solutions for redirecting users [10], [11]. The methods based on Artificial Potential Functions (APF) are influential among them. Thomas et al.'s APF (TAPF) [12] proposed Push/Pull Reactive, which utilizes the attractive or repulsive force calculated by the predefined APFs to push users away from obstacles or pull users towards desirable areas. Bachmann et al. [13] use APF to repel the user from obstacles and other users, allowing multiple users to walk in the same tracking space. Messinger et al. [14] proposed a follow-up work of Bachmann et al.'s APF, which refined the original work in the calculation of the force vectors and extended its adaptability in irregular tracking space. Dong et al. proposed Dynamic APF (DAPF) [15], which also uses dynamic artificial potential field to deal with multi-user scenarios, taking into account the joint repulsive forces of obstacles, users and the users' future states, as well as the gravitational force of the steering target.

Reinforcement learning-based methods are another rapidly raised branch of RDW methods. Lee et al. [16] proposed Steer-to-Optimal-Target (S2OT), which estimates optimal steering target from 25 predefined targets through reinforcement learning. Lee et al. [17] later proposed Multiuser-Steer-to-Optimal-Target (MS2OT), a reinforcement learning-based RDW method for multiple users, which improves S2OT by considering pre-reset actions, using more steering targets, and improving the reward function. Strauss et al. [18] proposed Steering via Reinforcement Learning (SRL), which applies the reinforcement learning technique to directly determine the locomotion gains when steering the users. Chang et al. [19] also proposed a method with plannability/versatility to redirect users using reinforcement learning.

Recently alignment-based RDW methods [5], [20] have been proposed to redirect users. Thomas et al. [21] introduced Reactive Environmental Alignment (REA), which transits the physical and virtual environments from misaligned states to aligned states. Williams et al. [5] proposed an alignment-based Redirection Controller (ARC) to make proximity to obstacles in PE matches that in VE. Williams et al. [20] also proposed to use visibility polygons to calculate the walkable spaces and align the visibility polygons of PE and VE with steering the user. Nevertheless, those algorithms perform alignment concerning obstacles in VE. Those algorithms will be less effective if VE contains no obstacles and the VE is infinite in size.

In addition to alignment-based methods, there are some other methods focusing on user experience in VE to redirect users. Some methods use virtual distractors to redirect users [22]. Adding visual [23] or auditory [24] virtual distractors in the VE may attract users' attention, thereby altering their walking trajectories to avoid physical collisions. But distractors may also affect the content and expression of the VE. There are also methods that try to avoid resetting the user at their points of interest in the VE to protect the user experience [25].

RDW methods for dynamic environments and multiplayer are also research hotspots. Besides the above methods involving multi-users, other works are dedicated to redirecting multiple users. Bachmann et al. [26] extended RDW controllers to support two-player VR apps by forecasting potential collisions. Azmandian et al. [27] explored three fundamental components of collision prevention for two-user VR scenarios and compared collision prevention strategies with PE subdividing. Z.Dong et al. [28] proposed to redirect multiple users by smoothly mapping virtual scenes to PEs and avoiding collisions with dynamic virtual avatars. T.Dong et al. [29] proposed an RDW method for multi-user navigation in large-scale virtual environments based on dynamic density.

However, the existing RDW methods are primarily for single or multiple users in the shared physical space with irrelevant locomotion in VE. Effective RDW strategy for multi-user in separated PEs while related in a shared online VE with locomotion fairness is still a challenge.

B. Locomotion Gains for Redirected Walking

To manipulate the mapping between the virtual and real-world imperceptibly, certain gains should be applied. Early methods proposed to use scaled translation [30] and augmenting motion in the user's intended direction [31] for redirecting the user. Steinicke et al. [32] then presented a taxonomy of redirection gains, including *translation*, *rotation* and *curvature* gains. *Translation* gains scale the user's positional displacement, making the user move faster or slower in the physical environment. *Rotation* gains apply a scale factor to the user's rotating speed, leading to a faster or slower turn. *Curvature* gains steer the user's direction when the user is walking forward. Langbehn et al. [33] found that when users travel along a curved path in the virtual space, their perception of redirection will become less sensitive, and they defined bending gain to describe this phenomenon. To avoid simulator sickness, the gains should be constrained within certain thresholds. It has been measured that in average case, translation gains should be in the range of 0.86 to 1.26, and rotation gains should be in the range of 0.8 to 1.49 [34], [35]. It has been measured and widely accepted by the recent works that the curvature gain should have a minimal radius of 7.5m [5], [8], [9], [12], [13], [14], [15], [17], [20], [26], [29].

III. METHOD

Suppose we have n users $U = \{u_1, u_2, \ldots, u_n\}$ participating the same online VR application in n separated physical spaces $\{E_1, E_2, \ldots, E_n\}$. Without loss of generality, we assign each user to a mission for each trial, where user u_i has to move towards a virtual target located in the VE continuously. When u_i reaches the target, a new virtual target for the user will appear to keep u_i moving. We denote the latest target of user u_i as w_i. The targets $W = \{w_1, w_2, \ldots, w_n\}$ for all users can be at different locations in the VE. Since the system specifies the users' present tasks, the RDW controller knows the current W. In the interests of fairness, all users need to be frozen and reset when some of them encounter obstacles in their physical spaces to ensure the relative positions of all users in the VE remain unchanged. Thus the number of resets will increase for every single user compared with the single-player game, as the users are implicated in resetting even when they do not encounter an obstacle. Therefore, our goal is to minimize the overall number of resets for all the involved off-site users. Before we introduce our redirection strategy for all the users, we first start by describing the estimation of *time before encountering obstacles* and *spatial extent of the reachable area with a given time* for a single user in the following sections.

A. Time Span Before Encountering Obstacles

In this subsection, we aim to figure out the time that a user could walk before encountering obstacles in PE when walking towards a target in VE. Given a user's current physical pose $P = (\boldsymbol{p}, \theta)$, whereas $\boldsymbol{p} = (p_x, p_y)$ is the user's position and θ is the orientation, the user can be steered to different physical positions imperceptibly with different curvature gains. The user's motion typically consists of a series of segments of straight walk toward the current virtual target. For one segment of walk, it begins when the user turns around or a reset ends, and finishes when the user reaches the target or triggers a reset. We choose to steer the user with a constant curvature gain during one segment of
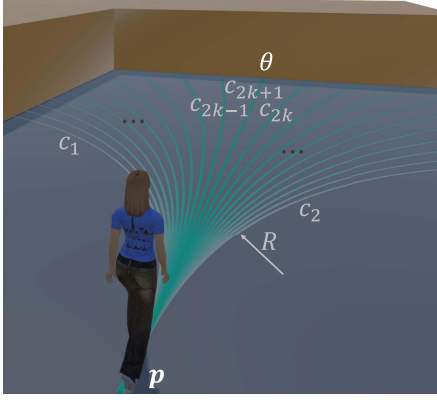
Fig. 2. The *inscribed circle strategy* for time span estimation of encountering obstacles.

Fig. 3. The *concentric circles strategy* for spatial extent estimation of the reachable area at given time T after a reset.

their walk for users' comfort. To resolve the time that could be taken to encounter obstacles, we propose to use an *inscribed circle strategy*. The idea is to scan the physical space with a series of radial curves from P, and estimate the time needed to hit an obstacle along these curves.

Taking θ as the tangent direction and p as the contact point, we can draw a series of mutually inscribed circles $C = \{c_i\}$ to the left and right of P with different radius $r_i = |\alpha_i|R$, as shown in Fig. 2. The equation of c_i can be written as:

$$\left\| \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - p_x \\ y - p_y \end{bmatrix} - \begin{bmatrix} 0 \\ \alpha_i R \end{bmatrix} \right\|_2^2 = (\alpha_i R)^2 \quad (1)$$

R is the radius of the curvature gain threshold and $|\alpha_i| \geq 1$ is a factor for generating different curvature gains. Note that we can steer the user to the left or right of p, the sign of α_i in equation (1) just reflects the direction to steer, i.e., $\alpha_i > 0$ means steering the user to the left, and $\alpha_i < 0$ means right. If we steer the user using one of the $r_i \geq R$ as the radius of the curvature gain with the direction indicated by α_i, the user's physical path will be coincident with the circle c_i. If α has k pairs of opposite finite values, it is easy to know that there are a total of $2k$ circles starting from p. To enable C to scan the PE uniformly, we make α_i grow slowly when i is close to 1, and diverge to infinity as i increases to $2k$. Specifically, we make our α_i as:

$$\alpha_i = \frac{(-1)^{(i-1)}}{\cos\left(\frac{\pi}{2k} \times \lfloor \frac{i-1}{2} \rfloor\right)} \quad (1 \leq i \leq 2k) \quad (2)$$

We take circle c_i as a candidate path and intersect it with all potential obstacles in PE, and get the length l_i from p to the nearest obstacle along c_i. If c_i happens to have no intersection with all obstacles (this rarely occurs unless the PE is extremely large), we designate l_i as infinity which means the user can walk any length along this path. Suppose the user's walking speed in the virtual space is v (v can be different for different users), the permitted time for a user walking along path c_i without considering translation gain is:
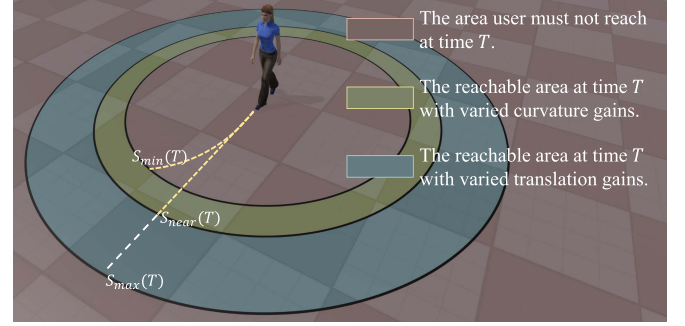
$$t_i = \frac{l_i}{v} \quad (1 \leq i \leq 2k) \quad (3)$$

It is worth noting that when $\alpha \to \infty$, the circle will approach a straight line, and it is equivalent to using no curvature gain when steering the user. We consider the straight path as c_{2k+1} and calculate its intersection with the obstacles to obtain the walking time with no curvature gain as t_{2k+1}. Considering that we can also speed up or slow down a user's physical speed with a translation gain g_t, the possible range of walking time t_i along c_i is further scaled to $[t_i g_t^{min}, t_i g_t^{max}]$. As long as k is large to make the candidate paths C sufficiently dense, we can approximate the maximum time that the user could have before encountering obstacles from a starting pose P as:

$$T_{max}(P) = \max_i \left(t_i g_t^{max} \right) \quad (4)$$

Considering that the user can voluntarily reset before encountering the obstacles, the RDW controller can make the user to reset at any time between $t \in [0, T_{max}(P)]$ when the user is starting from pose P.

B. Spatial Extent of Reachable Area With Given Time

When the user is about to reset, predicting the user's reachable area at time T after that reset benefits our RDW controller to reset and redirect the user to a more favorable position, thus helping to minimize the number of common resets for all users. Assuming that a reset just happened at the user's position p, we then solve the problem of the spatial area that a user can be possibly navigated to at a given walking time T after that reset.

We propose an intuitive and effective way called *concentric circles strategy* to solve this problem, as shown in Fig. 3. We found that since a user can face any needed direction after a reset, all their reachable positions must fall in a particular circle area centered at the user's current position p without considering the obstacles. We thus inspect the displacement s from a position to the user's current position p to determine whether it is reachable for the user at time T after a reset.

Let us consider the upper bound of the possible displacement S_{max} first. To make the user walk to the position with the farthest displacement, the controller has to apply the translation gain with the minimum value g_t^{min} to make their physical walking speed faster. Meanwhile, the user must be kept walking straight, and the controller should give no curvature gain; otherwise, their trajectory will be bent. Therefore, the maximum possible displacement of the user at time T after reset can be calculated

as:

$$S_{max}(T) = \frac{v \cdot T}{g_t^{min}} \tag{5}$$

Where v is the speed in the virtual space. Likewise, we can make the user walk for a shorter distance S_{near} at the given time T by simply adjusting the translation gain to the opposite threshold g_t^{max}:

$$S_{near}(T) = \frac{v \cdot T}{g_t^{max}} \tag{6}$$

Conceivably, any position with a displacement of $s \in [S_{near}(T), S_{max}(T)]$ from p can be reached at time T after resetting. It can be done by resetting the user to the corresponding direction and adjusting the translation gain to $g_t = v \cdot T/s$. To further verify the navigability for a queried position q in the PE, we test whether the straight line connecting q and the user's current position p intersects with any obstacles.

Nevertheless, $S_{near}(T)$ is not the minimum reachable displacement at time T. Recall that we can still shorten the reachable displacement through the curvature gain anyway. By applying the maximum curvature gain, i.e., the radius of the threshold R, we can then get an even smaller reachable displacement $S_{min}(T)$ by bending the straight path to its maximum extent:

$$S_{min}(T) = 2R \sin \frac{v \cdot T}{g_t^{max} \cdot 2R} = 2R \sin \frac{S_{near}(T)}{2R} \tag{7}$$

S_{min} is the chord length corresponding to an arc of length S_{near} on a circle of radius R. Having a displacement S_{min} means walking jointly using the maximum curvature gain and minimum translation gain in time T. Owing to the fact that both gains have reached their bounds, $S_{min}(T)$ is the minimum displacement that can be reached using the given time T.

For any position q with a displacement $s \in [S_{min}(T), S_{near}(T))$ away from p, we can steer the user to let them reach q at time T by relaxing the curvature gain radius to a suitable value. Suppose the proper curvature gain radius for arriving at q is R', the relationship between R' and s is:

$$2R' \sin \frac{v \cdot T}{g_t^{max} \cdot 2R'} = s \quad (s \in [S_{min}(T), S_{near}(T)), R' \geq R) \tag{8}$$

Equation (8) is a *transcendental equation* with respect to R' and there is no algebraic solution. However, we can just re-use the k different $\{r_i\}$ in Section III-A to approximate the optimal solution. Thanks to the fact that the left side of Equation (8) is monotonically increasing in its domain of definition, the optimal solution for R' can be quickly found by binary searching among $\{r_i\}$. Once R' is found, we further verify whether the arc path could be blocked by any obstacles in the PE. We connect q and p with the arc of radius R' and check if the arc intersects with the obstacles.

Note that adjusting the translation gain is also a potential way to reach q when $s \in [S_{min}(T), S_{near}(T))$. However this manner can't guarantee all $s \in [S_{min}(T), S_{near}(T))$ are covered in geometry. Moreover, the user's sickness exacerbates when a larger curvature gain is applied while their physical walking is
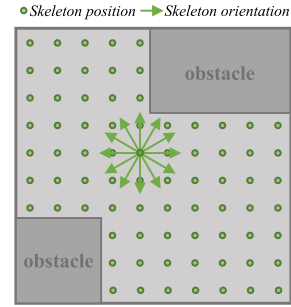

Fig. 4.   The illustration of the *skeleton positions* and the *skeleton orientations*.

accelerated. Thus, we only fine-tune the curvature gain to find candidate paths in this case.

By above steps, we have found paths with less sickness to reach all possible positions from p with displacements $s \in [S_{min}(T), S_{max}(T)]$ at time T. We have also verified the navigability of the path from p to a specified position. We denote the set of all positions reachable from p at time T as $\varphi(p, T)$.

C. The Multi-User Online RDW Controller

To coordinate the redirection of all users for a fewer total number of resets, we first analyze the current state of each user in terms of the remaining time to their target and their maximum time to encounter an obstacle. We then find the bottleneck user that needs to be reset in the shortest time, and reset all the users to their reachable poses that allow them to walk for their longest possible distances once a common reset is triggered.

Skeleton Poses As the number of users' poses in the physical space is infinite, we abstract the physical space with a series of discrete *skeleton poses* to represent the possible states in it. We divide a given physical space E into squares of side lengths δ and use the centers of the squares p_c (except those in obstacles) as the *skeleton positions* $\chi(E) = \{p_c\}$. We define λ *skeleton orientations* as $\Theta = \{2\pi i/\lambda \mid i = 1, 2, \ldots, \lambda\}$. The *skeleton positions* and the *skeleton orientations* then form all the *skeleton poses* $\Gamma(E) = \{(p, \theta) \mid p \in \chi(E), \theta \in \Theta\}$ for the given space, as shown in Fig. 4.

We perform an offline preprocessing by computing the maximum walking time before encountering obstacles $T_{max}(P)$ for all *skeleton poses* $P \in \Gamma(E)$. After that, we measure each *skeleton position* p_c in $\chi(E)$ with two indicators:

$$L(p_c) = \max_{\theta \in \Theta} T_{max}((p_c, \theta)) \tag{9}$$

$$H(p_c) = \lambda \left(\sum_{\theta \in \Theta} \frac{1}{T_{max}((p_c, \theta))} \right)^{-1} \tag{10}$$

We call $L(p_c)$ the escapability of position p_c, which represents the longest walking time a user can travel from position p_c if the user is just reset to the optimal direction at p_c. We call $H(p_c)$ the safety of position p_c, which represents the harmonic mean of the maximum walking times in all *skeleton orientations* of p_c, and indicates the overall safety in case the user turns to an unknown direction at p_c. Here we use the harmonic mean instead of the arithmetic mean to make $H(p_c)$ more sensitive to

the directions with a shorter maximum time to encountering an obstacle.

User State Analysis

The user's movement in VE is typically composed of walking toward the current virtual target and turning around to the next virtual target. One segment of immersive straight walking begins when the user have turned around or finishes a reset, and ends when the user reaches the target or a reset occurs. For the users' immersion, our RDW controller redirects each user with constant gains during one segment of their straight walking, and only activates the decision-making process for new gains when a common reset occurs or the user has turned around.

As the user's walking speed in VE is known to the RDW controller, we can estimate the remaining time τ_i for each user u_i to their virtual target w_i. We suppose that u_i will keep walking in VE within time τ_i. Meanwhile, we calculate the maximum time for user u_i to encounter obstacles in all *skeleton orientations* $\theta \in \Theta$ at their current position $\boldsymbol{p}_i$, and find the largest maximum time to encounter obstacles among all *skeleton orientations* as:

$$\Pi\left(u_i\right) = \max_{\theta \in \Theta} T_{max}\left((\boldsymbol{p}_i, \theta)\right) \qquad (11)$$

$\Pi(u_i)$ implies the upper bound of u_i's walking time after being reset to the best orientation at their current position $\boldsymbol{p}_i$. If $\Pi(u_i) \geq \tau_i$, the user u_i could get to the target w_i without encountering obstacles again after the current reset, we then put u_i into a *Safe User Set U_S*. But if $\Pi(u_i) < \tau_i$, the user u_i cannot reach his target w_i after the current reset and will need a further reset before reaching w_i, we thus put him into the *Unsafe User Set U_N*.

Redirection Strategy For each user u_i, we have three essential ways to reset and redirect them, i.e. reset and redirect the user to walk for the maximum permitted time, walk to the position with the largest escapability or walk to the position with the largest safety.

If U_N is not empty, the "unsafe" users will need at least one more reset before reaching their targets, as they couldn't get to their targets in one fell swoop no matter what direction they are reset to. In this case, the user in U_N with the smallest maximum time to encounter obstacles is supposed to cause the next reset for other users, so we consider that user as the "bottleneck" user. We denote the subscript of the bottleneck user as $*$, so:

$$u_* = \arg\min_{u_i \in U_N} \Pi\left(u_i\right) \qquad (12)$$

We reset and redirect the bottleneck user u_* to walk for the exact maximum permitted time $\Pi(u_*)$ to maximize the time interval to the next common reset for all the online users. In order to do that, we reset u_* to the optimal direction of $\arg\max_{\theta \in \Theta} T_{max}((\boldsymbol{p}_*, \theta))$, and steer u_* with the curvature gain leading to the longest walking time $\Pi(u_*)$, where the gain is selected from the $(2k + 1)$ candidate curvature gains mentioned in Section III-A.

For other "safe" or "unsafe" users whose remaining time are greater than the bottleneck time, since they are expected to reset at time $\Pi(u_*)$, we need to reset and redirect them to more favorable physical positions in time $\Pi(u_*)$ so that they can travel further after the reset triggered by the bottleneck user u_*.

To achieve this goal, we estimate the reachable spatial extent after reset $\varphi(\boldsymbol{p_i}, \Pi(u_*))$ using the algorithm in Section III-B within the bottleneck time $\Pi(u_*)$ for each of the other users. For a specific user u_i, if there are *skeleton positions* in their reachable area $\varphi(\boldsymbol{p_i}, \Pi(u_*))$, we select the one of all the reachable *skeleton positions* with the largest escapability $L(\boldsymbol{p}_c)$ as their physical destination, denoting as $\boldsymbol{p}_i^D$, and redirect the user u_i to $\boldsymbol{p}_i^D$ using the reset direction and gains obtained using the method mentioned in Section III-B.

$$\boldsymbol{p}_i^D = \arg\max_{\boldsymbol{p}_c \in \chi(E_i) \cap \varphi(\boldsymbol{p_i}, \Pi(u_*))} L\left(\boldsymbol{p}_c\right) \qquad (13)$$

This ensures u_i to have a longer walking time after the scheduled reset to be triggered by the bottleneck user. In case there is no *skeleton position* in u_i's reachable area $\varphi(\boldsymbol{p_i}, \Pi(u_*))$, which is a rare case that only may happen when $\Pi(u_*)$ is too small or the *skeleton positions* are too sparse, we reset and redirect u_i to walk for the longest time $\Pi(u_i)$ instead, with the same operation as u_*.

However, for other "safe" users with the remaining time $\leq$ the bottleneck time, as they will reach their virtual targets before the common reset happens and turn around to the unknown new targets, it's unreasonable to also navigate them to the maximum escapability destinations with the bottleneck time. Such users cannot be ignored, as when U_N is empty, the bottleneck user doesn't exist and the bottleneck time is equivalent to $+\infty$, all users are "safe" users with the remaining time $\leq$ the bottleneck time. For each such user u_i, we just need to reset and redirect the user to a more reassuring position for the unknown next target during the user's walking time τ_i. In order to do that, we firstly calculate the reachable spatial area $\varphi(\boldsymbol{p_i}, \tau_i)$ with remaining time τ_i for the user u_i. Then we find the *skeleton positions* in $\varphi(\boldsymbol{p_i}, \tau_i)$ with the greatest safety $H(\boldsymbol{p}_c)$ as u_i's physical destination $\boldsymbol{p}_i^D$, thereby to determine the reset direction and redirection gains.

$$\boldsymbol{p}_i^D = \arg\max_{\boldsymbol{p}_c \in \chi(E_i) \cap \varphi(\boldsymbol{p_i}, \tau_i)} H\left(\boldsymbol{p}_c\right) \qquad (14)$$

The reason we optimize the position by checking the overall safety $H(\boldsymbol{p}_c)$ rather than the escapability $L(\boldsymbol{p}_c)$ is to cope with the user's unpredictable turning direction to the next random target. Once u_i has reached his current target w_i and has turned ready for moving to the next target, we calculate the longest time he can walk from his new pose, and redirect u_i accordingly until a new reset is triggered by some user. If a common reset is triggered just when the user is turning, we allow the user to continue to turn in place but not move until the common reset ends. If the user starts walking before the common reset is completed, the resetting interface will pop up to guide the user back to the starting position. When everyone is done, all users can start moving at the same time.

IV. EXPERIMENTS

A. Implementation Details

In our experiments, we use $k = 10$ as the number of different candidate radii for the time span and spatial extent estimation. Therefore, considering the steering direction and zero curvature
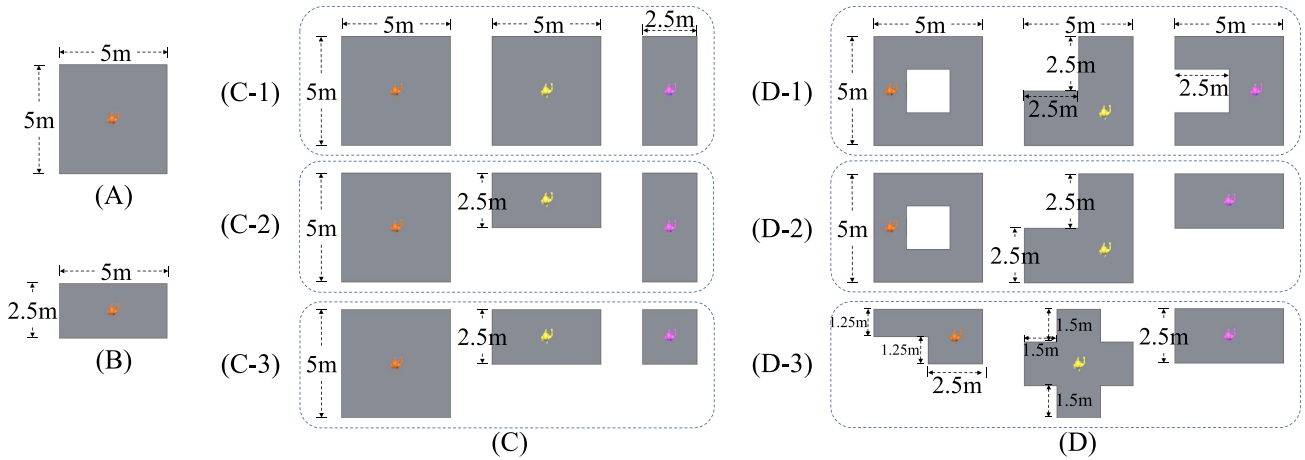
Fig. 5. The physical environment layouts used for experiments.

gain, there are a total of 21 potential curvature gains. We set the spacing between the *skeleton poses* as $\delta = 0.5\,m$, which is a suitable density for users' possible position since the adult's shoulder width is $34cm \sim 38\,cm$. The number of orientations of skeleton poses is set to $\lambda = 30$ for a dense scan of the potential directions. The radius of curvature gain threshold is set to $R = 7.5\,m$, and the translation gain thresholds are set to $g_t^{min} = 0.86$ and $g_t^{max} = 1.26$, respectively. The gain thresholds have been verified and commonly used by extensive RDW literature [5], [8], [9], [12], [13], [14], [15], [17], [20], [26], [29]. With the above parameters, our RDW controller runs in real-time on a PC with Intel Core i7-9750H CPU 2.60 GHz and 8 GB of RAM.

Simulation is widely recognized as an effective way to evaluate RDW methods [5], [9], [12], [14], [15], [18], [20], [26], [27]. We use the motion model proposed by Azmandian et al. [9] to simulate the users' movements in VE. As with most methods, we let the simulated user in VE translate at a constant speed of 1 m/s and rotate at a constant rate of $\pi/2$ rad/s. Each target in the VE is randomly generated at a random distance between $2\,m$ and $6\,m$ from the user and at a random angle within 180 degrees in front of the user. The user's starting position is random in PE. A simulation trial ends when all users have walked more than 400 meters, and we have 100 simulate trials for each method under each PE configuration.

We compare our method with several state-of-the-art RDW methods, including Steer-to-Center (S2C) [8], Steer-to-Orbit (S2O) [8], ZigZag [7], Dynamic APF (DAPF) [15] and the steer-by-reinforcement learning method (SRL) [18]. To make the comparison as fair as possible, when a common reset happens in the comparison methods, we do not make the users who were not encountering obstacles wait in vain, but reset them simultaneously to better orientations of their current locations. In addition, we use reset-to-gradient (R2G) [12] as the reset strategy for the methods (S2C, S2O and Zigzag) without a reset policy.

B. Performance With Different Numbers of Users

Intuitively, the more online users, the more frequently users trigger resets. Therefore, we firstly evaluate how the user

quantity affects the number of resets for our method and the comparison methods. To control for variables, we use identical PE for each user and test methods with varying numbers of users. Specifically, we first configured a square PE with a side length of $5\,m$ for each user as shown in Fig. 5 (A), and tested the methods with different user numbers of 2, 3, 4 and 6.

The distributions of the number of resets for all methods under different user quantities are shown in the box-plots in Fig. 6 (A). We can find from the distribution that the number of resets for our method is always significantly smaller than that for the comparison methods regardless of the online user quantity. We can also find that the number of resets for all methods increases as the online user quantity increases, but the increase in the reset times for our method is noticeably slower than that for the comparison methods.

Although we reset all the users to their optimal orientations to reduce the resets for the comparison methods, the comparison methods do not have the scheme to consider the collaboration between different users, which makes users often implicated in the reset of other users. On the contrary, our method reasons and maximizes the bottleneck time to the subsequent common reset for all the users and redirects the users to more favorable positions in their PEs to prepare for the next reset in advance, thus triggering the fewest common resets.

The numbers of resets of the comparison methods increase rapidly along with the user number. Since the comparison methods do not coordinate the redirection of all users, it is not surprising that the probability of being implicated in a reset triggered by other user will boost for each user as the user number increases. As our method maximizes the time interval to the next common reset for all the online users, our method is more advantageous when the number of users increases.

To further increase the challenge of the evaluation, we next upgraded the difficulty of the PE by switching it to an even smaller shape, i.e. a $2.5m \times 5\,m$ rectangle with one side only half the length of the original, as shown in Fig. 5 (B). We also tested the performance of all methods with different user numbers of 2, 3, 4 and 6, and the results are shown in Fig. 6 (B). As resets are more easily to be triggered in such small PEs, from
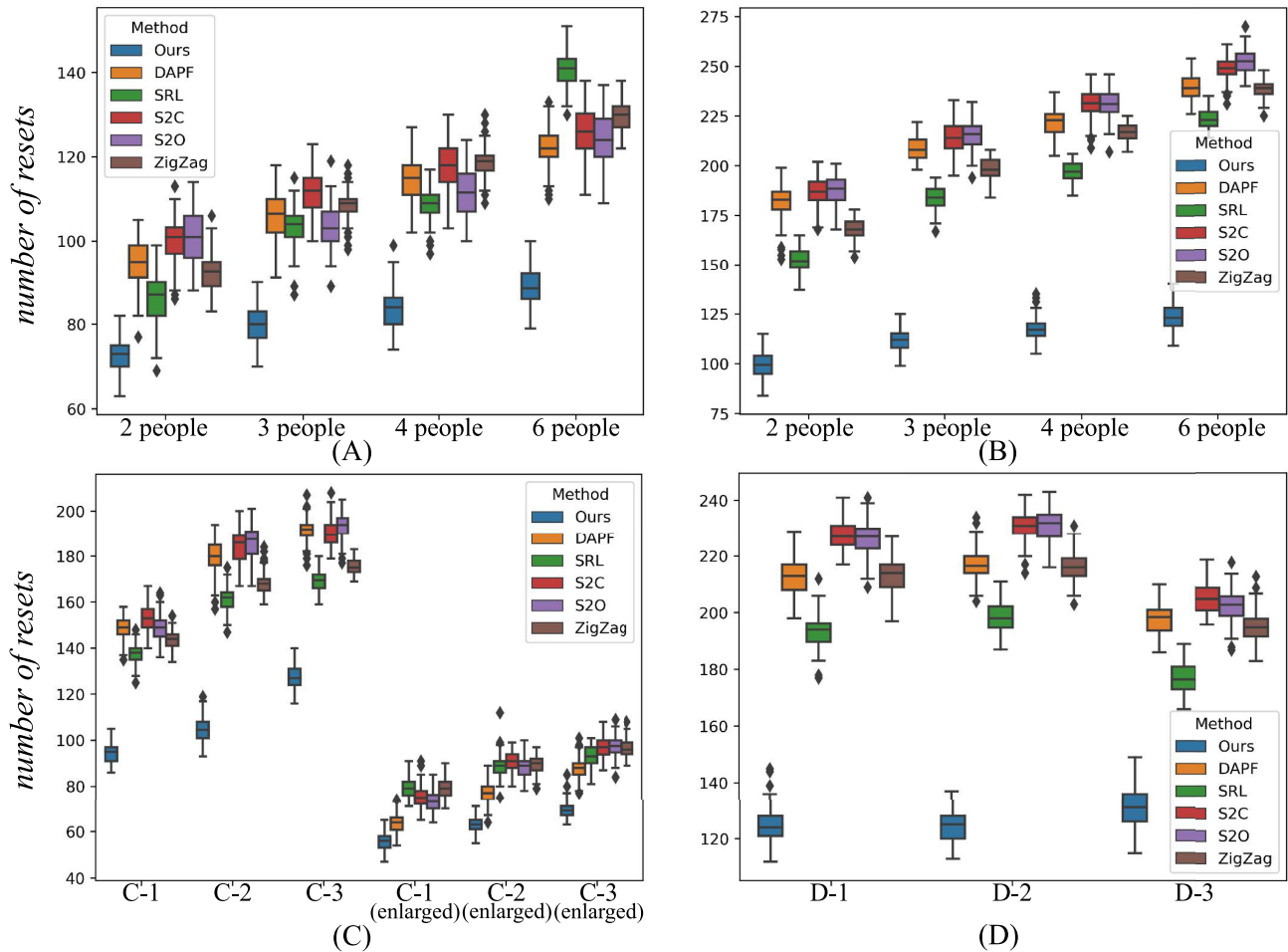
Fig. 6. The box-plots of the number of resets. (A) is the box-plot for the results of each method tested with different numbers of users located in off-site indentical $5\,m \times 5\,m$ square PEs. (B) is the box-plot for the results of each method tested with different numbers of users located in off-site indentical $2.5\,m \times 5\,m$ rectangle PEs. (C) is the box-plot for the results of each method tested with 3 users located in off-site unequal PEs. (D) is the box-plot for the results of each method tested with 3 users located in off-site unequal and irregular PEs.

the results we can see that the number of resets for all methods becomes larger in this kind of more challenging environment, but our method demonstrates more advantages over the other methods.

We conducted the Kruskal-Wallis H test on all the above results and found that the result distributions are statistically different. Post hoc pairwise comparisons by Mann–Whitney U test find satisfactory significance between our method and all the comparison methods ($p << 0.01$ after Bonferroni adjustment).

Despite the statistical differences between the comparative methods and our method, we want to clarify that the comparison methods were not designed for off-site multiplayer online VR scenarios, so the results do not mean that the comparison methods are poor. We also find from the results that S2C may not be a good idea for the off-site multiplayer online VR scenarios compared to S2O. If the user is too close to the PE center at the time when a common reset occurs, the user may not have a maximum walking distance after the reset, so the next reset will be triggered earlier. This makes it possible that S2C may be inferior to methods such as S2O in the off-site multiplayer online VR cases, although S2C is proved to perform better than S2O in most general cases [9].

C. Performance on Unequal and Irregular PEs

Since real-world online VR users are usually distributed in physical spaces of different shapes, we then test the performance of all the above methods under unequal PE layouts to better evaluate the method in real-world use cases. To control for variables, we fix the number of online users to 3 in these experiments, and use various combinations of unequal PE layouts to test the RDW controllers. Specifically, we first use the three combinations of layouts shown in Fig. 5 (C) to test the methods. In these three combinations of PE layouts, part of the PEs become smaller than others, and the combination gets harder and harder from (C-1) to (C-3). In addition to the layout shape, the PE sizes also affect the number of resets. To further evaluate the impact of PE size on methods' performance, we reuse the PE layout combinations in Fig. 5 (C) and enlarge each PE by doubling its original side length to make a larger PE with the same shape, and then we test these methods again in the larger PEs. So totally we have tested the methods under 6 sets of PE combinations.

The distributions of the results are reported in Fig. 6 (C) as box-plots. From the results we can find that our method has an absolute advantage over the comparison methods when the PE

layouts are unequal. That is because in unequal PEs the users with relatively small spaces will easily become "bottleneck" users and can markedly "drive" other users to reset. However, by analyzing the bottleneck time to the next reset and redirecting all the users accordingly, our method can synchronize the resets for more users and make a larger time interval between common resets. Although the number of resets get smaller in the enlarged unequal PEs for all the methods, the Kruskal-Wallis H test and the post hoc comparisons by Mann–Whitney U test still confirmed the statistical difference between our method and the comparison methods ($p << 0.01$ after Bonferroni adjustment).

Since the PE is usually irregular due to obstacles in real-world scenarios, we then challenge to test the methods with unequal and irregular PEs. We add various obstacles into the PE to make the PE irregular, and the irregular PE layout combinations used for testing are shown in Fig. 5 (D). The number of users is also fixed to 3. As we can see, the irregular PEs are smaller, usually non-convex and sometimes have voids inside due to obstacles.

The box plots of the number of resets for irregular PE combinations are shown in Fig. 6 (D). As complex obstacles limit the user's walking, the reset times for all methods are highly increased. However, we can find that our method still significantly improves over the comparison methods in these challenging irregular PEs. This illustrates that our mechanism for maximizing the time interval to the next common reset can work effectively in the irregular PEs to greatly reduce the number of resets.

V. USER STUDY

A. Experiment Design

In order to assess our method in real-world applications, we conducted a live user study. We implemented our algorithm with VIVE Pro Eye headsets in Unity, and conducted experiments with a *three-user* scenario. We assign the users to three separate physical spaces, each $5m * 3m$, similar to common household rooms. During the experiment, each participant with a headset was located in their own physical space, exploring a shared VE simultaneously. We mounted two tracking stations on two opposite sides for each space, ensuring that each user in their venue could be tracked. We use the Photon Unity Networking 2 to implement the communication among the three sets of equipment. Each participant starts at their site center, facing north. In our experimental online VR application, each user is required to walk toward a randomly generated target in the virtual space. Once the participant reaches their target, a new target will be generated. As a fair online application, when one of the participants has to be reset, the RDW controller makes all of the users reset at the same time. Considering that the user's physical position may change during the reset process, we included a position calibration function in the reset interface: by having the user align two arrows in the reset interface, we ensure the user return to the original physical position where they were prompted to reset. It is difficult for our test to record the number of resets at a specified physical walking distance since the three participants may have different walking distances. Alternatively, we recorded the distance each user has traveled
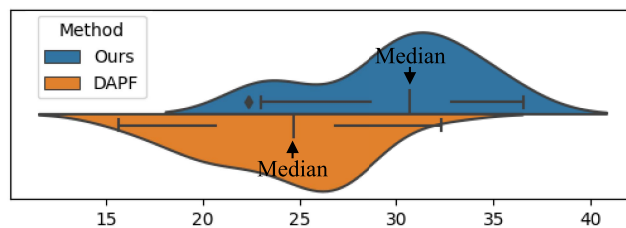

Fig. 7. The distributions of the participants' average walking distance (m).

in VE when a fixed number of 8 common resets have been triggered as the measurement to evaluate all the RDW methods. In the reported results below, *the longer the users walked, the better the RDW method is*. We use the state-of-the-art method DAPF [15] as the comparison method in the user study. As the participants' walking speed v is unknown in advance, we instructed the participants to walk calmly as in [3], and used the average speed v of the last segment of their straight walking as the estimated walking speed of the following segment of straight walking.

B. Results

We recruited 27 participants (10 females and 17 males, aged 19 to 28) to join our tests. Since one experiment requires three participants, we randomly assigned the participants to different groups and asked each of them to join three experiments. Thus we conducted a total of 27 multi-user experiments. In each experiment, the 3 participants were given adequate rest time between the two trials of the comparison methods. The results are shown in Fig. 7.

From Fig. 7, we can find that users can have a larger expected walking distance between resets using our method, which results in a better immersive experience. Mann-Whitney U test further finds the statistical difference between the comparison methods ($U = 109.0, P < .001$). We had also asked the participants to fill an SSQ [36] table to record their subjective feelings before and after the trials in terms of the quality of the experiences. The SSQ records find no significant difference between our method and DAPF, showing that the participants were not feeling noticeably uncomfortable with either method.

VI. LIMITATIONS AND FUTURE WORK

Our work has several limitations. We only use a fixed curvature gain during one segment of the user's walk before stopping in physical space. However, if we do not consider the sacrifice of the user comfort, we can steer the user with dynamically changing curvature gains within one segment of walk to make the user's physical path more flexible, thus the actual walking time to a given physical position will also be longer.

As the user's walking time is naturally related to the walking speed in VE, our calculations for both the time before encountering obstacles and the reachable area at a given time are related to the users' speeds. Although most existing methods assume the users to have constant walking speeds, this assumption may not hold if the user walks too fast in VE, as the speed can

fluctuate greatly. A more desirable speed estimation strategy would help our method cope with this situation. Likewise, our method currently requires knowledge of the user's next virtual target, so it is not general in situations where the user's next virtual target is difficult to know and estimate.

Although our method has outperformed the state-of-the-art methods in online VR scenarios, the performance of our method could be further improved by involving rotation gains. By adjusting rotation gains, we will be able to redirect users to a better orientation with a more refined control when they turn to walk towards new virtual targets, which could further increase the user's longest walking time and reduce the overallnumber of resets. The strategy considering rotation gains remains a future work of our method. In addition, we currently only conduct user study under one PE layout combination, and more PEs should be tested to further verify the performance of the method.

For fairness in online VR applications, our current *locomotion fairness constraint* requires all users to reset together, but this constraint may be too strong and may degrade the experience for users in larger rooms. How to relax this constraint while ensuring the fairness of online applications is also a future research direction.
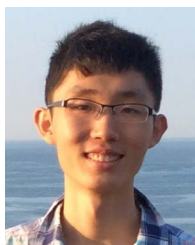
VII. Conclusion

In this work, we propose a redirected walking method for online multiplayer VR applications, which provides users with fair locomotion opportunities regardless of different physical environment layouts, and coordinates the redirection and reset for all the users in different physical spaces to reduce the total number of resets. We achieve our goal by first estimating all the users' walking time before encountering obstacles and reachable areas to find the bottleneck user leading to the next reset. Then, we redirect all users to their favorable poses before resetting. The simulation experiments and user study demonstrate the ability of our method to provide more continuous walking experiences for multiple online players in the same virtual environment.

References

[1] C. Ondrejka, "Escaping the gilded cage: User created content and building the metaverse," *NYL Sch. L. Rev.*, vol. 49, 2004, Art. no. 81.

[2] M. Usoh et al., "Walking> walking-in-place> flying, in virtual environments," in *Proc. 26th Annu. Conf. Comput. Graph. Interactive Techn.*, 1999, pp. 359–364.

[3] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected walking," 2001, doi: 10.2312/egs.20011036.

[4] H. Sakono, K. Matsumoto, T. Narumi, and H. Kuzuoka, "Redirected walking using continuous curvature manipulation," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 11, pp. 4278–4288, Nov. 2021.

[5] N. L. Williams, A. Bera, and D. Manocha, "ARC: Alignment-based redirection controller for redirected walking in complex environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 5, pp. 2535–2544, May 2021.

[6] Y.-J. Li, M. Wang, F. Steinicke, and Q. Zhao, "OpenRDW: A redirected walking library and benchmark with multi-user, learning-based functionalities and state-of-the-art algorithms," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2021, pp. 21–30.

[7] S. Razzaque, "Redirected walking," Ph.D. dissertation, University of North Carolina at Chapel Hill, USA, 2005. [Online]. Available: https://techreports.cs.unc.edu/papers/05--018.pdf

[8] E. Hodgson and E. Bachmann, "Comparing four approaches to generalized redirected walking: Simulation and live user data," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 4, pp. 634–643, Apr. 2013.

[9] M. Azmandian, T. Grechkin, M. T. Bolas, and E. A. Suma, "Physical space requirements for redirected walking: How size and shape affect performance," in *Proc. Int. Conf. Artif. Reality Telexistence Eurograph. Symp. Virtual Environ.*, 2015, pp. 93–100.

[10] L. Fan, H. Li, and M. Shi, "Redirected walking for exploring immersive virtual spaces with HMD: A comprehensive review and recent advances," *IEEE Trans. Vis. Comput. Graph.*, to be published, doi: 10.1109/TVCG.2022.3179269.

[11] Y.-J. Li, F. Steinicke, and M. Wang, "A comprehensive review of redirected walking techniques: Taxonomy, methods, and future directions," *J. Comput. Sci. Technol.*, vol. 37, no. 3, pp. 561–583, 2022.

[12] J. Thomas and E. S. Rosenberg, "A general reactive algorithm for redirected walking using artificial potential functions," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 56–62.

[13] E. R. Bachmann, E. Hodgson, C. Hoffbauer, and J. Messinger, "Multi-user redirected walking and resetting using artificial potential fields," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 5, pp. 2022–2031, May 2019.

[14] J. Messinger, E. Hodgson, and E. R. Bachmann, "Effects of tracking area shape and size on artificial potential field redirected walking," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 72–80.

[15] T. Dong, X. Chen, Y. Song, W. Ying, and J. Fan, "Dynamic artificial potential fields for multi-user redirected walking," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2020, pp. 146–154.

[16] D.-Y. Lee, Y.-H. Cho, and I.-K. Lee, "Real-time optimal planning for redirected walking using deep Q-learning," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 63–71.

[17] D.-Y. Lee, Y.-H. Cho, D.-H. Min, and I.-K. Lee, "Optimal planning for redirected walking based on reinforcement learning in multi-user environment with irregularly shaped physical space," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2020, pp. 155–163.

[18] R. R. Strauss, R. Ramanujan, A. Becker, and T. C. Peck, "A steering algorithm for redirected walking using reinforcement learning," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 5, pp. 1955–1963, May 2020.

[19] Y. Chang, K. Matsumoto, T. Narumi, T. Tanikawa, and M. Hirose, "Redirection controller using reinforcement learning," 2019, *arXiv:1909.09505*.

[20] N. L. Williams, A. Bera, and D. Manocha, "Redirected walking in static and dynamic scenes using visibility polygons," 2021, *arXiv:2106.06807*.

[21] J. Thomas, C. Hutton Pospick, and E. Suma Rosenberg, "Towards physically interactive virtual environments: Reactive alignment with redirected walking," in *Proc. 26th ACM Symp. Virtual Reality Softw. Technol.*, 2020, pp. 1–10.

[22] T. C. Peck, H. Fuchs, and M. C. Whitton, "Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments," in *Proc. IEEE Virtual Reality Conf.*, 2010, pp. 35–38.

[23] R. Cools and A. L. Simeone, "Investigating the effect of distractor interactivity for redirected walking in virtual reality," in *Proc. Symp. Spatial User Interact.*, 2019, pp. 1–5.

[24] N. Rewkowski, A. Rungta, M. Whitton, and M. Lin, "Evaluating the effectiveness of redirected walking with auditory distractors for navigation in virtual environments," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 395–404.

[25] S.-Z. Xu, T.-Q. Liu, J.-H. Liu, S. Zollmann, and S.-H. Zhang, "Making resets away from targets: POI aware redirected walking," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 11, pp. 3778–3787, Nov. 2022.

[26] E. R. Bachmann, J. Holm, M. A. Zmuda, and E. Hodgson, "Collision prediction and prevention in a simultaneous two-user immersive virtual environment," in *Proc. IEEE Conf. Virtual Reality*, 2013, pp. 89–90.

[27] M. Azmandian, T. Grechkin, and E. S. Rosenberg, "An evaluation of strategies for two-user redirected walking in shared physical spaces," in *Proc. IEEE Conf. Virtual Reality*, 2017, pp. 91–98.

[28] Z.-C. Dong, X.-M. Fu, Z. Yang, and L. Liu, "Redirected smooth mappings for multiuser real walking in virtual reality," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–17, 2019.

[29] T. Dong, Y. Shen, T. Gao, and J. Fan, "Dynamic density-based redirected walking towards multi-user virtual environments," in *Proc. IEEE Virtual Reality 3D User Interfaces*, 2021, pp. 626–634.

[30] B. Williams, G. Narasimham, T. P. McNamara, T. H. Carr, J. J. Rieser, and B. Bodenheimer, "Updating orientation in large virtual environments using scaled translational gain," in *Proc. 3rd Symp. Appl. Percep. Graph. Vis.*, 2006, pp. 21–28.

[31] V. Interrante, B. Ries, and L. Anderson, "Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments," in *Proc. IEEE Symp. 3D User Interfaces*, 2007.

[32] F. Steinicke, G. Bruder, L. Kohli, J. Jerald, and K. Hinrichs, "Taxonomy and implementation of redirection techniques for ubiquitous passive haptic feedback," in *Proc. IEEE Int. Conf. Cyberworlds*, 2008, pp. 217–223.

[33] E. Langbehn, P. Lubos, G. Bruder, and F. Steinicke, "Bending the curve: Sensitivity to bending of curved paths and application in room-scale VR," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 4, pp. 1389–1398, Apr. 2017.

[34] F. Steinicke, G. Bruder, K. Hinrichs, J. Jerald, H. Frenz, and M. Lappe, "Real walking through virtual environments by redirection techniques," *JVRB-J. Virtual Reality Broadcast.*, vol. 6, no. 2, 2009.

[35] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, "Estimation of detection thresholds for redirected walking techniques," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 1, pp. 17–27, Jan./Feb. 2010.

[36] R. S. Kennedy and J. E. Fowlkes, "Simulator sickness is polygenic and polysymptomatic: Implications for research," *Int. J. Aviation Psychol.*, vol. 2, no. 1, pp. 23–38, 1992.
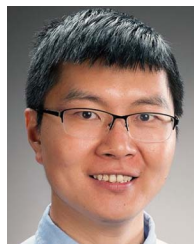
Miao Wang (Member, IEEE) received the bachelor's degree from Xidian University in 2011 and the PhD degree from Tsinghua University in 2016. He is an associate professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, China. His research interests include Virtual Reality, Computer Graphics and Visual Computing. During 2016-2018, he did postdoc research in Visual Computing with Tsinghua University. He serves a program committee member of IEEE VR and ISMAR conferences. He is a member of ACM and Asiagraphics.

Sen-Zhe Xu received the PhD degree of Computer Science and Technology from Tsinghua University, Beijing, in 2021. He is currently a postdoc in Yau Mathematical Sciences Center with Tsinghua University. His research interests include virtual reality and image/video processing.

Fang-Lue Zhang (Member, IEEE) received the bachelor's degree from Zhejiang University, Hangzhou, China, in 2009, and the Doctoral degree from Tsinghua University, Beijing, China, in 2015. He is currently a senior Lecturer with the Victoria University of Wellington, New Zealand. His research interests include image and video editing, mixed reality, and image-based graphics. He is a member of ACM. He received Victoria Early Career Research Excellence Award in 2019. He is on the editorial board of Computer & Graphics. He is a committee member of IEEE Central New Zealand sector.

Jia-Hong Liu is currently working toward the undergraduation degree with the Department of Computer Science and Technology, Tsinghua University. His research interests include virtual reality and 3D scene reconstruction.

Song-Hai Zhang (Member, IEEE) received the PhD degree in computer science and technology from Tsinghua University, Beijing, in 2007. He is currently an associate professor with the Department of Computer Science and Technology at Tsinghua University. His research interests include computer graphics, virtual reality and image/video processing.