


APF-S2T: Steering to Target Redirection Walking Based on Artificial Potential Fields

Jun-Jie Chen, Huan-Chang Hung, Yu-Ru Sun, and Jung-Hong Chuang 

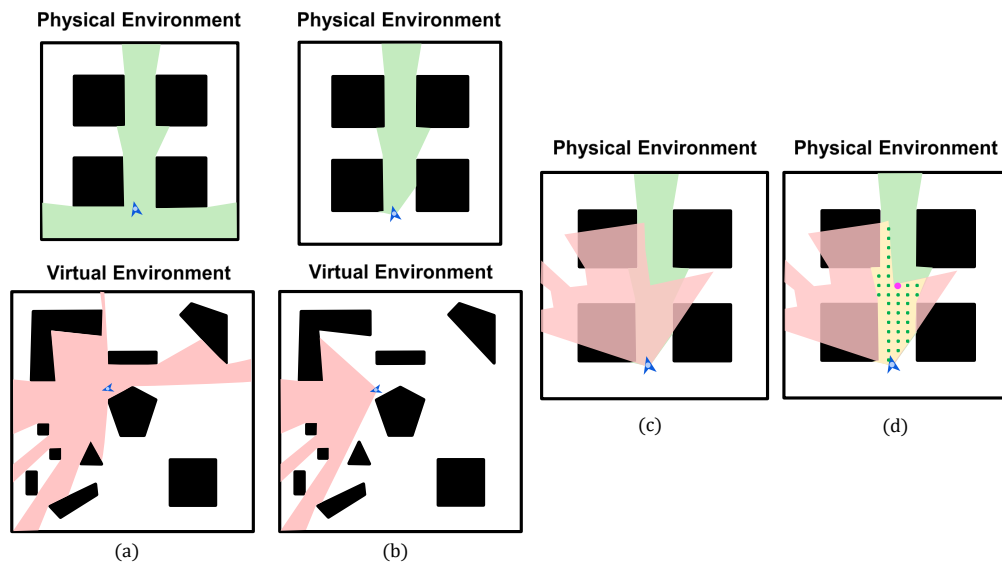


Fig. 1: Overview of APF-S2T: (a) visibility polygons for the user positions in the physical and virtual environments, (b) physical and virtual cells for the user poses in the physical and virtual environments, (c) embed the virtual cell in the physical environment such that the poses are coincident, (d) find the target sample point with the lowest score within the area where the physical and embedded virtual cells overlap, and determine the steering direction for setting RDW gains.

Abstract—Redirected walking (RDW) enables users to walk naturally within a virtual environment that is larger than the physical environment. Recently, several artificial potential field (APF) and alignment-based redirected controllers have been developed and have been demonstrated to significantly outperform conventional controllers. APF Steer-to-Gradient (APF-S2G) and APF Redirected Walking (APF-RDW) utilize the negative gradient and the total force vector, respectively, which are localized to the user's position. These vectors usually point towards the opposite wall when the user is in corridors, resulting in frequent resets within those regions. This paper introduces the APF Steer-to-Target (APF-S2T), a redirected controller that first finds the target sample point with the lowest score in the user's walkable area in both physical and virtual environments. The score of a sample point is determined by the APF value at the point and the distance from the user's position. The direction from the user's position to the target point is then used as the steering direction for setting RDW gains. We conducted a simulation-based evaluation to compare APF-S2T, APF-S2G, APF-RDW, Visibility Polygon-based alignment (Vis.-Poly.) and Alignment-Optimized controllers in terms of the number of resets and the average distance between resets. The results indicated that APF-S2T significantly outperformed the state-of-the-art controllers.

Index Terms—Redirected walking, artificial potential field, locomotion, virtual reality

1 INTRODUCTION

Redirected walking (RDW) enables natural walking in virtual environments (VEs) that are larger than the physical environments (PEs) available. Compared to other locomotion techniques, such as walking in place, steering and teleportation, natural walking has been shown to improve user presence sense [35], efficient navigation [24, 29], and spatial knowledge acquisition in VE [25, 42]. Redirected walking supports

natural walking by imperceptibly steering the user along a physical path that differs from their virtual path, with the amount of redirection controlled by *gains*. To ensure that redirection remains imperceptible to the user during steering, the gains applied must be within the perceptual thresholds [26]. The redirection controllers determine the gains to steer the user to avoid collisions with obstacles and issue a reset to reorient the user away from obstacles if an incoming collision is detected. When a reset is triggered, the user is required to pause and make the necessary adjustments to their position or orientation to avoid any potential obstacles. Consequently, the immersive experience is interrupted. Minimizing the frequency of resets or maximizing the average distance between resets is a significant concern when evaluating the effectiveness of the design of the redirection controller [6, 23, 33, 37, 38, 41].

Classic controllers, such as Steer-to-Center (S2C), Steer-to-Orbit (S2O) and Steer-to-Multiple-Targets (S2MT) [23], do not make use of the layout of PE and rely on the assumptions that the PE is convex and free of obstacles. Until recently, artificial potential fields (APFs) have

• Jun-Jie Chen, Huan-Chang Hung, Yu-Ru Sun, Jung-Hong Chuang are with National Yang Ming Chiao Tung University. E-mail: {jay8832791.cs09|hchung.cs10|yuru1219.cs11}@nycu.edu.tw, jhchuang@cs.nycu.edu.tw.

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

been applied to the design of RDW controllers. The concept of APF was initially developed in the field of robotics to address the issue of motion planning with collision avoidance. The aim of APF is to direct a robot towards a specific target while also ensuring that it does not collide with any obstacles in its path. The APF at a point combines a repulsive term that aims to move the robot away from obstacles and an attractive term that moves the robot toward the target. When APF is applied to redirection controllers, only the repulsive term is used. The APF-based approach takes into account the layout of the PE, allowing controllers to work with concave PEs with obstacles. APF-S2G steers the user according to the negative gradient of the APF at the user's location [33], while APF-RDW uses the total force vector of the APF at the point for the RDW steering [6, 19]. Later, controllers have been proposed that guide users by aligning their walkable areas in physical and virtual environments [37–39]. It has been observed that if the user's walkable areas in PE and VE are perfectly aligned, they will not experience a collision with PE while moving within the walkable area in VE. These are reactive and use information from both PE and VE to steer the user to the aligned walkable area in PE.

APF-S2G and APF-RDW have been shown to be more effective than previous methods and have been successful in managing multiple users within the same physical environment [19, 33]. However, the steering direction they used, which was a negative gradient or total force vector, was local information at the user's position. This often caused the steering direction to point toward the wall when the user was in a corridor, resulting in frequent resets in these areas. APF values can be used to determine which point is farther away from any obstacles or boundaries in the vicinity. This information is beneficial for redirected walking as it can help identify which point has the most open space around it. In this sense, the APF can be a valuable source of information for redirected walking. More research is anticipated to maximize the potential of APF for the development of RDW controllers.

Our objective in this study is to utilize the APF information that is not local to the user's position, aiming to steer the user toward the most open space in front of them. The proposed controller searches for a target point with the lowest score within the user's walkable area in PE that is limited to be accessible by the user in VE. The point's score is calculated based on the APF value at the point and the user's distance from the point. The APF has a repulsive effect, meaning that the point with the lowest APF value is farthest away from obstacles and boundaries, and thus the space around it is the most open. The target point we find is not farthest away, but is relatively farther away from obstacles and boundaries. Since the target point is within the user's walkable areas in both PE and VE, it is likely accessible to the user from their current positions. The found target point is considered the steering target, and the direction from the user's position to the target point is used as the steering direction for setting gains, with the goal of steering the user toward the most open space in front of them. Therefore, the proposed controller is called APF-S2T. Finally, we present a simulation-based evaluation demonstrating that the APF-S2T controller significantly outperforms the APF-S2G [33], APF-RDW [19], Vis.-Poly [38], and Alignment-Optimized [39] controllers. The main contributions of our work include (a) the introduction of a new APF-based controller, APF-S2T, and (b) a simulation-based comparative study to evaluate the performance of APF-S2T against the state-of-the-art controllers in terms of the number of resets and the average distance between resets.

2 RELATED WORK

2.1 Redirected Walking

When exploring a virtual environment (VE), walking is the most natural way. However, the virtual environment is often larger than the physical environment (PE). Redirecting Walking (RDW), originally proposed by Razzaque et al. [23], imperceptibly steers users to walk a physical path that is different from the virtual path to avoid physical obstacles or go beyond the tracking space. The amount of redirection is controlled by the *gains*, that is, the translation gain, the rotation gain, and the curvature gain [27]. The translation gain scales the velocity when walking. The rotation gain scales the user's rotation. The curvature gain steers

users to walk on a curved path in the PE when they walk in a straight line in the VE. To ensure that the user remains imperceptible during steering, the gains applied must fall within the perceptual threshold. According to Steinicke et al. [26], the translation gain thresholds ranged from 0.86 to 1.26, from 0.67 to 1.24 for the rotation gain, and when applying the curvature gain, the radius of the circular arc must be equal to or greater than 22 meters. However, the maximum curvature gain that is used most frequently in recent literature is usually set at a radius of 7.5 meters, as reported in [13].

In addition to steering algorithms, reset plays a crucial role in the performance of redirected walking techniques. When the user approaches obstacles closely, a reset is triggered that forces them to turn towards a safer direction. However, a reset can disrupt the user's experience and diminish immersion. As a result, it is advisable to minimize the frequency of reset unless necessary. Numerous reset strategies have been proposed in the past. For example, Williams et al. [36] introduced the 2:1-Turn and Freeze-Turn strategy, which involves doubling the user's rotation in the VE or freezing the VE display. This allows users to rotate in the PE without altering their VE orientation. Some reset strategies are custom-tailored to complement specific steering algorithms. For example, in conjunction with the APF-S2G algorithm, Thomas et al. [33] introduced three reset strategies: modified reset to center (MR2C), reset to gradient (R2G), and step-forward reset to gradient (SFR2G). In Williams et al.'s [37, 38] alignment-based controllers, the reset strategy involves identifying the direction within the PE that aligns most closely with the user's forward direction in the VE. Specifically, they measure the distance to obstacles in the user's heading direction in the VE. Then, measure distances to obstacles from various directions in the PE and select the direction that closely matches the heading distance in the VE as the reset direction.

2.2 Redirected Controllers

Generally, redirection controllers are classified as scripted, reactive, or predictive [18, 21]. Scripted controllers steer users to walk on a predefined real path and virtual path [4, 23]. Controllers employ environmental manipulation techniques which utilize concepts like change blindness or impossible overlapping spaces and may be classified as scripted [15, 29, 30]. Predictive controllers employ a prediction model to predict user movements in VE, allowing them to determine the appropriate steering actions. Examples of such controllers include FORCE [43], MPCRed [20]. In general, predictive controllers outperform reactive controllers when path prediction is relevant [20, 43]. However, reactive controllers tend to have a broader applicability in scenarios that require unrestricted movement and exploration of the environment.

Reactive controllers steer the user based only on the available information on the current state, without knowing the user's future position or path. Classic reactive controllers disregard the layout of the PE and operate under the assumption of a convex and obstacle-free PE. Steer-to-Center (S2C), Steer-to-Orbit (S2O) and Steer-to-Multiple-Target (S2MT) [23] steer the user toward the center of the tracking space, onto a circular path around the center of the tracking space and one of the predefined targets in the tracking space, respectively. Recently, controllers based on artificial potential field (APF) [6, 8, 19, 33] have taken into account the layout of the PE when guiding users. Consequently, these controllers can be applied in concave PEs with obstacles. APF adds a repulsive force to all obstacles and limits, pushing the user away. Bachmann et al. [6] proposed APF-RDW, which steers the user toward the vector sum of repulsive forces. APF-RDW can also be adapted for multi-user redirected walking by considering other users as obstacles, which also exert repulsive forces. In both simulation and live studies, APF-RDW consistently outperforms S2C in rectangular physical spaces. Messinger et al. [19] proposed APF-SC, a revised version of APF-RDW that works effectively in irregular rectangular physical spaces. In APF-SC, an obstacle is divided into segments and the force exerted by each segment is determined by its length. APF-SC calculates forces only from the segments facing the user position. Simulation studies have shown that APF-SC outperforms S2C in regular and irregular physical spaces. Dong et al. [8] proposed a dynamic

version of APF-RDW that incorporates a steering target that produces a gravitational pull on the user. Thomas et al. [33] proposed APF-S2G, a controller that steers the user according to the negative gradient of the APF function at the position of the user. Reports indicated that APF-S2G performs better than S2C in non-convex PE with and without obstacles.

Thomas et al. [32, 34] proposed the first form of alignment-based steering that aims to support passive haptic. When the user moves toward an interactive target in the VE, redirected walking is used to align the position of the interactive target in the VE with a proxy object in the PE. Consequently, the user can interact with both the target in the VE and the proxy object in the PE simultaneously, resulting in passive haptic. Subsequent research has explored the alignment of walkable areas in both VE and PE and developed RDW controllers [37–39]. If the walkable areas of the PE and the VE are perfectly aligned, users will not encounter any obstacles when walking in the walkable area in the VE. These controllers make use of both PE and VE data and operate as reactive controllers. The first controller of this kind was ARC [37]. ARC calculates the distance from the user to the nearest obstacle in the front, left, and right directions within both virtual and physical environments to assess the similarity between walkable areas in PE and VE. It then sets gain values to guide users towards the physical location that best corresponds to the virtual space. The second alignment-based controller is Vis.-Poly. [38] which begins by dividing the visibility polygons of PE and VE into slices, and then chooses the physical slice that best matches the virtual slice to steer the user. Simulation-based studies showed that Vis.-Poly. controller outperforms ARC, APF-S2G, and S2C in both static and dynamic scenes. The Alignment-Optimized controller, introduced by Wu et al. [39], defined an objective function that involves the variable θ . This variable represents a signed angle that indicates the deviation from the user's heading. The purpose of this objective function is to quantify the extent to which the overlapped area of the walkable areas in the PE and VE compares with the size of the virtual walkable area. The steering direction for setting the RDW gains is determined based on the optimal angle obtained by optimizing the objective function. The Alignment-Optimized controller offers an alignment measurement that is different from, but more precise than, the Vis.-Poly. controller. It has been shown to be more effective compared to Vis.-Poly., ARC, APF-S2G and APF-RDW controllers.

Some recent redirected walking controllers employed reinforcement learning techniques. For example, the Steer-to-Optimal Target (S2OT) [16] controller employed deep Q-learning to choose the steering target during redirection, with the objective of steering the user away from the spatial boundaries and toward the center. An alternative reinforcement learning-based redirection controller is Steering via Reinforcement Learning (SRL) [28]. SRL directly calculated the gain for redirected walking to effectively steer the user away from obstacles in their forward, left, and right movements. In addition to single-user RDW controllers, there have been developments in multi-user RDW controllers utilizing reinforcement learning techniques. Lee et al. [17] proposed the Multi-user Steer-to-Optimal Target (MS2OT), which was an extension built on S2OT. Furthermore, Jeon et al. [14] proposed Optimal Space Partitioning (OSP), a technique that divides the shared physical space in real-time to avoid user collisions and minimize the need for resets. Alternatively, Croucher et al. [7] proposed the Lo-CoMoTe framework to improve the use of recorded live user data in learning-based RDW algorithms. The framework categorizes previous VR locomotion experiments based on three themes: navigational decisions, technique implementation, and modalities. The goal is to provide a better understanding of the experimental context and enable easier comparisons between different experiments.

2.3 Evaluation of RDW Controllers

Simulation-based methods are frequently employed to assess the effectiveness of RDW controllers [2, 3, 6, 16, 19, 33, 37–39]. By using simulation, it is feasible to effectively and comprehensively assess the controller performance under different parameters, environmental conditions, and virtual trajectory configurations. Simulation-based evaluation involves the use of virtual paths that are generated using

patterns [2, 3, 33, 37–39] or recorded from real user paths [6, 16, 19]. This enables the evaluation of the effectiveness and performance of controllers in various environments and parameter settings using the same paths. This approach allows for easy comparisons between different controllers under identical conditions. Unlike simulation-based evaluations, live user studies usually occur in a smaller number of physical environments, offering a limited range of sizes and layouts. Moreover, the comparative study often includes a limited number of controllers [6, 10, 16, 28, 40].

Despite the advantages of simulation-based evaluation, such as facilitating quick iterations of controller prototyping and allowing extensive comparative testing of various controllers, there are still some concerns. In a recent study conducted by Hirt et al. [11], the main objective was to examine the chaotic characteristics of RDW. The findings revealed that slight variations in the user's initial position and orientation within the PE can result in significantly diverse walking trajectories for a given virtual path. This highlights the sensitivity of the RDW process to minor alterations that can ultimately impact steering outcomes. Furthermore, the research discovered distinctions between pattern-based paths and actual user paths in certain situations, such as navigating, searching, or exploring within the virtual environment. The findings highlight the sensitivity of RDW and the importance of real user studies. The effectiveness of RDW steering may be influenced by the exclusion of specific user behaviors during simulation, such as strafing, accelerating, and moving in directions other than the user's heading. Azmandian et al. conducted a comparative study of S2C and S2O with fixed paths using live user studies and simulation-based evaluations [5]. Their findings revealed that the controllers exhibit similar performance between the simulated and real user studies in terms of the number of resets. Furthermore, simulation-based evaluations maintained performance trends among controllers compared to user studies. As a result, simulation-based evaluation is believed to offer a conservative estimate of average performance compared to live user studies.

3 APF-S2T: APF CONTROLLER WITH STEERING-TO-TARGET

3.1 Computing the Steering Direction

Previous APF-based redirection controllers employ APF information at the user's location; for example, APF-RDW utilizes the total force vector, and APF-S2G uses the negative gradient of the APF function at the user's position as the steering direction. The proposed APF-S2T is designed to find a target point within the user's walkable area in both physical and virtual environments and consider that point as the steering target. The target point is preferred to be located close to the center of a relatively larger open space in front of the user in PE. To this end, a pre-processing step is performed in which PE undergoes uniform point sampling. For each sample, the APF value is calculated. The utilized APF only takes into account the repulsive forces associated with the obstacles and boundaries that can be seen by the sample point. Therefore, the lower the APF value, the farther away the point is from the obstacles or boundaries.

The search for the target sample is limited to candidate samples that are located within walkable areas where the user can freely walk in both the PE and the VE. That is, the candidate samples are inside the overlapped region of the user's walkable areas in both PE and VE. To compute the steering direction for APF-S2T, the target sample with the lowest score from the candidate sample set is identified, and the direction from the user's position to the target sample is then used as the steering direction for setting RDW gains. The score is defined as a function of the APF value and the distance from the user's position to the sample position. The following paragraphs provide detailed information on the pre-processing steps, the definition of walkable areas, and the methodology for selecting the target sample based on score.

Pre-Processing In the pre-processing stage, a uniform sampling of points in the physical environment (see Fig. 2) and the APF value for each sample are calculated. For APF-S2G, the APF is calculated based on all obstacles and boundaries, whereas for APF-RDW, only the obstacles and boundaries facing the point are taken into account. Additionally, in APF-RDW, obstacles and boundaries

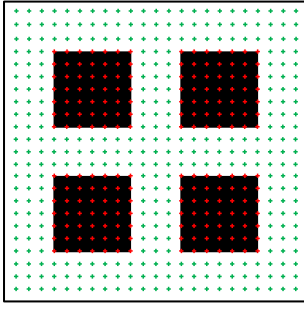


Fig. 2: The uniform sample points in the physical environment; green points in the open space and red points in the obstacles.

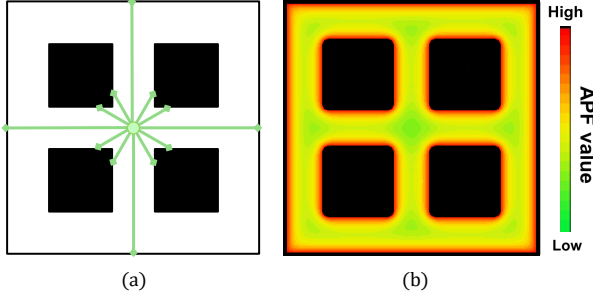


Fig. 3: (a) Ray samples for calculating the APF value. (b) Heatmap of the APF value.

are segmented. In APF-S2T, only obstacles and boundaries that are visible to the position of the sample point are considered when computing the APF, as we view the APF values as a means of showing the relative difference between the distances from the sample points to the obstacles and boundaries.

To calculate the APF value at a point p , n rays originating from p are uniformly sampled, and for each ray, the distance from p to the first intersection of the ray and the environment is computed; see Fig.3a. The APF value at p is defined as follows:

$$APF(p) = \sum_{i=1}^n \frac{1}{d(p, \theta_i)}, \quad (1)$$

where $d(p, \theta_i)$ is the distance between p and the first intersection of the ray i , represented by the angle $2\pi \left(\frac{i}{n}\right)$, and the environment. Fig.3b shows the heatmap of the APF value for an environment.

Visibility Polygon and Walkable Areas in PE and VE The concept of the visibility polygon in computational geometry [31] is employed to determine walkable areas for a user located in an environment E at a point P with heading \mathbf{h} , as seen in [38].

The visibility polygon for a user positioned at P in an environment E is the polygonal region of all points that can be seen from P . This region can be unbounded and is determined by the environments E and P . The visibility polygon of P , denoted $VP(P)$, is defined by a collection of n vertices, with each pair of adjacent vertices v_i and v_{i+1} forming an edge of the polygon. The calculation is performed based on the approach proposed by Suri et al. [31].

The walkable area of the user is an area ahead of the user in which the user could possibly navigate; hence, it should be bounded by the visibility polygon. The walkable area, denoted $C(P, \mathbf{h})$, for the user located in P with a heading of \mathbf{h} is a 2D polygonal region that is similar to a 2D view volume and is bounded by the visibility polygon $VP(P)$. Its origin is in the user position P , and its viewing direction is set to \mathbf{h} . This region has an included angle of 104° at P , which is comparable to the field of view (FOV) of modern head-mounted displays (HMDs) such as the Oculus Quest2 [1]. To calculate the walkable area, two

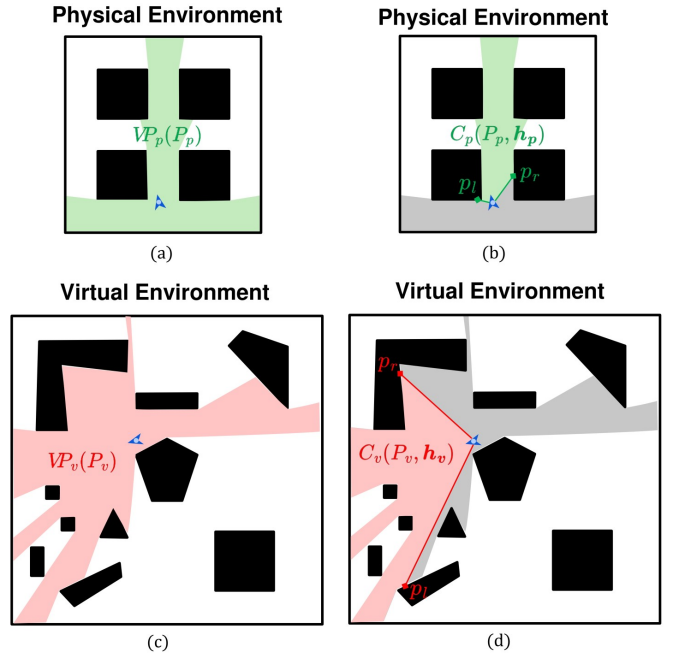


Fig. 4: (a) Visibility polygon in PE, (b) physical cell, (c) visibility polygon in VE, and (d) virtual cell.

rays are cast from P such that they are symmetric with respect to the user's heading and with the included angle of 104° . Their intersection points with the visibility polygon $VP(P)$ are denoted by p_l and p_r . Let S be the set of vertices of $VP(P)$ that fall within the area bounded by the two rays. The walkable area $C(P, \mathbf{h})$ is formed by points P , p_l , vertices in S , and p_r . The physical cell and the virtual cell are the terms used to denote the walkable areas for PE and VE, respectively.

Finding the Target Sample Based on Score Suppose that the user is located in the PE at P_p with a heading of \mathbf{h}_p and in the VE at P_v with a heading of \mathbf{h}_v . In the first step of APF-S2T, the visibility polygons $VP_p(P_p)$ in the PE and $VP_v(P_v)$ in the VE are computed (refer to Fig. 4a). Additionally, their corresponding physical cell $C_p(P_p, \mathbf{h}_p)$ and virtual cell $C_v(P_v, \mathbf{h}_v)$ are determined (see Fig. 4b).

The next step of APF-S2T is to locate the set of candidate samples from which we can pinpoint a target sample to serve as a steering target. Candidate samples to be considered must be located within the walkable area of the user in both PE and VE to ensure that it is accessible to the user. To do this, the virtual cell $C_v(P_v, \mathbf{h}_v)$ is embedded in the PE in such a way that the vertex of P_v and the vector \mathbf{h}_v align with P_p and \mathbf{h}_p , respectively, as shown in Fig. 5a. The embedded virtual cell is denoted as $C_v^{embedded}(P_v, \mathbf{h}_v)$. Therefore, candidate samples must be located within the overlap region of the physical cell $C_p(P_p, \mathbf{h}_p)$ and the embedded virtual cell $C_v^{embedded}(P_v, \mathbf{h}_v)$, that is, the candidate samples are in $C_p(P_p, \mathbf{h}_p) \cap C_v^{embedded}(P_v, \mathbf{h}_v)$ (see Fig. 5b).

In addition, to avoid resets that occur when the user moves from their current position toward the target sample point, the candidate samples are additionally limited. This is done by ensuring that the line segment from the user's position to the sample point is at least a distance l away from the boundary of the intersection region between $C_p(P_p, \mathbf{h}_p)$ and $C_v^{embedded}(P_v, \mathbf{h}_v)$. Here, l represents the distance used to determine if the user will collide with an obstacle during reset detection. Let $CandidateSampleSet$ be the collection of samples that meet the criteria mentioned above. Note that the inclusion of $C_v^{embedded}(P_v, \mathbf{h}_v)$ implies that the APF-S2T controller takes into account not only the PE data but also some VE information, such as walkable areas.

Finally, a target sample in $CandidateSampleSet$ is selected by identifying the one with the lowest score. The APF value alone can be

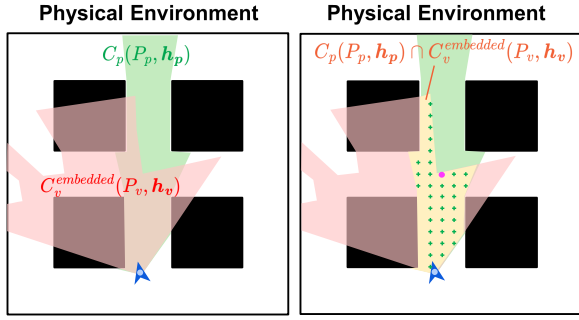


Fig. 5: (a) Embedded virtual cell and physical cell in PE. (b) overlapped area (yellow) of physical cell and embedded virtual cell, candidate sample set (green points), and the target sample point (purple point).

used as the score; however, this could lead to the user being stuck in the vicinity of a local minimum of APF when they are close to it. To avoid this situation, the distance between the sample point and the user position is taken into account, and the score of the sample point p_i is defined as follows:

$$score(p_i) = \frac{1}{1 + c(p_i, P_p)} APF(p_i), \quad (2)$$

where $c(p_i, P_p)$ is the distance between the sample point p_i and the user position P_p , and $APF(p_i)$ is the APF value at the p_i . The $score(p_i)$ reflects how far the position p_i is from the user and obstacles. A lower score signifies that the position is farther away from the user and the obstacles. Consequently, we select the sample point with the lowest score in *CandidateSampleSet* as the target sample, with the aim of choosing a target point that has the most open space surrounding it and also possibly provides a longer steering distance. The direction from the user's position to the target sample will serve as the steering direction for APF-S2T.

3.2 RDW Gains Setting

The RDW controller will derive the curvature gain, the rotation gain, and the translation gain and use them to steer the user's movement. Let θ^* be the signed angle from the user's heading to the steering direction. In APF-S2T controllers, the derivation of the curvature gain, the rotation gain, and the translation gain is based on the steering direction or its corresponding signed angle θ^* . Next, we describe how to set the gain values for APF-S2T.

Curvature Gain When the user is moving, the curvature gain is used to steer the user toward the target point. When locating the target sample point, the magnitude of the signed angle θ^* may exceed the maximum rotation angle of the curvature gain. Therefore, when setting the curvature gain θ^* needs to be checked if it exceeds the maximum rotation angle. If it does, the maximum curvature gain of $\frac{1}{7.5}$ is applied. If it does not exceed this threshold, the curvature gain is set to $\frac{\theta^*}{d}$, where d is the user's movement distance in a time step. This means that when the user walks for a distance of d , this curvature gain is applied to generate a rotation of θ^* to steer the user towards the target. The curvature gain is set as follows:

$$g_c = \begin{cases} \frac{\theta^*}{d} & \text{if } |\theta^*| \leq \text{maximum curvature gain} * d, \\ \frac{\text{sign}(\theta^*)}{7.5} & \text{if } |\theta^*| > \text{maximum curvature gain} * d, \end{cases} \quad (3)$$

where $\text{sign}(\theta)$ function returns 1 when θ is positive, -1 when θ is negative. We use this function to decide whether to turn left or right. d is the displacement distance of the user in a time step. That is, d is equal to the walking speed multiplied by the time step ($d = 1.0(m/s) * 0.05(s) = 0.05(m)$). The value *maximum*

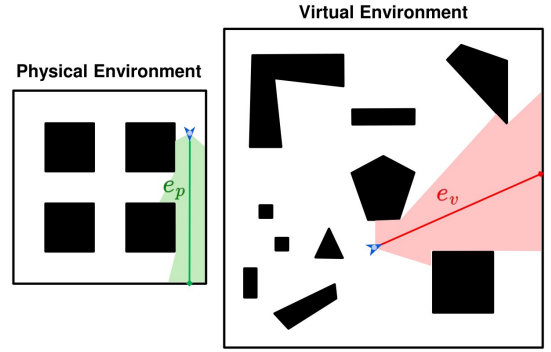


Fig. 6: Distances between the user and any obstacles or boundaries that lie ahead.

*curvature gain * d* represents the maximum rotation angle that can be generated by the curvature gain corresponding to the moving distance d .

Rotation Gain As the user turns in place, our aim is to steer them towards the steering direction as much as possible. Let r be the signed angle of the user's rotation at P_p . If the signs of θ^* and r are the same, the minimum rotation gain of 0.67 [26] is used. This serves to decrease the user's rotation within the VE, making them execute more rotations in the PE to reach their desired direction in the VE. As a result, this reduces the angular difference between the user's heading and steering direction. Otherwise, the maximum rotation gain of 1.24 [26] is used to mitigate the increase in the angular difference. Specifically, the rotation gain g_r is set as follows:

$$g_r = \begin{cases} 0.67 & \text{if } \theta^* r > 0, \\ 1.24 & \text{if } \theta^* r < 0. \end{cases} \quad (4)$$

Translation Gain Similar to setting the curvature gain, when determining the translation gain, the signed angle θ^* is checked if it exceeds the maximum rotation angle of the curvature gain. If not, the maximum translation gain of 1.26 is applied. This allows us to accelerate the user's velocity in VE, enabling them to reach the target point more quickly and increasing the chances of changing direction and turning toward the steering direction in PE. If text is positive, the current distances between the user and the obstacles ahead in both VE and PE are considered. As shown in Fig. 6, a ray originating from the user's position P_p with the user's heading direction h_p in PE is cast and its intersection with the visibility polygon $VP_p(P_p)$ is computed. Let e_p be the distance from P_p to the intersection point. A ray originating from P_v with the user's heading direction h_v in VE is also cast and its intersection with the visibility polygon $VP_v(P_v)$ is computed. Let e_v be the distance from P_v to the intersection point. The translation gain g_t is set as follows:

$$g_t = \begin{cases} 1.26 & \text{if } |\theta^*| > \text{maximum curvature gain} * d, \\ \text{clamp}(\frac{e_v}{e_p}, 0.86, 1.26) & \text{if } |\theta^*| \leq \text{maximum curvature gain} * d. \end{cases} \quad (5)$$

Similar to Eq. 3, d represents the user's displacement distance in a time step, and *maximum curvature gain * d* denotes the corresponding rotation angle.

3.3 Reset Strategies

It is necessary to implement a resetting strategy to detect when the user is about to collide with an obstacle in the PE and reorient them when it occurs. The trigger condition for the reset proposed by Williams et al. [37] is used. The specific process is as follows: Let $\theta_{heading}$ represent the user's heading direction in the PE after applying the gain, two tests are performed. The first is to check if the next position along $\theta_{heading}$ is within an obstacle or outside the boundary. If it is, the reset

is activated; if not, the second test is performed. In the second test, if the next step is close to an obstacle or the boundary, and the angle between the user's heading and the normal of the obstacle is more than 90 degrees, the reset will be triggered.

When a reset is activated at the user position P_p , a modified version of the APF-S2T technique is used to determine the direction of reorientation. It is performed as follows:

1. To derive the reorientation direction for resetting using the modified APF-S2T, the heading of the user positioned at P_p is set to the normal of the obstacle surface \mathbf{n}_p . With this user pose, the physical cell with the included angle of 170° is derived. The included angle is much larger than the one used in RDW steering as we attempt to explore a larger walkable area to locate the target sample for reorientation.
2. The target sample point with the lowest score in PE is identified by using the techniques described in Section 3.1. The virtual cell $C_v(P_v, \mathbf{h}_v)$ is then embedded such that P_v and \mathbf{h}_v match P_p and \mathbf{n}_p in the physical space and the target sample point is checked if it is within the embedded virtual cell.
3. If the target sample point found is located within the embedded virtual cell, it is used as the target point for steering direction. If not, the sample point with the next lowest score is considered, and Steps 2 and 3 are repeated until the desired target sample point is found.

Once the target point has been found, the reorientation process is similar to that of face-center [36] and 2:1-turn [22], except that the user will be facing the direction that points to the target sample point.

4 EVALUATION STUDY DESIGN

In designing the experiments for evaluation, we aim to assess whether the proposed APF-S2T would have better performance in terms of the number of resets and the average distance between resets, compared to the state-of-the-art controllers. We designed a series of four environment pairs and conducted a comparative study between APF-S2T, APF-S2G [33], APF-RDW [19], Vis.-Poly. [38], and Alignment-Optimized [39] controllers for each environment pair. APF-S2G [33] and APF-RDW [19] were well-known APF-based controllers. As the current implementation of APF-S2T was designed for a single user RDW controller, we did not include other APF-based methods that were designed to deal with multiple users within the same physical environment [8, 9] as conditions. Vis.-Poly. [38] and Alignment-Optimized controllers were included, as they are representative alignment-based controllers. The five controllers share the same criteria for reset detection, but each has its own method of reorientation. Since simulation-based evaluation can be effective in collecting large experimental data and understanding controller performance, strength, and weakness, we conducted a simulation-based study. For each experiment, we simulated the use of controllers for randomly created virtual paths. Number of resets and the average reset distance served as performance measures in the experiments. The following hypotheses were proposed for the results of the experiments:

H1 APF-S2T will perform better than APF-based controllers such as APF-S2G and APF-RDW in terms of the number of resets and the average distance between resets.

H2 APF-S2T will perform better than alignment-based controllers such as Visibility-Polygon (Vis.-Poly.) and Alignment-Optimized controller in terms of the number of resets and the average distance between resets.

4.1 Environment Pairs

The layout of an environment pair has an influence on the performance of RDW controllers [2, 19], so we included various environment layouts in the experiments to gain a thorough understanding of how controllers behave in different environments. We conducted the experiments using four pairs of environments with distinct features. The layouts of the four

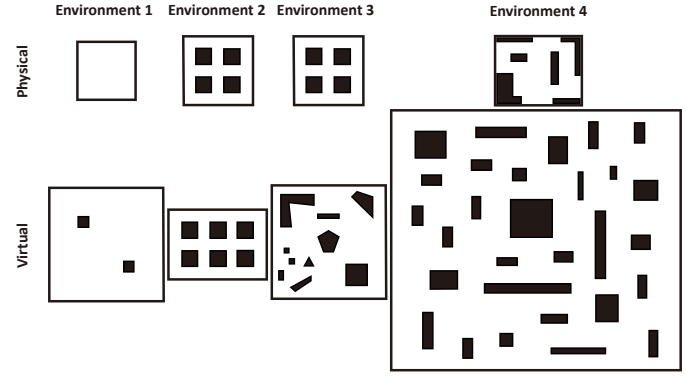


Fig. 7: Layouts of four environment pairs used in the experiments.

pairs of environments are shown in Figure 7. Environment 1 consisted of a $10m$ by $10m$ PE with no obstacles and a $20m$ by $20m$ VE with two $2m$ by $2m$ obstacles. The obstacle-free square PE provided an ideal setting for RDW, making it a common choice to test the performance of the RDW controller [4, 33, 40, 41]. Environment 2 was composed of a $12m$ by $12m$ PE and a $17m$ by $12m$ VE. PE and VE were made up of narrow corridors and had a high level of local similarity. Environment 3 had the same PE as Environment 2, but with a different $20m$ by $20m$ VE. The VE in Environment 3 contained irregular and concave obstacles, leading to a low level of local similarity to the PE. Note that Environment 2 and Environment 3 have previously been used to evaluate RDW controllers by Williams et al. [38] for evaluation. Environment 4 consisted of a PE measuring $15m$ by $12m$ and a VE of $51m$ by $46m$, which is much larger than PE. The PE consisted of narrow and open areas, making it narrower than Environment 1 but more open than Environments 2 and 3. Note that PE in Environment 4 resembles a living room, which is a common physical space for home entertainment.

4.2 Simulation Design and Setting

For each condition in the experiment with an environment pair, we ran simulations of the controller being used on 100 randomly generated paths in the virtual environment. The simulated user would start from a random pose and walk along the virtual path in both the physical and virtual environments. The path model is grounded in the waypoint generation model proposed by Azmandian et al. [2] and the user walking trajectory model introduced by Fan et al. [10]. Initially, the waypoint generation model creates waypoints using a *distance sampling distribution* ranging from $2m$ to $6m$ and an angle sampling distribution ranging from $-\pi$ to π radians. These generated waypoints represent the paths taken by the user during the exploration of the VE [2]. In contrast to connecting waypoints with straight lines, the walking trajectory model [10] is applied to construct paths between waypoints. This model involves the random insertion of three types of segments: straight lines, left turns and right turns, each turning segment with a turning radius of $3m$, until the cumulative length of the segments inserted is sufficient to connect two waypoints. This introduces perturbations that aim to simulate the behavior of user walking. Regarding the *waypoint count*, 100 waypoints are generated in the first three environments, while Environment 4 involves generating 200 waypoints. This decision is motivated by the larger virtual environment in Environment 4, allowing the walking range to cover the VE more comprehensively. Note that since the same 100 virtual paths were used throughout an experiment, the factor of the RDW controller was a factor within the subject in each experiment.

The simulation shared several parameter settings as Williams et al. [38]: the time step of the simulations was $0.05s$, the simulated user was represented as a circle with a radius of $0.5m$, and the simulated user walked along the paths with a speed of $1m/s$ and an angular velocity of $90^\circ/s$. In our APF-S2T implementation, the spacing between the sample points is set to $0.1m$. APF-S2G was implemented according

to Thomas et al. [33] with the reset strategy of SFR2G, which took 10 steps with a step size of 0.05m along the negative gradient of the APF to locate a target for reorientation. APF-RDW was implemented based on Messinger et al. [19], and Vis.-Poly. was implemented as in Williams et al. [38]. The thresholds for the RDW gains were the same in all experiments: the thresholds for the translation and rotation gains were those determined by Steinicke et al. [26], and the thresholds for the radius of curvature gain of 7.5m were a commonly adopted threshold in the literature [2, 12, 19, 33, 37, 38]. All experiments shared the same method to trigger a reset, as described in subsection 3.3.

5 RESULTS

5.1 Number of Resets

We conducted the Shapiro-Wilk's test to examine the normality of the data and also the Mauchly's test to assess the sphericity. The results of the tests showed that some of the data violated the assumptions of normality and sphericity, so nonparametric tests were used for the further analyses. The Friedman test was used to evaluate the effect of the factor of RDW controllers, and the Wilcoxon signed-rank test was applied for the post-hoc pairwise comparisons. The tests were carried out using a significance level of 0.05, which was adjusted by the Bonferroni correction in the post-hoc tests. The box plots along with the results of the post-hoc tests are shown in Figure 8.

5.1.1 Environment 1

The Friedman test revealed a significant difference in the number of resets between the RDW controllers, $\chi^2(4) = 244.264$, $p < 0.001$. The box plot for the number of resets and the results of the post-hoc Wilcoxon signed-rank tests are shown in Figure 8a, where asterisks indicate high significance with a p -value of less than 0.001. The mean number of resets was 20.38 for APF-S2T ($MED = 20.5$, $SD = 3.82$), 18.62 for Alignment-Optimized ($MED = 18$, $SD = 4.35$), 28.91 for Vis.-Poly. ($MED = 28$, $SD = 5.84$), 28.81 for APF-RDW ($MED = 29$, $SD = 3.78$), and 20.61 for APF-S2G ($MED = 21$, $SD = 3.40$).

5.1.2 Environment 2

The Friedman test showed a significant difference in the number of resets between the RDW controllers, $\chi^2(4) = 389.762$, $p < 0.001$. The box plot for the number of resets and the results of the post-hoc Wilcoxon signed-rank tests are depicted in Figure 8b, where asterisks indicate high significance with a p -value of less than 0.001. The mean number of resets was 64.08 for APF-S2T ($MED = 64$, $SD = 7.04$), 90.54 for Alignment-Optimized ($MED = 89.5$, $SD = 11.59$), 108.88 for Vis.-Poly. ($MED = 108$, $SD = 10.65$), 246.25 for APF-RDW ($MED = 242$, $SD = 26.95$), and 316.97 for APF-S2G ($MED = 322$, $SD = 30.16$).

5.1.3 Environment 3

The Friedman test revealed a significant difference in the number of resets between the RDW controllers, $\chi^2(4) = 383.367$, $p < 0.001$. The box plot for the number of resets and the results of the post-hoc Wilcoxon signed-rank tests are shown in Figure 8c, where asterisks indicate high significance with a p -value of less than 0.001. The mean number of resets was 74.53 for APF-S2T ($MED = 74$, $SD = 9.43$), 108.13 for Alignment-Optimized ($MED = 108$, $SD = 9.94$), 119.48 for Vis.-Poly. ($MED = 120$, $SD = 9.16$), 256.25 for APF-RDW ($MED = 254.5$, $SD = 25.46$), and 313.73 for APF-S2G ($MED = 312$, $SD = 23.93$).

5.1.4 Environment 4

The Friedman test showed a significant difference in the number of resets in the RDW controllers, $\chi^2(4) = 311.448$, $p < 0.001$. The box plot for the number of resets and the results of the post-hoc Wilcoxon signed-rank tests are depicted in Figure 8d, where asterisks indicate high significance with a p -value of less than 0.001. The mean number of resets was 77.12 for APF-S2T ($MED = 76$, $SD = 9.35$), 94.49 for Alignment-Optimized ($MED = 93.5$, $SD = 11.88$), 109.47 for Vis.-Poly. ($MED = 109$, $SD = 10.77$), 152.03 for APF-RDW ($MED = 150$, $SD = 32.89$), and 153.04 for APF-S2G ($MED = 143$, $SD = 48.49$).

5.2 Average Virtual Distance Between Resets

Similar to the analysis of the number of resets, we first used the Friedman test to evaluate the effect of the RDW controllers and then conducted post-hoc pairwise comparisons with the Wilcoxon signed-rank test. The significance level was set at 0.05, and the Bonferroni correction was applied during the post-hoc comparisons. Box plots along with the results of the post hoc tests are shown in Figure 9.

5.2.1 Environment 1

The Friedman test reveals a significant difference in the average virtual distance between resets between controllers, $\chi^2(4) = 244.264$, $p < 0.001$. The box plot for the average virtual distance between resets and the results of the post-hoc Wilcoxon signed-rank tests are depicted in Figure 9a, where asterisks indicate high significance with $p < 0.001$. 18.40 for APF-S2T ($MED = 17.72$, $SD = 3.68$), 20.42 for Alignment-Optimized ($MED = 19.66$, $SD = 5.07$), 13.20 for Vis.-Poly ($MED = 12.86$, $SD = 2.64$), 12.95 for APF-RDW ($MED = 12.84$, $SD = 1.64$) and 18.04 for APF-S2G. ($MED = 17.20$, $SD = 2.98$).

5.2.2 Environment 2

The Friedman test reveals a significant difference in the average virtual distance between resets between controllers, $\chi^2(4) = 389.762$, $p < 0.001$. The box plot for the average virtual distance between resets and the results of the post-hoc Wilcoxon signed-rank tests are depicted in Figure 9b, where asterisks indicate high significance with $p < 0.001$. 5.28 for APF-S2T ($MED = 5.21$, $SD = 0.61$), 3.77 for Alignment-Optimized ($MED = 3.74$, $SD = 0.47$), 3.12 for Vis.-Poly ($MED = 3.11$, $SD = 0.29$), 1.39 for APF-RDW ($MED = 1.40$, $SD = 0.13$) and 1.08 for APF-S2G. ($MED = 1.05$, $SD = 0.10$).

5.2.3 Environment 3

The Friedman test reveals a significant difference in the average virtual distance between resets between controllers, $\chi^2(4) = 383.367$, $p < 0.001$. The box plot for the average virtual distance between resets and the results of the post-hoc Wilcoxon signed-rank tests are depicted in Figure 9c, where asterisks indicate high significance with $p < 0.001$. The average virtual distance between resets was 4.87 for APF-S2T ($MED = 4.87$, $SD = 0.54$), 3.36 for Alignment-Optimized ($MED = 3.35$, $SD = 0.30$), 3.03 for Vis.-Poly ($MED = 3.00$, $SD = 0.23$), 1.42 for APF-RDW ($MED = 1.42$, $SD = 0.13$) and 1.16 for APF-S2G. ($MED = 1.16$, $SD = 0.08$).

5.2.4 Environment 4

The Friedman test reveals a significant difference in the average virtual distance between resets between controllers, $\chi^2(4) = 311.448$, $p < 0.001$. The box plot for the average virtual distance between resets and the results of the post-hoc Wilcoxon signed-rank tests are depicted in Figure 9d, where asterisks indicate high significance with $p < 0.001$. The average virtual distance between resets was 9.74 for APF-S2T ($MED = 9.77$, $SD = 1.11$), 7.98 for Alignment-Optimized ($MED = 7.91$, $SD = 0.95$), 6.86 for Vis.-Poly ($MED = 6.85$, $SD = 0.64$), 5.14 for APF-RDW ($MED = 5.03$, $SD = 1.09$) and 5.32 for APF-S2G. ($MED = 5.25$, $SD = 1.55$).

5.3 Hypotheses Verification

The statistical results for Environments 2, 3, and 4 showed that APF-S2T had a significantly lower number of resets and a significantly longer average reset distance than APF-RDW and APF-S2G. In Environment 1, APF-S2T performed significantly better than APF-RDW, and there were no significant differences between it and APF-S2G. Our first hypothesis H1 was corroborated by these results. In the Environments 2, 3, and 4, APF-S2T was found to be significantly better than Alignment-Optimized and Vis.-Poly. controller in terms of the number of resets and the average distance between resets. In Environment 1, APF-S2T performed significantly better than Vis.-Poly. and there was no significant difference between it and Alignment-Optimized. This result also confirms our second hypothesis H2.

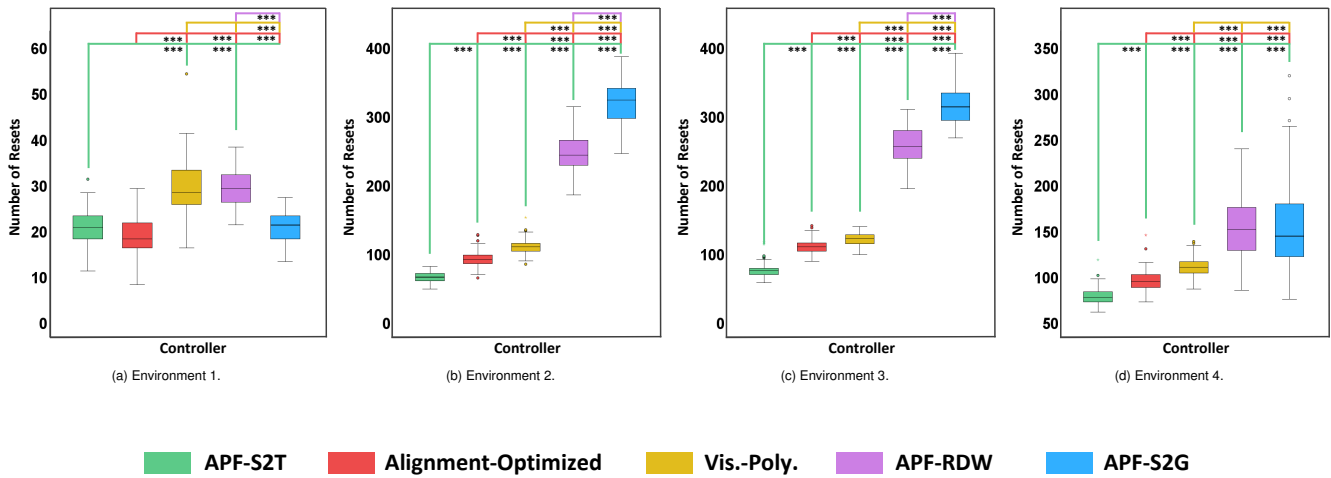


Fig. 8: Boxplots of the number of resets for each controller in the experiments. The colored lines with asterisks (***) indicate highly significant post-hoc pairwise differences with $p < 0.001$.

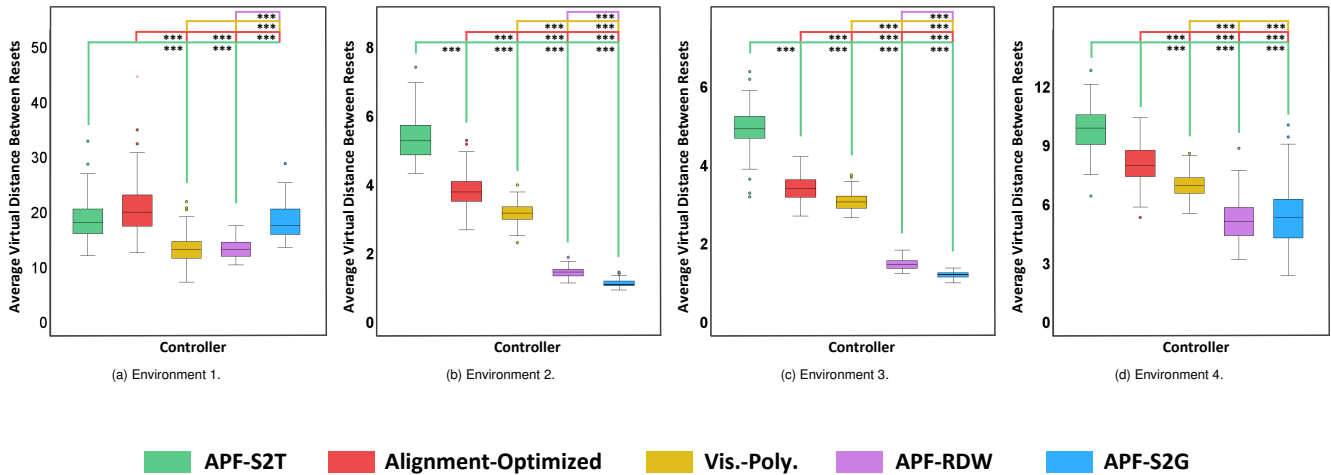


Fig. 9: Boxplots of the average virtual distance between resets for each controller in the experiments. The colored lines with asterisks (***) indicate highly significant post-hoc pairwise differences with $p < 0.001$.

6 DISCUSSION

The experimental results demonstrate that APF S2T performs better than APF-RDW and APF-S2G in terms of the number of resets and the average distance between resets in Environments 2, 3, and 4. In Environment 1, it performs better than APF-RDW and is comparable to APF-S2G. Differences in performance between various pairs of environments are mainly due to the narrowness of the PE. Now we will discuss each of these scenarios in detail. In Environments 2 and 3, the PE consists of narrow corridors. When using techniques like APF-RDW and APF-S2G, which steer the user based on the negative gradient of the APF or the total force vector at the user’s location, the presence of obstacles on both sides of the corridor can significantly impact the user’s movement. This frequently leads the user to be redirected in the opposite direction of the walls, occasionally resulting in them becoming trapped between two walls. On the other hand, APF-S2T selects a target sample point that has the lowest scores within the overlapping area of physical and virtual cells. It has a tendency to choose a target sample point that is close to the local minima of the APF, such as corners or intersections of corridors, as the steering target. This helps to minimize the impact of nearby obstacles. Even if the overlapping region of physical and virtual cells does not contain any local minima, APF-S2T tends to steer the user toward the central axis of

the corridor, where the APF is relatively low, making the user stay away from obstacles. Compared to Environments 2 and 3, Environments 1 and 4 have more open physical spaces. Environment 1 is completely empty, with minimal APF values at the center and increasing APF value with distance from it. This leads APF-based controllers to steer users toward the center, keeping them away from the boundaries. APF-based approach is generally effective in open environments. Environment 4 has narrow and open areas. APF-S2T allows users to benefit from the APF-based controller while walking in open areas, and swiftly pass through narrow areas, avoiding interference from nearby obstacles.

APF-S2T is found to perform better than alignment-based controllers, such as Alignment-Optimized and Vis.-Poly. controllers. We observed that the steering direction derived from alignment-based controllers usually does not point to the most open space in the PE, even when the alignment level is high. For Vis.-Poly. Controller, we noticed that alignment based on the decomposition of the visibility polygons in both PE and VE can be challenging due to two issues. Variations in the layout of the environment can have an impact on the decomposition of the visibility polygon. For example, the addition of a small obstacle close to a boundary wall can lead to the segmentation of an area into smaller slices. However, since the obstacle may be located far from the user, it may not necessarily affect the RDW steering. In Environment 1, there is a PE that is an open room and a VE that contains two small

obstacles. If the user moves towards the obstacles in VE, it may result in choosing a slice with the smallest area in PE as the steering target. This can affect the effectiveness of the steering and potentially cause a reset. Additionally, in Vis.-Poly., the alignment is based on the shape similarity, which is determined by the difference in area size. However, areas with the same area size may be very different in shape, making the alignment measure inaccurate.

Among the four pairs of environment, APF-S2G and APF-RDW were found to perform the best in Environment 1, followed by Environment 4, and the worst in Environment 3 and Environment 4. Therefore, APF-S2G and APF-RDW are more suitable for PE with a larger open space. Vis.-Poly. controller outperformed APF-S2G and APF-RDW in Environments 2, 3, and 4, but was comparable to APF-RDW and worse than APF-S2G in Environment 1. Alignment-Optimized performed better than Vis.-Poly., APF-S2G, and APF-RDW in all four environments, which is consistent with the findings reported in [39]. APF-S2T demonstrated better performance compared to the other four controllers in Environments 2, 3, and 4, while being comparable to Alignment-Optimized and APF-S2G in Environment 1. Note that the PE in Environment 4 resembles a living room, which is a common physical space for home entertainment.

Limitations The APF-S2T has certain limitations. After a target sample is identified and used as the steering target, the target sample point may not always be reached, especially when the user frequently turns on the VE. In this case, APF-S2T may become less effective. APF-S2T identifies the steering target by considering both physical and virtual cells determined by the user's headings. Therefore, APF-S2T is appropriate for situations where the user moves in directions within their field of vision. However, if the user frequently looks forward but moves backward or takes lateral steps, the identified steering target may no longer be optimal, and the performance of APF-S2T may be compromised.

To assess the capabilities and features of the controllers in different situations, simulation-based evaluation allows for an effective evaluation. However, the RDW controllers will be used in actual circumstances, and certain user behaviors, such as stumbling, walking acceleration, and head oscillation, are excluded from the simulation-based evaluation. Moreover, walking paths may vary greatly depending on the user's task in VR. Therefore, conducting real-time user studies is crucial to comprehending its functionality and identifying potential problems when applied in real-world VR applications. APF-S2T, like other controllers based on APF or alignment, depends on the layouts of PE and VE. However, this information may not always be available. If the environment does not provide these data, then real-time or pre-processed scene reconstruction is necessary, which is still a difficult task.

7 CONCLUSION AND FUTURE WORK

This paper introduced a new redirected controller based on APF, APF Steer-to-Target (APF-S2T). It differs from the existing APF-based controllers in that it locates a target sample point with the lowest score within the user's walkable areas in both physical and virtual environments; hence, the target point is likely to have the most open space surrounding it and be accessible to the user from their current positions. The score is calculated based on the APF value and the distance to the user. Thus, a lower score indicates that the sample point is relatively farther away from obstacles and boundaries. The simulation-based evaluation showed that APF-S2T outperforms the state-of-the-art controllers, such as APF-S2G, APF-RDW, Vis.-Poly. and Alignment-Optimized controllers.

This research suggests that when designing reactive controllers, it is beneficial to utilize the pertinent information ahead of the user in addition to the current user poses. APF-S2T locates a target sample point with the lowest score within the user's walkable areas in both physical and virtual environments and steers the user towards the most likely open space ahead of the user. Another example of this direction is the alignment-based approach, which determines the steering direction by aligning the workable areas in front of the user in both physical

and virtual environments. We anticipate that there are still numerous avenues to explore along this line of research.

As indicated in Section 6, the target sample point found in APF-S2T may not always be reached, thus reducing the effectiveness of the approach. Additionally, when steering the user toward the target sample point, collisions with obstacles or boundaries may occur. To address these two issues, it might be beneficial to combine APF-S2T with alignment-based approaches. APF-S2T was originally designed for single-user scenarios; however, it can be adapted to accommodate multiple users in the same physical space.

ACKNOWLEDGMENTS

This work was partly supported by the National Science and Technology Council, R.O.C., under Grant No. NSTC 112-2221-E-A49-111.

REFERENCES

- [1] HMD geometry database. <https://risa2000.github.io/hmdgdb/>, 2024. 4
- [2] M. Azmandian, T. Grechkin, M. T. Bolas, and E. A. Suma. Physical space requirements for redirected walking: How size and shape affect performance. In *ICAT-EGVE*, pp. 93–100, 2015. 3, 6, 7
- [3] M. Azmandian, T. Grechkin, and E. S. Rosenberg. An evaluation of strategies for two-user redirected walking in shared physical spaces. In *2017 IEEE Virtual Reality (VR)*, pp. 91–98, 2017. doi: 10.1109/VR.2017.7892235 3
- [4] M. Azmandian, R. Yahata, T. Grechkin, and E. S. Rosenberg. Adaptive redirection: A context-aware redirected walking meta-strategy. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2277–2287, 2022. 2, 6
- [5] M. Azmandian, R. Yahata, T. Grechkin, J. Thomas, and E. S. Rosenberg. Validating simulation-based evaluation of redirected walking systems. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2288 – 2298, 2022. 3
- [6] E. R. Bachmann, E. Hodgson, C. Hoffbauer, and J. Messinger. Multi-user redirected walking and resetting using artificial potential fields. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2022–2031, 2019. 1, 2, 3
- [7] C. Croucher, W. Powell, B. Stevens, M. Miller-Dicks, V. Powell, T. J. Wiltshire, and P. Spronck. Locomote – a framework for classification of natural locomotion in vr by task, technique and modality. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–19, 2023. doi: 10.1109/TVCG.2023.3313439 3
- [8] T. Dong, X. Chen, Y. Song, W. Ying, and J. Fan. Dynamic artificial potential fields for multi-user redirected walking. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 146–154. IEEE, 2020. 2, 6
- [9] T. Dong, Y. Shen, T. Gao, and J. Fan. Dynamic density-based redirected walking towards multi-user virtual environments. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 626–634. IEEE, 2021. 6
- [10] C.-W. Fan, S.-Z. Xu, P. Yu, F.-L. Zhang, and S.-H. Zhang. Redirected walking based on historical user walking data. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 53–62, 2023. doi: 10.1109/VR55154.2023.00021 3, 6
- [11] C. Hirt, Y. Kompis, C. Holz, and A. Kunz. The chaotic behavior of redirection - revisiting simulations in redirected walking. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2022. 3
- [12] E. Hodgson and E. Bachmann. Comparing four approaches to generalized redirected walking: Simulation and live user data. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):634–643, Apr. 2013. 7
- [13] E. Hodgson, E. Bachmann, and D. Waller. Redirected walking to explore virtual environments: Assessing the potential for spatial interference. *ACM Trans. Appl. Percept.*, 8(4), dec 2008. doi: 10.1145/2043603.2043604 2
- [14] S.-B. Jeon, S.-U. Kwon, J.-Y. Hwang, Y.-H. Cho, H. Kim, J. Park, and I.-K. Lee. Dynamic optimal space partitioning for redirected walking in multi-user environment. *ACM Trans. Graph.*, 41(4), jul 2022. doi: 10.1145/3528223.3530113 3
- [15] S.-U. Kwon, S.-B. Jeon, J.-Y. Hwang, Y.-H. Cho, J. Park, and I.-K. Lee. Infinite virtual space exploration using space tiling and perceivable reset at fixed positions. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 758–767, 2022. doi: 10.1109/ISMAR55827.2022.00094 2

- [16] D.-Y. Lee, Y.-H. Cho, and I.-K. Lee. Real-time optimal planning for redirected walking using deep q-learning. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 63–71. IEEE, 2019. 3
- [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 *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 155–163, 2020. doi: 10.1109/VR46266.2020.00034 3
- [18] Y.-J. Li, F. Steinicke, and M. Wang. A comprehensive review of redirected walking techniques: Taxonomy, methods, and future directions. *Journal of Computer Science and Technology*, 37(3):561–583, 2022. 2
- [19] J. Messinger, E. Hodgson, and E. R. Bachmann. Effects of tracking area shape and size on artificial potential field redirected walking. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 72–80. IEEE, 2019. 2, 3, 6, 7
- [20] T. Nescher, Y.-Y. Huang, and A. Kunz. Planning redirection techniques for optimal free walking experience using model predictive control. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 111–118. IEEE, 2014. 2
- [21] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, F. Steinicke, and E. S. Rosenberg. 15 years of research on redirected walking in immersive virtual environments. *IEEE Computer Graphics and Applications*, 38(2):44–56, 2018. 2
- [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 *2010 IEEE Virtual Reality Conference (VR)*, pp. 35–38. IEEE, 2010. 6
- [23] S. Razaque, Z. Kohn, and M. C. Whitton. Redirected walking. In *Proceedings of Eurographics*, 2001. 1, 2
- [24] R. A. Ruddle and S. Lessels. The benefits of using a walking interface to navigate virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(1):1–18, 2009. 1
- [25] R. A. Ruddle, E. Volkova, and H. H. Bülhoff. Walking improves your cognitive map in environments that are large-scale and large in extent. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 18(2):1–20, 2011. 1
- [26] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):17–27, 2009. 1, 2, 5, 7
- [27] F. Steinicke, G. Bruder, L. Kohli, J. Jerald, and K. Hinrichs. Taxonomy and implementation of redirection techniques for ubiquitous passive haptic feedback. In *2008 International Conference on Cyberworlds*, pp. 217–223, 2008. doi: 10.1109/CW.2008.53 2
- [28] R. R. Strauss, R. Ramanujan, A. Becker, and T. C. Peck. A steering algorithm for redirected walking using reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1955–1963, 2020. 3
- [29] E. A. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte. Leveraging change blindness for redirection in virtual environments. In *2011 IEEE Virtual Reality Conference*, pp. 159–166. IEEE, 2011. 1, 2
- [30] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum, and M. Bolas. Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):555–564, 2012. doi: 10.1109/TVCG.2012.47 2
- [31] S. Suri and J. O’Rourke. Worst-case optimal algorithms for constructing visibility polygons with holes. In *Proceedings of the second annual symposium on Computational geometry*, pp. 14–23, 1986. 4
- [32] J. Thomas, C. Hutton Pospick, and E. Suma Rosenberg. Towards physically interactive virtual environments: Reactive alignment with redirected walking. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology, VRST ’20*. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3385956.3418966 3
- [33] J. Thomas and E. S. Rosenberg. A general reactive algorithm for redirected walking using artificial potential functions. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR)*, pp. 56–62. IEEE, 2019. 1, 2, 3, 6, 7
- [34] J. Thomas and E. S. Rosenberg. Reactive alignment of virtual and physical environments using redirected walking. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 317–323, 2020. doi: 10.1109/VRW50115.2020.00071 3
- [35] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr. Walking > walking-in-place > flying, in virtual environments. In *Proceedings of the 26th Annual Conference on Computer graphics and Interactive Techniques*, pp. 359–364, 1999. 1
- [36] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. Rieser, and B. Bodenheimer. Exploring large virtual environments with an HMD when physical space is limited. In *Proceedings of the 4th symposium on Applied perception in graphics and visualization*, pp. 41–48, 2007. 2, 6
- [37] N. L. Williams, A. Bera, and D. Manocha. ARC: Alignment-based redirection controller for redirected walking in complex environments. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2535–2544, 2021. 1, 2, 3, 5, 7
- [38] N. L. Williams, A. Bera, and D. Manocha. Redirected walking in static and dynamic scenes using visibility polygons. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4267–4277, 2021. 1, 2, 3, 4, 6, 7
- [39] X.-L. Wu, H.-C. Hung, S. V. Babu, and J.-H. Chuang. Novel design and evaluation of redirection controllers using optimized alignment and artificial potential field. *IEEE Transactions on Visualization and Computer Graphics*, 29(11):4556–4566, 2023. doi: 10.1109/TVCG.2023.3320208 2, 3, 6, 9
- [40] 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 Transactions on Visualization and Computer Graphics*, 28(11):3778–3787, 2022. 3, 6
- [41] S.-Z. Xu, T. Lv, G. He, C.-H. Chen, F.-L. Zhang, and S.-H. Zhang. Optimal pose guided redirected walking with pose score precomputation. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 655–663, 2022. doi: 10.1109/VR51125.2022.00086 1, 6
- [42] C. A. Zanbaka, B. C. Lok, S. V. Babu, A. C. Ulinski, and L. F. Hodges. Comparison of path visualizations and cognitive measures relative to travel technique in a virtual environment. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):694–705, 2005. 1
- [43] M. A. Zmuda, J. L. Wonser, E. R. Bachmann, and E. Hodgson. Optimizing constrained-environment redirected walking instructions using search techniques. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1872–1884, 2013. 2