Multi-User Redirected Walking and Resetting Using Artificial Potential Fields

Eric R. Bachmann, Eric Hodgson, Cole Hoffbauer, and Justin Messinger

Fig. 1. Two immersed users are redirected around each other as they explore a virtual environment.

Abstract—Head-mounted displays (HMDs) and large area position tracking systems can enable users to navigate virtual worlds through natural walking. Redirected walking (RDW) imperceptibly steers immersed users away from physical world obstacles allowing them to explore unbounded virtual worlds while walking in limited physical space. In cases of imminent collisions, resetting techniques can reorient them into open space. This work introduces categorically new RDW and resetting algorithms based on the use of artificial potential fields that "push" users away from obstacles and other users. Data from human subject experiments indicate that these methods reduce potential single-user resets by 66% and increase the average distance between resets by 86% compared to previous techniques. A live multi-user study demonstrates the viability of the algorithm with up to 3 concurrent users, and simulation results indicate that the algorithm scales efficiently up to at least 8 users and is effective with larger groups.

Index Terms—Virtual environment, redirected walking, resetting, artificial potential field, collision avoidance

1 Introduction

Immersive virtual reality (VR) systems that combine head mounted displays (HMDs) and a large area position tracking system (e.g., [32]) enable users to move through virtual environments (VEs) as naturally as the physical world. This can be more intuitive than navigation via joysticks or specialized hardware such as treadmills. However, the size of the virtual world in such systems can be no larger than the physical tracking area if collisions with tracking area boundaries are to be avoided.

In order to simulate worlds of unlimited size, the correspondence between physical and virtual motion can be manipulated to facilitate navigation on a larger scale. Redirected Walking (RDW) algorithms have been designed to systematically guide users away from physical tracking area boundaries and into open space [22], [21]. Essentially, these techniques continuously rotate the virtual viewpoint to induce users to unknowingly walk in circles, resulting in reuse of the same physical space to explore new virtual areas. There is a trade-off, however, between how strongly a user is redirected and how perceptible (and potentially distracting) the steering is. Since ideally, RDW rotations are imperceptible to users steering must be limited to known perceptual limits [26], [24], [25]. This limits the minimum radius a user can be induced to follow without noticing, and thus how much physical space is required. Even so, it should be expected that users will periodically approach tracking area boundaries or other obstacles. To prevent physical collisions in these cases, resetting techniques are used to discretely reorient the user back into open tracking space [19], [20], [33], [34].

RDW algorithms [4], [21], [35] have been designed and evaluated [6], [7] with respect to how well they meet the sometimes-conflicting goals of minimizing (a) potential collisions and resets, (b) the size of the required tracking area, and (c) perceptual distortions. Though notable progress has been made, there are still a number of drawbacks to existing approaches. Few RDW or resetting algorithms make full use of a given tracking area when generating steering instructions (but see [35]). Instead, steering is functionally identical in a round room, square room, rectangular room, or L-shaped room. Existing algorithms also lack support for multiple users who are concurrently in the same tracking area. Some limited approaches have have been proposed for two users (e.g., [2], [1]), as discussed below, but these do not scale well to larger groups.

This work presents the development and experimental evaluation of categorically new approaches to redirected walking and resetting based on the use of artificial force vectors, or Artificial Potential Fields (APF) [9]. The approaches will be referred to as Artificial Potential Field Redirected Walking (APF-RDW) and Artificial Potential Field

- *Eric Bachmann, Cole Hoffbauer, and Justin Messinger are with the Department of Computer Science and Software Engineering, Miami University, Oxford, OH, 45056. E-mail: Eric.Bachmann@miamiOh.edu*
- *Eric Hodgson is with the Armstrong Institute for Interactive Media Studies, Miami University, Oxford, OH, 45056. E-mail: Eric.Hodgson@miamiOh.edu*

Manuscript received 10 Sept. 2018; accepted 7 Feb. 2019.
Date of publication 17 Feb. 2019; date of current version 27 Mar. 2019.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2019.2898764

Resetting (APF-R) for brevity. Instead of *attracting* users towards a goal location or idealized orbit like most generalized RDW algorithms [17], obstacles *repel* users and push them into open space. Each wall, obstacle, or other user generates a force vector that is directed away from that obstacle and has a length inversely proportional to its distance from the user. APF-RDW determines the steering direction by summing the individual force vectors, which points toward open space. APF-R works similarly to re-orient the user towards the safest available area during resets. The approach is computationally simple, with each wall and additional user adding a single force vector to be summed. Because users constantly repel each other, it is unnecessary to predict collisions or make explicit decisions about how to prevent them. This approach also maximizes use of the full tracking area as users can disperse more fully into the space. Moreover, there is no requirement to modify to the virtual world or task being simulated.

To preview the experimental results below, APF-RDW significantly outperformed Steer-to-Center (STC) for single users, increasing the average distance between resets by 86% and decreasing the number of resets by 64%. In live trials involving multiple concurrent users, APF-RDW was able to redirect up to 3 users away from each other without measurably increasing average steering rates, and with only a modest increase in the total number of resets. Further simulation results for up to 8 concurrent users indicate that the algorithm scales efficiently and effectively to large numbers of users. Specific contributions include:

- Presentation of the first full implementation of live multi-user RDW, to the authors' knowledge.
- Development, testing, and evaluation of a new generalized RDW algorithm based on APF.
- Development and demonstration of a new generalized resetting algorithm based on APF.
- Experimental results indicating APF-RDW outperforms previous methods for single users.
- Experimental demonstration that APF-RDW and APF-R support arbitrary numbers of users.

The remainder of this document is organized as follows: Section 2 provides a brief background of related work. For a more comprehensive overview of redirected walking and resetting see [17]. Section 3 provides a detailed description of the APF-RDW and APF-R algorithms. Section 4 presents the experiments and resulting data. The document is concluded by a discussion of the implications and limitations of the work.

2 Background

2.1 Redirected Walking

Traditionally, RDW algorithms have attempted to steer users towards a goal [17]. Razzaque's *Steer-to-Center*, *Steer-to-Orbit*, and *Steer-to-Multiple Targets* algorithms, for example, respectively attempt to redirect users towards the middle of the tracking space, onto an orbit around that center, or to a series of predetermined way points [21]. Razzaque proposed different magnitudes of rotation depending on how the user is moving. Steering rates depend on whether users are standing still, rotating, or walking. This approach can react to users' spontaneous actions in a generalized fashion, but has also been used to steer users along pre-scripted paths containing enough turns to keep users contained in small areas [3] [22].

Steinicke et al. [26], [27], [24], [25] performed multiple experiments to discover imperceptible threshold rates for redirection while walking and turning. These estimates have proven important, but likely conservative, given their participants' sole focus on detecting the manipulations. Other work in which users are engrossed in a primary task have demonstrated low rates of detection with higher rates [8], [6]. Neth et al. also found that curvature gains were highly dependent on walking speed [16] and suggested that these results may explain the range of walking radii found in previous studies. In [30], Suma et al. created a taxonomy of a broad range of redirection techniques and demonstrated that subtle reorientation techniques yielded the fewest breaks in presence.

Field et al. were the first to compare different RDW techniques to see which performed best [4]. They developed simulations to compare Steer-to-Center with two other methods of their design: Large Circle and Small Circle. These circle techniques used information about what direction users were going to turn and when they were going to turn based on their actions in the VE. Such path-prediction techniques have been the subject of further work, which use information about the task and VE to predict where users might go and thus where they can best be redirected [35]. Hodgson and Bachmann [6], [7] further compared the performance of Steer-to-Center, Steer-to-Multiple-Targets, and Steer-to-Orbit methods in a series of experiments. They found that steer-to-center performed best in open environments in which users could change direction freely, but that steer-to-orbit was slightly more effective for a constrained VE like a grocery store. This has implications for the present study, which was conducted in a relatively open VE and used Steer-to-Center as a comparison condition to represent the currently known best method of steering.

2.2 Multi-User Redirected Walking

The ability to support multiple users simultaneously could, in theory, be a strength of RDW. However, work in this area has been quite limited and has not progressed beyond simulation studies [2], [1]. Furthermore, most multi-user approaches require uncertain collision prediction [2] and complicated prevention decisions [1], since steering two users away from each other could send one towards a wall and the other towards a third user. In [2], for example, each user's rendering computer ran simulated RDW algorithms for every other user to predict not just where they were currently heading plus some moderate amount of uncertainty, but also where they might be steered from that path. Any predicted collision was then classified by type (head-on; crossing; rear-end collision; etc.) and a corresponding steering change was implemented independently by each user. This solution was computationally complex, limited by prediction uncertainty, and did not scale well with more than two users. Ironically, multi-user RDW with existing techniques can actually encourage collisions since generalized algorithms are designed to pull all users toward a central area or onto a shared orbit around the center. To solve this, users are sometimes relegated to their own non-overlapping sub-region, rather than using the full tracking space. Some techniques have been developed to allow users to dynamically swap regions to give the illusion that they have more tracking space available [15]. However, this approach requires adjacent users to both approach a shared boundary at the same time in order to switch, limiting its utility. Others have attempted let users handle collision avoidance by displaying visual indicators of nearby users when a collision is imminent [14] [23] to allow users to negotiate avoidance themselves.

2.3 Resetting

Resetting may be used alone or in conjunction with RDW so that virtual navigation can continue safely when a user approaches a tracking area boundary or obstacle. Williams et al. propose some of the earliest resetting methods, including Freeze-Backup, Freeze-Turn, and the 2:1-Turn [33]. The 2:1-turn method has been most influential, and doubles a user's virtual turn relative to their physical turn. This results in a physical turn of 180°, but a 360° turn in the virtual world that allows navigation to continue. Zmuda et al. [34] expanded the idea of a 2:1 rotation in place and introduced the idea of Intelligent Imperceptible Alignment (IIA) to reorient users in a more optimal direction after turning. An alternative method uses *distractors* to induce users to rotate their head while a rotation gain adjusts the viewpoint and results in reorientation [19], [18]. Users can find this quite distracting, however. Others have proposed using overt portals [5] that allow users to physically turn and progress through a portal towards their goal, or imperceptible change blindness [28] that moves doorways and relies on users' unlikeliness to recognize small changes made to a VE out of their view. Similarly, adjacent virtual rooms can be overlapped by more

than half before users detect they are in a geometrically impossible space [29].

2.4 Artificial Potential Fields

Proposed in the 1980s by Krogh [13] and Khatlib [10], [11], [12]. Artificial potential fields are used to solve motion planning problems for mobile robots. In this context, the solution is a continuous motion path that connects a start configuration to a goal configuration. APF does not require knowledge of the entire space to be navigated and reduces computational complexity compared to other approaches, making it possible to continuously generate a dynamic trajectory that adapts to changing conditions. These properties make it a good fit for RDW, and multi-user RDW in particular, as the algorithm seeks to guide users along a dynamically generated path to avoid both static and moving obstacles.

Under APF, the direction of the motion path is determined by a vector field based on robot's position and velocity as well as the relative locations of obstacles and a goal. The potential field is the sum of avoidance vectors associated with obstacles and an attraction vector associated with the goal. At any give time, the robot is accelerated in the direction indicated by the vector sum. In essence, the robot follows a gradient of least resistance that goes away from obstacles and towards the goal. Attraction vectors are not used in the current implementation of APF-RDW since users plan their own goals, but could be included in future iterations to attract users to a position or to another user for social interaction.

APF solutions in robotics typically also include vehicle or manipulator dynamics models that serve to ensure the robot can physically follow the motion path. The parallel here would be ensuring that RDW rates are kept within known imperceptible limits (e.g., [25]), so that human users can adjust their navigation without being disrupted.

3 APF Redirected Walking and Resetting

APF-RDW directs immersed users away from obstacles and into open space. For the purposes of clarity and testing, this work implements a relatively simple version of the APF steering. It is intended as a minimally viable algorithm for proof-of-concept and to spark further work. The algorithms distorts virtual and physical world correspondence only through rotation of the virtual viewpoint (curvature gains and rotation gains). No linear gains are included.

3.1 APF-RDW Algorithm

APF-RDW is based on a total force vector, $\mathbf{t}$, to determine the direction in which the user should move to increase distance from the line segments defining the boundaries of the tracking area and other users. The vector $\mathbf{t}$ is the sum of individual force vectors associated with boundary line segments, $\mathbf{w}_i$, and individual force vectors associated with other users, $\mathbf{u}_j$, and is simply

$$\mathbf{t} = \sum_{i=0}^{n} \mathbf{w}_i + \sum_{j=0}^{m} \mathbf{u}_j \tag{1}$$

where n is the number of line segments defining the bounds of the tracking area and m is the number of other users in the tracking area. $\mathbf{t}$ is re-calculated continuously as the user moves. Force vectors point away from the associated boundary line segments or other users and have a magnitude that is scaled to a length that is inversely proportional to the distance of the user from the closest point on a boundary segment or the position of the other user. The closer a user is to an obstacle, the larger its effect is on $\mathbf{t}$ (Fig. 2).

3.1.1 Individual Force Vector Calculation

For line segments defining tracking area boundaries, individual force vector calculation begins by finding the difference vector between the two dimensional position of the user's center, $\mathbf{p}$, and the closest point on the line segment, $\mathbf{l}_i$.

$$\mathbf{d}_i = \mathbf{p} - \mathbf{l}_i \tag{2}$$

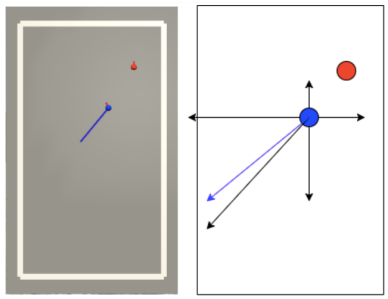

Fig. 2. Left: Generated force vector for blue user with second red user, Right: Diagram depicting hypothetical force vectors for each wall and the other user (black arrows) and the total force vector (blue arrow). Adapted from [9]

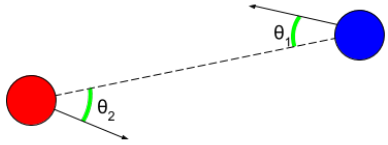

Fig. 3. Relative heading angles, θ_1 and θ_2 (green arcs), for two users (blue and red dots). User movement directions are represented by black arrows.

Potential collisions between users are primarily a concern if they are moving towards one another and they are in close proximity. To take these ideas into account, the magnitude of the forces users exert on one another are altered based on their relative headings and their influence decreases more quickly as the distance between the users increases. For other users in the tracking area, the vector difference is scaled by a relative heading factor, κ, that is based on the average of the cosines of angles between the user movement directions and a line segment connecting the positions of the users (shown in Fig. 3 as θ_1 and θ_2). The cosine values are calculated by taking the dot product of normalized vectors and clamped to values between 0 and 1. For every other user in the tracking space a pair of cosine values is calculated. The relative heading factor is given by

$$\kappa = \text{clamp}\left(\frac{\cos\theta_1 + \cos\theta_2}{2}, 0, 1\right) \tag{3}$$

Using κ, the difference vector for others users, is modified from (2) as follows

$$\mathbf{h}_j = \kappa_j(\mathbf{p} - \mathbf{c}_j) \tag{4}$$

where $\mathbf{c}_j$ is the position of the other user. This has the effect of including the entire difference vector in further calculations if two users are walking directly towards one another or none of the difference for users that are on parallel paths or walking away from one another.

The sum of the distances to all boundary line segments and all other users is simply the sum of the lengths of the difference vectors and is given by

$$d = \sum_{i=0}^{n} \|\mathbf{d}_i\| + \sum_{j=0}^{m} \|\mathbf{h}_j\| \tag{5}$$

where $\mathbf{d}_i$ is calculated using (2) for boundary line segments and $\mathbf{h}_j$ by (4).

To derive $\mathbf{w}_i$, for a boundary segment, the associated difference vector is normalized to unit length and scaled by the ratio of the sum of distances given by (5) to the distance to the individual obstacle.

$$\mathbf{w}_i = \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|}\frac{d}{\|\mathbf{d}_i\|} \tag{6}$$

The individual force vector for another user in the tracking area is similar to (6), with the addition of a user fall off exponent, γ, in the

denominator of the right hand term on the right hand side of the equality and the relative heading factor from (3). It is given by

$$\mathbf{u}_j = \kappa_j \frac{\mathbf{h}_j}{\|\mathbf{h}_j\|} \frac{d}{\|\mathbf{h}_j\|^{\gamma}} \tag{7}$$

The use of γ causes the influence of users to fall off exponentially instead of linearly as is done with line segments representing walls or boundaries.

3.1.2 Rotation Calculation

The remaining calculations used in the APF-RDW algorithm are similar to those described in [6]. However, there are two notable exceptions. The first is that all steering instructions produced by the algorithm are based on the direction of the total force vector given by (1). The second is that steering rates are increased when it is determined that the magnitude of total force vector has become greater than the norm and steering rates are decreased when the total force is less than the norm.

On each update, the virtual viewpoint is incrementally rotated within the virtual world with the goal of causing the user to unconsciously rotate to align their movement direction with the current force vector. The magnitude of the rotation applied to the virtual viewpoint is determined by finding the maximum of three different rate calculations. The calculations produce a *baseline rotation rate* for users standing still, a *moving rotation rate* for walking, and a *head rotation rate* for turning. The maximum of the three is used to calculate the largest incremental rotation that can be imperceptibly applied to redirect the user away from obstacles and other users.

The baseline rotation rate, *baseRate*, is a constant that represents a small amount of redirection that can be applied while the user is standing still without rotating their head [21].

The moving rotation rate guides a user to walk a curved path in the physical world while they follow a straight path in the virtual world. Up to a specified velocity, this rate is increased with faster walking speeds. When linear velocity, v, is below a minimum movement threshold, ν, the moving rotation rate will be zero. The moving rotation rate is given by

$$movingRate = \begin{cases} 360\left(\frac{v}{2\pi r}\right), & \text{if } v > \nu. \\ 0, \text{ Otherwise.} \end{cases} \tag{8}$$

where the constant, r, is the radius of the arc on which a walking user is being redirected [26].

The head rotation rate acts by amplifying or compressing user head rotations. When the user is walking, amplification and compression of head movements can lead to loss of balance. Thus, the head rotation rate will be zero whenever linear velocity exceeds a minimum movement threshold. Otherwise, the more a user rotates their head, the more those rotations can be altered. The head rotation rate is determined as follows:

$$headRate = \begin{cases} \dot{\psi}\left(angRateScale\right), & \text{if } v < \nu. \\ 0, \text{ Otherwise.} \end{cases} \tag{9}$$

where $\dot{\psi}$ is the yaw rate of the head and *angRateScale* is an angular rate scaling factor that is varied depending upon whether the physical rotation of the head is being amplified or compressed in the virtual world.

3.1.3 Rotation Scaling

The determined moving and head rotation rates are scaled based on the magnitude of the total force vector, $\mathbf{t}$. In a rectangular tracking area like the one used here, and for a user walking a straight virtual path, the desired result of this modification would be an oval shaped physical track instead of a circle. The rotation scale factor is is given by:

$$s = \frac{\|\mathbf{t}\|}{\|\mathbf{t}_a\|}\left(scaleMultiplier\right) \tag{10}$$

where $\|\mathbf{t}_a\|$ is the average length of the total force vector. It is multiplied times the results of (8) and (9) before each is clamped to a maximum

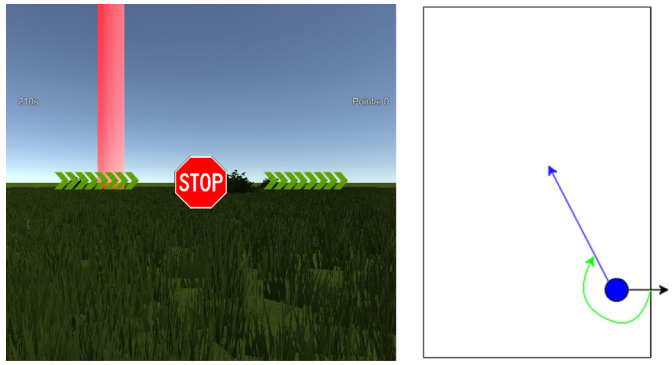

Fig. 4. Left: User's view at the beginning of the reset, instructing them to stop and turn in the direction of the green arrows. Right: Example of a user (black heading) reset rightward to face the heading indicated by the force vector (blue). Adapted from [9]

value. For instance the final scaled and clamped moving rotation rate becomes

$$movingRate' = \text{clamp}\left(s(movingRate), 0, maxMoveRate\right) \tag{11}$$

The final head rotation rate is derived in a similar manner and is clamped to a *maxHeadRate*. The magnitude of the incremental rotation to be applied to the virtual viewpoint in the next frame is calculated by:

$$\max(baseRate, movingRate', headRate')\Delta t \tag{12}$$

This rotation is then signed to result in a turn in a direction that will result in aligning the user movement direction with the total force vector as quickly as possible while still being imperceptible.

3.2 Artificial Potential Field Resetting Algorithm

When a collision with an obstacle or another user is imminent, APF-RDW is suspended and a reset is triggered. APF-R utilizes the total force vector at the beginning of the reset as the desired final orientation of the user following the reset. When a reset begins, a UI element is displayed indicating that the user should stop as well as the direction in which they should turn (Fig. 4). Users then turn in place until physically aligned with the force vector while the corresponding virtual rotation is scaled to equal 360° resulting in a return to the original virtual heading. In this way, the method is similar to the 2:1 turn reset method [33]. The desired turn is in the direction of larger of the two angles between the user's movement direction and the force vector at the start of the reset (Fig. 4). This reduces the gain required to yield a 360° virtual turn and minimizes perceptual distortion. The rotation gain during reset is given by:

$$resetRotationGain = \left| \frac{360 - |rotateAngle|}{rotateAngle} \right| \tag{13}$$

Force vectors are not updated during the reset, so as not to change the target heading during the turn.

3.3 Constant Parameter Values used in this Study

The constant parameter values used in this study are presented in the Table 1. Values associated with the limitations of human perception were taken from Steineke et al. [26], [27], [24], [25] or previous work by the authors [8], [6], [7] that was reported by users as imperceptible. The user fall-off exponent, γ, in (7) and the *scaleMultiplier* in (10) are both specific to the APF-RDW algorithm and were set via manual tuning based on observed algorithm performance in pilot testing. The average length of the total force vector, $\|\mathbf{t}_a\|$, is dependent on tracking area dimensions and was ≈ 15.0 during this study.

Table 1. Constant Parameter Values

Constant	Value	Constant	Value
baseRate	1.5°/sec.	γ	1.5
v	0.1 m/sec.	r	7.5 m
scaleMultipler	2.5	*angRateScale* (compress)	0.85
angRateScale (dilate)	1.3	*maxMoveRate*	$15°/sec.$
maxHeadRate	$30°/sec.$		

4 Experimental Evaluation

Experimental evaluations of APF-RDW and APF-R were completed through both live user experiments and simulations using previously recorded live user paths. Evaluations were conducted for both single users and multiple concurrent users. In all experiments the immersed user task within the virtual world was the same - collecting a series of posts - and was independent for each user. Concurrent users could not see each other.

The experiment was conducted in the HIVE [32], a 25 m x 45 m open gymnasium. Users' head position was tracked using a WorldViz PPT 12-camera system modified for large-area tracking and streamed via Wi-Fi. Rendering and data collection was performed by Alienware 13 R3 laptops with Nvidia GeForce GTX 1060 GPUs. Laptops were mounted on a back-packing frame with external batteries to boost runtime. Users wore an Oculus Rift CV1, which also provided orientation tracking. The Unity game engine (5.6.3f) [31] was used to render the VE, log data, and implement APF-RDW and APF-R. Simulations were also conducted in Unity, using an accelerated time scale.

Participants were greeted in the HIVE and gave informed consent after a brief introduction to the task and equipment. Participants started in one of three locations, facing North: the room's center, or 10m offset to the NW or SE. Single users were started in the SE location. Dual users were started in the NW and SE locations. Participants began in the middle of a virtual field and collected points by walking to colored posts. In each trial, eight colors were used with randomly assigned point values per color. Three different-colored posts were spawned in an arc roughly in front of the user at a distance of 50m, each 30° apart. When participants chose a direction and walked into the post they were awarded points and the existing posts were replaced with a new set fifty meters further, also 30° apart. Participants repeated this cycle for 4 min while attempting to determine higher-point colors through trial and error and maximize their score. Each participant was asked to complete 4 trials.

4.1 Live User Experiments

Two different sets of live users were used to evaluate the effectiveness of APF-RDW for *Single Users*, and *Multiple Users*. For single users, AFP-RDW was compared to a steer-to-center (STC) approach previously demonstrated to produce the best results in open VEs like the one used here [21], [6], [7]. The same code-base and parameters were used for both algorithms to ensure comparability. For STC, the force vector was fixed to always point towards the center, and implemented traditional steering rate dampening for users facing towards the middle or nearing it [21], [8]. This prevents undesirable and noticeable changes in steering direction.

For multiple users, APF-RDW was compared to a control condition in which no steering occurred. While it would be ideal to compare APF-RDW to an existing state-of-the-art multi-user RDW algorithm, no such algorithms exist for more than two users. Furthermore, existing two-user approaches have known limitations that make them difficult to scale to more users [2] [1]. Thus, the multi-user experiment below is best considered a proof-of-concept for APF-RDW, in which the no-steering condition provides a baseline of how many wall and user resets might otherwise be expected. In all conditions, users were reset using APF-R.

For both experiments, the primary metrics were:

- Average number of resets near a wall or another user (multi-user only)
- Average distance traveled between resets
- Average steering rate
- Average physical distance from center
- Number of posts collected (task performance)
- Total distance traveled

Live participants were asked to complete a simulator sickness questionnaire (SSQ) both before and after the experiment. Simulator sickness was generally not a factor, with SSQ scores averaging 0.87 prior to the experiment and 2.11 afterwards (out of a maximum of 48). SSQ scores were not recorded in between sessions to test differences between algorithms. The perceptibly of redirection was also not measured. Because STC and APF-RDW used identical gains and steering rates that were previously shown to be imperceptible and comfortable for users [8] [6] [24] [25], there was no reason to expect any differences between algorithms on these measures.

4.1.1 Live Single User Results

21 users participated in the single-user experiment. One participant was removed due to technical difficulties. The final sample of 20 participants was aged 19 - 24, including 13 males and 7 females recruited from a university in the Midwestern United States. Participants were compensated with course credit. Results for the single user experiment are listed in Table 2 and described below. Data was collapsed across multiple trials of the same algorithm, and compared with paired-samples t-tests to determine differences between APF-RWD and STC.

Resets: Participants reached the tracking boundaries less often with APF (1.10 per trial, or 0.275 per min) than with STC (3.20, or 0.80 per min). This was both statistically significant ($t(19) = 5.95$, $p < .0001$) as well as functionally significant; it represents a 65.6% reduction in presence-breaking interruptions. Users were also able to travel further between interruptions (242.4 m for APF; 170.0. m for STC; $t(19) = 5.14$, $p < .0001$). Distance between resets was defined as the average distance walked per trial divided by the average number of resets plus one. Thus, zero resets yields one continuous path, one reset divides the distance into two sections, etc. Redirecting users with APF allowed them to travel more than 70 m further between resets, which is a conservative estimate, as more than a third of participants (7) experienced zero resets during at least one trial with APF-RDW and could have traveled further.

Average Steering Rate: Much of the reduction in resets can be attributed to the higher average rate of redirection for APF (9.64°/sec) than for STC (8.16°/sec; $t(19) = 9.69$, $p < .0001$). Because both algorithms used identical parameter values, this reflects categorical differences of when and where to steer users. While STC dampens steering (sometimes to 0°/sec) for users who are heading towards the center of the tracking space, APF continually seeks useful areas to steer towards, taking advantage of the full space. AFP is also designed to be sensitive to wall proximity and temporarily increase rates up to maximally imperceptible levels. STC is not designed to account for walls or proximity. These high-level differences in strategy lead to higher average steering rates for APF, and thus more effective steering.

Distance to Center: Users tended to be slightly closer to the center of the tracking area on average with AFP (7.69 m) than with STC (8.54 m; $t(20) = 5.81$, $p < .001$). This may seem counter-intuitive, given that STC is explicitly guiding people to the center and AFP is not. However, one can consider APF's strategy for single users as guiding them away from walls and thus towards the center *region* of the tracking space rather than the center *point*.

Task Performance: The final measures relate to users' ability to perform their primary task. With fewer resets interrupting them, participants traveled a higher total distance with APF-RWD (307.5 m) versus STC (280.6 m; $t(19) = 4.39$, $p < .001$). They also collected more virtual posts with APF (5.1 vs 4.5;$t(19) = 3.33$, $p < .01$).

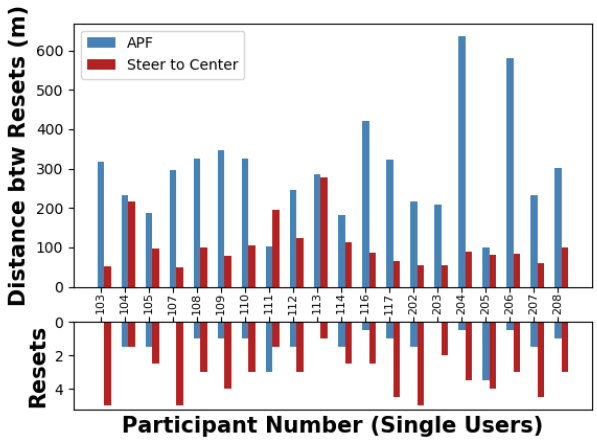

Fig. 5. Mean distance traveled between resets for each participant in the single-user experiment is shown in the top panel, with their mean number of resets per condition displayed below for reference. For participants with zero resets in a given condition, the figure shows their total distance traveled.

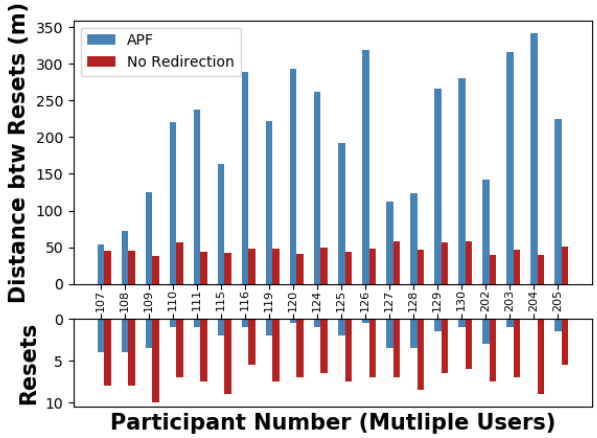

Fig. 6. Mean distance traveled between resets for each participant in the multi-user experiment is shown in the top panel, with their mean number of resets per condition displayed below for reference. For participants with zero resets in a given condition, the figure shows their total distance traveled.

4.1.2 Live Single User Discussion

For single users, steering and task performance was markedly better with APF than with STC, which was previously shown to be the best-performing algorithm in this type of task. The magnitude of APF's advantage varied across users due to individual differences in walking style, the propensity to look around, and navigational choices, but the direction of this effect was highly consistent (c.f. Fig. 5). APF was not merely better on average, but outperformed STC for every user but one.

4.1.3 Live Multi-User Results

A different set of 35 users participated in the live multi-user experiment, in groups of 2 or 3. Due to repeated technical problems with the position tracking system, 15 users were unable to finish the experiment or experienced large jumps in position that made their data unreliable. These problems were most pronounced with 3 concurrent users, as user positions would occasionally swap and/or follow each other. Planned tests with higher numbers of concurrent users unfortunately had to be dropped in favor of simulation. Final analyses were based on 20 users with reliable tracking data. Results were initially analyzed in the context of a 2 (steering / no steering) x 2 (n users: 2 or 3) repeated measures ANOVA. However, given the small sample sizes of 3 concurrent users, data was collapsed across this dimension and the two steering conditions were compared in paired-sample t-tests. Linear regressions were also conducted where appropriate.

Resets: As one might expect, there were significantly more wall collisions for un-steered users (6.33 per trial, or 1.58 per min) than for APF-RDW (1.35, or 0.33 resets per min; $t(19) = 17.22$, $p < .0001$). More importantly, APF successfully reduced the number of user-to-user collisions by about 50%, from an average of 1.05 user collisions per trial to only 0.53 per trial ($t(19) = 4.27$, $p < .001$).

Average Steering Rate: The average steering rate for APF-RDW was 9.90°/sec, which was nearly identical to rates in the single-user experiment (9.64°/sec). This indicates that the algorithm changed *where* it was steering users, but did not measurably increase the rate.

Task Performance: APF-RDW allowed participants to travel larger total distances (290.6 m) than the control condition (247.9 m; $t(19) = 3.41$, $p < .01$). They also collected about 40% more posts (4.13 vs 2.95). Both of these effects were expected given the high number of resets with no steering, which frequently interrupted the task.

4.1.4 Live Multi-User Discussion

These results demonstrate the viability of APF-RDW to redirect at least 3 live users in the same tracking area using a relatively simple approach. Given the low number of user collisions and near-zero increase in steering rates, it is likely that a room of this size (45m x 25m) could support at least a few more users. It should be noted that the total number of user collisions was relatively small in both conditions due to the rarity of only 2 or 3 users in a large space coming together by chance. Such collisions may to increase with additional users, which is tested in simulation below.

4.2 Simulation Experiments

4.2.1 Simulation Method

Fifty two paths were taken from individual trials of the live user experiments to serve as navigation data for the simulations. Each path represented a user's intended virtual path, complete with any indecision, velocity variation, and spontaneity, and was used to construct a series of redirected physical paths in the presence of other simulated users. Because the position tracking system had experienced problems during the live experiments, log files that contained excessive tracking jitter or obvious errors were excluded. Also, simulation runs in which tracking jitter caused a user to temporarily leave the tracking area were discarded (e.g., if a participant was near a wall when position jitter occurred), as the force vector of the nearby wall would repel the user even further out of bounds. Wall resets and user resets that occurred during the live trials were omitted from the logs to create uninterrupted virtual paths.

Simulations were run with combinations of 1 to 8 concurrent users, chosen randomly, and included the same APF-RDW and no-steering control conditions as the live multi-user experiment. Since the purpose of this study was to focus on multi-user implications, STC was not simulated for single users or any other condition. For single-user simulations, paths were run individually and yielded 50 successful trials for APF-RDW (2 out of bounds) and 48 successful trials without steering (4 out of bounds). For multiple users, paths were run in random combinations to create 500 unique redirected paths for each condition and number of users. Out-of-bounds runs were discarded and replaced. In sum, 7,098 successfully redirected paths were generated. To facilitate comparisons with the live experiment, it was assumed that users were situated in the same physical tracking space as above. For the control condition, APF-R was again used to handle resetting prior to possible wall or user collisions. The same metrics were collected with the exception of counting the number of posts collected, since posts were primarily used to motivate the original live users. Simulation results for APF-RDW are summarized in Table 3 and are described

Table 2. Summary of Live-Use Experiment Results (± 95% CIs)

	Single-User STC	Single-User APF	Two-User Control	Two-User APF
Avg. Dist. per Reset (m)	170.0 ± 21.1	242.4 ± 23.0	26.2 ± 1.3	117.3 ± 19.0
Avg. Total Resets Per Trial	3.20 ± .38	1.10 ± .29	7.38 ± 0.37	1.88 ± .40
Avg. Wall Resets Per Trial	3.20 ± .38	1.10 ± .29	6.33 ± .33	1.35 ± .32
Avg. User Resets Per Trial	n/a	n/a	1.05 ± .20	0.53 ± .21
Avg. Dist. from Center (m)	8.54 ± .17	7.69 ± .16	11.09 ± .58	7.95 ± .21
Avg. Dist. Traveled (m)	280.6 ± 16.6	307.5 ± 13.3	247.9 ± 9.8	290.6 ± 13.7
Avg. Steering Rate (deg./s)	8.16 ± .36	9.64 ± .44	n/a	9.90 ± .30
Avg. Posts Collected	4.53 ± .27	5.08 ± .32	2.95 ± .2	4.13 ± .32

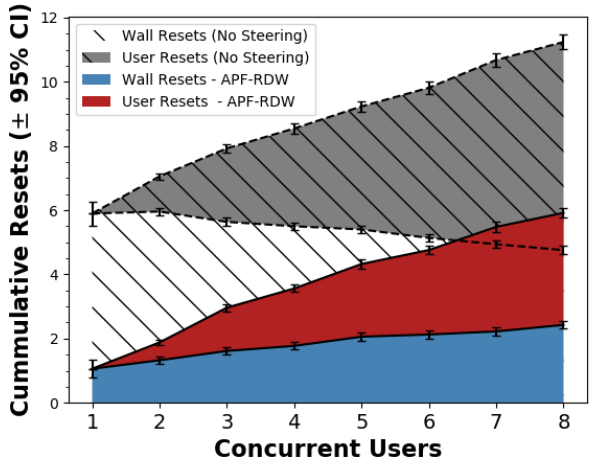

Fig. 7. Cumulative number of resets (wall resets + user resets) during a simulated 4 min walking task for APF-RDW steered and unsteered users.

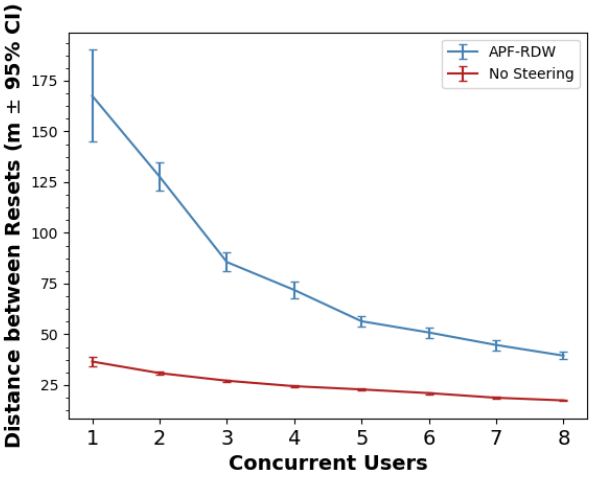

Fig. 8. Mean distance traveled in the VE between resets for multiple simulated users who are being redirected with APF versus un-steered users. Adding concurrent users quickly decreases the amount of travel between resets, but the cost per person levels off around 4-5 users.

in detail below. Results for the control simulation are summarized in the text for context. Data was analyzed in the context of a 2 (RDW condition) x 8 (concurrent users) ANOVA unless otherwise noted.

4.2.2 Multi-User Simulation Results

Resets: The number of wall resets and user resets are illustrated in Fig. 7. The grey shaded area illustrates the cumulative number of resets per 4 min trial for un-steered users. The colored region illustrates the same data for redirected users. As expected, RDW decreases the total number of resets compared to no steering, in this case by about 47%. Of more interest is the interplay between wall resets and user resets. For un-steered users, the number of wall collisions significantly *decreases* as more users are added, falling from 5.9 wall resets per trial for single users to 4.75 resets with 8 users (simple main effect: $F(7, 3547) = 39.12$, $p < .001$). While fewer wall resets seems desirable, in this case it indicated that users were spending less time walking and more time being reset.

For APF-RDW users, the trend was reversed; wall resets increased as more users were added (simple main effect: $F(7, 3549) = 39.31$, $p < .001$), pushing each other towards the walls. Regression analyses indicated that wall resets for APF-RDW increased at a rate of .18 resets per trial per additional user. It is encouraging that this increase was linear, since Multi-user RDW would become untenable quite quickly if resets increased at an exponential rate.

User-to-user resets increased significantly in both conditions. This makes sense, given that more users provide more opportunities for collision. The increase in user resets was significant for APF-RDW ($F(7, 3542) = 230.38$, $p < .001$) and for un-steered users ($F(7, 3540) = 415.52$, $p < .001$). A separate 2 (RDW condition) x 8 (users) analysis of covariance indicated the rate of increase in resets for un-steered users was significantly higher than for redirected users ($F(1, 7098) = 385.04$, $p < .001$). APF-RDW not only reduced the *total* number of user-to-user collisions, but also decreased the *rate of increase* per additional user. Separate linear regressions for each condition indicated that user resets increased 0.89 resets per trial per additional user without steering as compared to only 0.48 for APF-RDW. This represents a 45.4% reduction in the number of user-to-user collisions incurred by each additional user.

Because the relationship between user resets and wall resets is not independent while redirection is being used, it is instructive to also consider the *total* number of resets. For APF-RWD, the total number of resets increased linearly such that

$$totalResets = 0.815 + (0.661 * n)$$

where n is the number of concurrent users. Thus, each additional user contributed about 0.66 resets to each other user over a 4 min. walking task in a 25 m x 45 m tracking area. This rate of increase is such that eight concurrently redirected users each experienced the same number of total resets (5.92) as a single user would normally encounter if no steering was applied (5.90). With the formula above, developers could target an acceptable balance between the number of users and the number of resets expected. For example, if it was desirable to have one reset per min, then the present tracking area could support about 5 concurrent users.

Average Steering Rate: In the live user experiments above, there was no significant increase in average steering rate from 1 to 2-3 users, but it seems reasonable that the rate might eventually increase with larger groups as users are pushed apart more often. In the range of users tested

Table 3. Summary of Average Multi-user Simulation Results (± 95% CIs)

Number of Users	1	2	3	4	5	6	7	8
Avg. Dist. per Reset (m)	167.6 ± 22.7	127.7 ± 6.9	85.6 ± 4.6	71.9 ± 4.1	56.4 ± 2.5	50.9 ± 2.5	44.7 ± 2.5	39.5 ± 1.8
Avg. Resets Per Trial	1.06 ± .28	1.88 ± .12	2.96 ± .15	3.56 ± .16	4.32 ± .17	4.76 ± .17	5.48 ± .18	5.92 ± .17
Avg. Wall Resets Per Trial	1.06 ± .28	1.32 ± .10	1.61 ± .11	1.77 ± .11	2.06 ± .12	2.13 ± .12	2.22 ± .12	2.43 ± .12
Avg. User Resets Per Trial	n/a	0.56 ± .07	1.35 ± .11	1.79 ± .11	2.26 ± .15	2.63 ± .14	3.26 ± .16	3.49 ± .15
Avg. Dist. from Center (m)	7.54 ± .31	8.74 ± .09	9.33 ± .09	9.72 ± .10	10.01 ± .11	10.31 ± .11	10.49 ± .12	10.65 ± .12
Avg. Dist. Traveled (m)	278.7 ± 12.4	278.9 ± 3.8	269.7 ± 3.6	262.5 ± 3.6	259.4 ± 3.7	250.7 ± 3.8	246.8 ± 3.5	243.9 ± 3.6
Avg. Steering Rate (°/s)	9.07 ± .45	9.31 ± .14	9.43 ± .13	9.55 ± .13	9.69 ± .14	9.76 ± .16	9.88 ± .14	10.12 ± .16

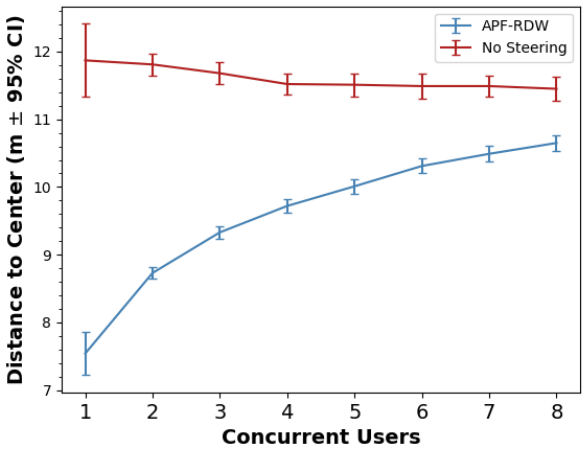

Fig. 9. Mean distance-to-center for simulated users who were steered with APF versus un-steered users. As more users were added, APF-RDW automatically spread them out in the rectangular tracking area while limiting their proximity to walls.

here, this was indeed the case. A one-way ANOVA indicated a small but measurable increase in steering rates with additional users ($F(7, 3542) = 13.21$, $p < .001$), which can be described with the equation

$$steeringRate = 9.037 + (0.128 * n)$$

where n is the number of concurrent users. Each additional user increased the average steering rate by a little more than 0.1°/sec. Because the steering rate is clamped to a maximum of 15°/sec, this means the algorithm could theoretically support 47 users in a gym-sized tracking space before it could no longer steer users fast enough. Of course, the increase in presence-breaking resets would be a far more relevant limiting factor in this case.

Physical Distance-to-Center: As mentioned in the introduction, one feature of APF-RDW is that it naturally takes advantage of available space in the tracking area. In the rectangular tracking area used here, for example, users should naturally disperse along the elongated axis of the room. This should result in larger distance-to-center metrics with higher numbers of users until some point where spreading out further is not possible. This was indeed the case, as can be seen in Fig. 9. While distance-to-center remained fairly constant (and high) for un-steered users, distance-to-center increased dramatically with the addition of the first few APF-RDW users and then leveled off. The ANOVA indicated a significant interaction between condition and number of users ($F(7, 7082) = 64.28$, $p < .001$), with both linear and quadratic components for APF-RDW (p's $< .001$).

Task Performance: To evaluate task performance, a time-on-task measure was constructed to analyze the amount of time spent walking rather than resetting. The ANOVA indicated a main effect of the number of users as time-on-task decreased for both conditions ($F(7, 7082) = 321.60$, $p < .001$) as well as a main effect of condition ($F(1, 7082) = 3465.29$, $p < .001$), as APF-RDW users were reset far less often. A lack of interaction between the condition and the number of users suggested that the effect was similar for both conditions ($F(7, 7082) = 1.43$, $p = .325$). On average, users would spend an extra 4.5 sec resetting per 4 min trial for each additional user. APF-RDW has a lower baseline of rests, however, to the point where eight concurrent APF-RDW users each spent about the same time in resets (47.4 sec) as single user with no steering (45.0 sec).

The mean distance traveled between resets was also calculated for each condition and number of concurrent users and analyzed in a 2 x 8 ANOVA. These results are shown in in Fig. 8. As one might expect, redirected users were able to travel farther than un-steered users ($F(1, 7082) = 2766.09$, $p < .001$), and each user could travel less distance as more concurrent users were added ($F(7, 7082) = 299.14$, $p < .001$). Importantly, though, these two factors interacted significantly ($F(7, 7082) = 165.89$, $p < .001$). For un-steered users, the fixed space between walls limited travel even for single users (39.5 m per reset) and adding more users decreased travel linearly to 17.4 m per reset for groups of 8. APF-RDW users were able to travel large distances between resets when alone (167.6 m). This distance dropped off rapidly as a few users were added, however, and then leveled out. As with other metrics, though, each of the 8 concurrent users in the simulation were able to travel farther between resets on average than a single user who was un-steered. These results suggest that there may be notable task performance implications between single-user RDW and multi-user RDW in general, but that the differences between, say, four versus seven users is more subtle.

4.2.3 Multi-User Simulation Discussion

The results of the simulation confirm and extend the results of the live multi-user experiment. Using logged live-user paths with a wide array of individual differences in navigation, APF-RDW was able to redirect up to 8 users in a simulated version of the 25m x 45m real-world tracking space. Users were steered around each other and away from walls with only small increases in steering rates and a modest increase in resets per additional user. These results indicate that the algorithm scales easily and is viable for a large numbers of users. Unlike previous two-user algorithms, it required no special considerations for more than two users or complex collision prediction to operate; the same code base was used for single users and arbitrary numbers of concurrent users. Each new user added a single force vector to be considered. The simulation suggests that the limiting factor of APF-RDW is not the algorithm itself, but the space available to host users. Further studies are underway to test its performance in different sized rooms.

5 Conclusion

In this paper, a new approach to RDW has been proposed in which users are repelled by walls, obstacles, and other users in order to redirect to the safest area. Borrowing the concept of Artificial Potential Fields (APF) from the robotics literature, force vectors are calculated for immersed users to redirect them in a way that utilizes the full tracking area and efficiently accounts for multiple users.

This work also presents, to the authors' knowledge, the first live demonstration of a generalized RDW algorithm allowing mutiple users to share the same tracking space. The number of concurrent users in this case was limited to three due to position tracking difficulties, but

simulation results indicated that the algorithm is capable of scaling up to significantly more users with ease. Results suggested the live user experiment could have supported 5-6 concurrent users with a reasonable number of resets. This marks a significant improvement over existing RDW approaches and is perhaps the most important contribution of the present work.

5.1 Limitations

It is worth noting the APF-RDW and APF-R algorithms presented here are intended as a minimally viable implementation and not fully optimized. There are doubtlessly many ways the algorithm could be improved through future work. There was no formal effort to balance the strength of the force vectors of walls relative to other users, for example, or to optimize the distance at which each ceases to push. A developer could reasonably choose to increase the strength of other-user force vectors to reduce user resets in favor of resetting them near the walls. Various forms of path prediction might also be integrated to improve performance. Attraction force-vectors might be included to reflect predicted paths, to pull users towards each other, or to attract towards a goal. Such questions are ripe for future study.

Even so, the simplified APF approach detailed here out-performed an equivalent STC approach for single users, and efficiently supported large numbers of concurrent users. This approach required no collision prediction or micro-managing of where to steer each user to prevent a collision, issues which have plagued prior implementations [2] [1]. Because of its simplicity, APF has less potential for edge cases and limitations that cause it to break down. In the current simulations, for example, the only situation that caused the algorithm to fail was tracking jitter that artificially reported a user as being outside the tracking area. Because the walls repel the user, and more strongly when they are nearby, such users were rapidly pushed further beyond the wall. Even this limitation, though, could be fixed by constraining force vectors to never point out of bounds.

Some potential strengths of APF-RDW also remain unexplored. As noted in the introduction, APF can theoretically work well in odd-shaped tracking spaces or adjust to rooms of different sizes. Because it is designed to take walls into account and push users into open space, APF could theoretically function in L-shaped or other spaces in which traditional approaches are not optimal. The STC approach, for example, assumes a circular tracking space by accounting only for the distance and direction of the user from the center, and would effectively ignore the extra wing of the tracking space or guide a user through the corner towards the tracking origin. APF-RDW could guide users around such corners with no modification to the algorithm, or allow near-zero redirection in the center of extremely large spaces. Further studies are underway to test these predictions.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grant no. 1423112.

REFERENCES

[1] 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)*, vol. 00, pp. 91–98, 2017. doi: 10.1109/VR.2017.7892235

[2] E. R. Bachmann, J. Holm, M. A. Zmuda, and E. Hodgson. Collision prediction and prevention in a simultaneous two-user immersive virtual environment. In *2013 IEEE Virtual Reality (VR)*, pp. 89–90, March 2013. doi: 10.1109/VR.2013.6549377

[3] D. Engel, C. Curio, L. Tcheang, B. Mohler, and H. H. Bülthoff. A psychophysically calibrated controller for navigating through large environments in a limited free-walking space. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pp. 157–164. ACM, New York, NY, USA, 2008. doi: 10.1145/1450579.1450612

[4] T. Field and P. Vamplew. Generalised algorithms for redirected walking in virtual environments. In *AISAT2004: International Conference on Artificial Intelligence in Science and Technology*, pp. 21–25. University of Tasmania, 2004.

[5] S. Freitag, D. Rausch, and T. Kuhlen. Reorientation in virtual environments using interactive portals. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 119–122, March 2014. doi: 10.1109/3DUI.2014.6798852

[6] E. Hodgson and E. R. Bachmann. Comparing four approaches to generalized redirected walking: Simulation and live user data. *IEEE transactions on visualization and computer graphics*, 19(4):634–643, 2013.

[7] E. Hodgson, E. R. Bachmann, and T. Thrash. Performance of redirected walking algorithms in a constrained virtual world. *IEEE transactions on visualization and computer graphics*, 20(4):579–587, 2014.

[8] E. Hodgson, E. R. Bachmann, and D. Waller. Redirected walking to explore virtual environments: Assessing the potential for spatial interference. *ACM Transactions on Applied Perception (TAP)*, 8(4):22, 2011.

[9] C. Hoffbauer. Multi-user redirected walking and resseting utilizing artificial potential fields. Master's thesis, Miami University, Oxford, Ohio, USA, 2018.

[10] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 500–505, March 1985. doi: 10.1109/ROBOT.1985.1087247

[11] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, February 1987. doi: 10.1109/JRA.1987.1087068

[12] O. Khatib. *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*, pp. 396–404. Springer New York, New York, NY, 1990.

[13] B. KROGH. A generalized potential field approach to obstacle avoidance control. *Proc. SME Conf. on Robotics Research : The Next Five Years and Beyond, Bethlehem, PA, 1984*, pp. 11–22, 1984.

[14] E. Langbehn, E. Harting, and F. Steinicke. Shadow-avatars: A visualization method to avoid collisions of physically co-located users in room-scale vr. In *Proceedings of IEEE Workshop on Everyday Virtual Reality (WEVR)*, 2018.

[15] S. Marwecki, M. Brehm, L. Wagner, L.-P. Cheng, F. F. Mueller, and P. Baudisch. Virtualspace - overloading physical space with multiple virtual reality users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 241:1–241:10. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173815

[16] C. T. Neth, J. L. Souman, D. Engel, U. Kloos, H. H. Bulthoff, and B. J. Mohler. Velocity-dependent dynamic curvature gain for redirected walking. *IEEE transactions on visualization and computer graphics*, 18(7):1041–1052, 2012.

[17] 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, Mar 2018. doi: 10.1109/MCG.2018.111125628

[18] T. Peck, H. Fuchs, and M. Whitton. Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. In *Virtual Reality Conference (VR), 2010 IEEE*, pp. 35–38, March 2010. doi: 10.1109/VR.2010.5444816

[19] T. Peck, M. Whitton, and H. Fuchs. Evaluation of reorientation techniques for walking in large virtual environments. In *Virtual Reality Conference, 2008. VR '08. IEEE*, pp. 121–127, March 2008. doi: 10.1109/VR.2008.4480761

[20] T. C. Peck, H. Fuchs, and M. C. Whitton. Evaluation of reorientation techniques and distrators for walking in large virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):383, 2009.

[21] S. Razzaque. *Redirected Walking*. PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 2005. AAI3190299.

[22] S. Razzaque, Z. Kohn, and M. C. Whitton. Redirected walking. In *Proceedings of EUROGRAPHICS*, vol. 9, pp. 105–106. Citeseer, 2001.

[23] A. Scavarelli and R. J. Teather. Vr collide! comparing collision-avoidance methods between co-located virtual reality users. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, pp. 2915–2921. ACM, New York, NY, USA, 2017. doi: 10.1145/3027063.3053180

[24] F. Steinicke, G. Bruder, K. Hinrichs, J. Jerald, H. Frenz, M. Lappe, J. Herder, S. Richir, and I. Thouvenin. Real walking through virtual environments by redirection techniques. *Journal of Virtual Reality and Broadcasting*, 6(2), 2009.

[25] 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, Jan 2010. doi: 10.1109/TVCG.2009.62

[26] F. Steinicke, G. Bruder, J. Jerald, and M. Frenz, H.and Lappe. Analyses

of human sensitivity to redirected walking. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pp. 149–156. ACM, New York, NY, USA, 2008. doi: 10.1145/1450579.1450611

[27] F. Steinicke, G. Bruder, T. Ropinski, and K. Hinrichs. Moving towards generally applicable redirected walking. In *Proceedings of the Virtual Reality International Conference (VRIC)*, pp. 15–24. IEEE Press, 2008.

[28] E. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte. Leveraging change blindness for redirection in virtual environments. In *Virtual Reality Conference (VR), 2011 IEEE*, pp. 159–166, March 2011. doi: 10.1109/VR.2011.5759455

[29] E. A. Suma, M. Azmandian, T. Grechkin, T. Phan, and M. Bolas. Making small spaces feel large: Infinite walking in virtual reality. In *ACM SIGGRAPH 2015 Emerging Technologies*, SIGGRAPH '15, pp. 16:1–16:1. ACM, New York, NY, USA, 2015. doi: 10.1145/2782782.2792496

[30] E. A. Suma, G. Bruder, F. Steinicke, D. M. Krum, and M. Bolas. A taxonomy for deploying redirection techniques in immersive virtual environments. In *2012 IEEE Virtual Reality Workshops (VRW)*, pp. 43–46, March 2012. doi: 10.1109/VR.2012.6180877

[31] Unity. Unity 3d game engine. Retrieved from https://unity3d.com/unity.

[32] D. Waller, E. R. Bachmann, E. Hodgson, and A. C. Beall. The hive: A huge immersive virtual environment for research in spatial cognition. *Behavior Research Methods*, 39(4):835–843, Nov 2007. doi: 10.3758/BF03192976

[33] 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*, APGV '07, pp. 41–48. ACM, New York, NY, USA, 2007. doi: 10.1145/1272582.1272590

[34] M. Zmuda, E. R. Bachmann, E. Hodgson, and J. Wonser. Improved resetting in virtual environments. In *Virtual Reality Conference (VR), 2013 IEEE*, pp. 91–92. IEEE, March 2013.

[35] 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.