Towards Physically Interactive Virtual Environments: Reactive Alignment with Redirected Walking

Jerald Thomas
thoma891@umn.edu
University of Minnesota

Courtney Hutton Pospick
hutto070@umn.edu
University of Minnesota

Evan Suma Rosenberg
suma@umn.edu
University of Minnesota

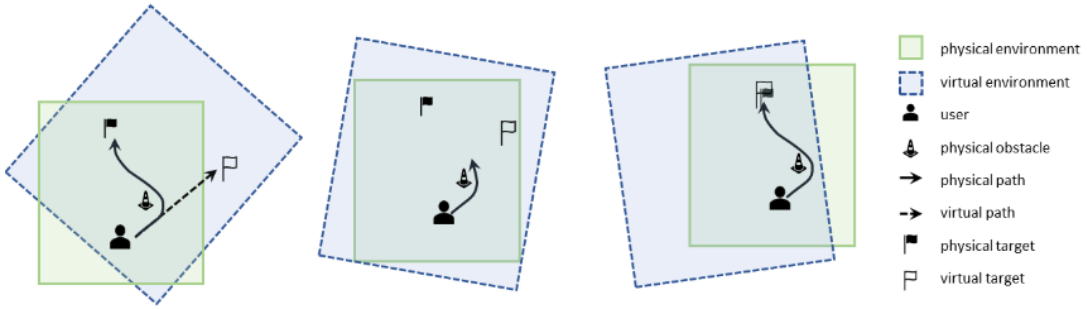

Figure 1: A demonstration of environmental alignment with obstacle avoidance. The left image shows the perceived path to a virtual target and the physical path to a proxy target. The middle and right images show the progression of the physical path to environmental alignment. When this occurs, the user can interact with the proxy target substituting for the virtual target.

ABSTRACT

Interactions with the physical environment, such as passive haptic feedback, have been previously shown to provide richer and more immersive virtual reality experiences. A strict correspondence between the virtual and real world coordinate systems is a staple requirement for physical interaction. However, many of the commonly employed VR locomotion techniques allow for, or even require, this relationship to change as the experience progresses. The outcome is that experience designers frequently have to choose between flexible locomotion or physical interactivity, as the two are often mutually exclusive. To address this limitation, this paper introduces reactive environmental alignment, a novel framework that leverages redirected walking techniques to achieve a desired configuration of the virtual and real world coordinate systems. This approach can transition the system from a misaligned state to an aligned state, thereby enabling the user to interact with physical proxy objects or passive haptic surfaces. Simulation-based experiments demonstrate the effectiveness of reactive alignment and provide insight into the mechanics and potential applications of the proposed algorithm. In the future, reactive environmental alignment can enhance the interactivity of virtual reality systems and inform new research vectors that combine redirected walking and passive haptics.

VRST '20, November 1–4, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-7619-8/20/11…$15.00
https://doi.org/10.1145/3385956.3418966

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Interaction techniques**.

KEYWORDS

virtual reality, redirected walking, alignment, locomotion

ACM Reference Format:
Jerald Thomas, Courtney Hutton Pospick, and Evan Suma Rosenberg. 2020. Towards Physically Interactive Virtual Environments: Reactive Alignment with Redirected Walking. In *26th ACM Symposium on Virtual Reality Software and Technology (VRST '20), November 1–4, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3385956.3418966

1 INTRODUCTION

Due to advances in immersive technology, "room-scale" and potentially even "building-scale" virtual reality (VR) experiences are increasingly available to developers and consumers. Along with the numerous benefits these emerging technologies provide, they also present new challenges. One of the fundamental problems that VR researchers and designers must solve is locomotion: movement through the virtual world. Specifically, how can the user navigate the virtual world while maximizing the available physical space?

Providing a VR experience that leverages physical interaction is one of the most compelling uses of the physical space. Research has shown this can significantly enhance the user's experience [10, 11]. These physical interactions require that the relationship, or mapping, between the user's virtual and physical pose remains constant. A constant mapping will always have the same offset between the user's virtual and physical pose. This necessitates that the mapping cannot be modified by the user's interactions with the virtual world (e.g. translating, rotating, interfacing). A variable mapping, the converse of a constant mapping, allows for

transformations between the user's physical and virtual poses; these transformations stem from user interaction or system manipulation.

Most locomotion techniques for navigation in VR require variable mappings, including common strategies such as teleportation and flying. With these techniques, the user translates and rotates their virtual pose without altering their physical pose. This introduces a variable offset between the physical and virtual poses. Conversely, locomotion via walking requires physical movement; this leads to virtual motion that is traditionally equal in magnitude and direction to the physical motion. This maintains a constant mapping between the physical and virtual pose. However, the desire to employ physical walking has led to the development of locomotion techniques such as redirected walking (RDW) that break this constant physical-virtual mapping. RDW alters the user's mapping on a frame-per-frame basis in a unpredictable way to maximize the physical space. VR experiences that employ RDW are unable to leverage interactions with the physical environment, which restricts the kind of immersive experiences available to the user. To address this deficiency, we draw on the field of robotics and artificial potential functions to propose a novel method for aligning the virtual and physical environment using RDW.

At the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces, three papers were presented that collectively represent a new computational framework for RDW: the Push/Pull Reactive (P2R) algorithm [30] and Artificial Potential Field RDW (APF-RDW) [5, 20]. Although they differ in implementation details, both algorithms employ artificial potential functions, a concept adapted from the field of robotics [12, 13, 17]. These potential functions are artificial as they are modeling an abstract potential energy rather than an actual physical phenomenon. The domain of an artificial potential function is comprised of the set of all possible system states. Generally, system states that are considered to be more ideal have a lower associated energy, while less ideal system states have a higher associated energy. In this case, the goal of the system is to be in a state that has the lowest corresponding energy. While artificial potential functions typically have an attractive component and a repulsive component, they are modified for RDW to only feature the repulsive component. The P2R and APF-RDW algorithms calculate the energy at a location within the physical environment through the euclidean distance from the point to obstacles and boundaries. Thus, the minima of the artificial potential function are the physical environment locations that are the furthest away from obstacles and boundaries. By calculating the negative gradient of the potential function at the user's position, these algorithms can choose RDW gains to steer the user in the most ideal direction.

This work is an in-depth evaluation of ideas and concepts that expand upon preliminary results that we presented at a recent workshop [31]. We investigate the attractive force component of artificial potential functions to find system configurations that support interaction with the physical environment, and we define this process as **environmental alignment**. To the best of our knowledge, previous research in computational approaches for RDW have exclusively focused on avoidance of physical obstacles [20, 29, 30] or collisions between multiple users [5]. This project represents a fundamentally new direction that can address one of the major usability limitations of current VR applications. The major contributions of this paper include:

- The introduction of *alignment*, a novel use of RDW that has the potential to increase usability and interactivity in VR applications.
- A mathematical framework for implementing *environmental alignment*, a class of alignment that addresses a prominent problem with VR locomotion techniques.
- The extension of an existing RDW algorithm to support environmental alignment.
- An experiment evaluating the effects of manipulating reactive environmental alignment implementation variables.
- An experiment evaluating the capability of reactive environmental alignment to reverse mapping offsets introduced by conventional RDW algorithms.
- An empirically validated method for generalizing alignment in order to optimize its performance in a reactive implementation.

These contributions lay the foundation for future research into more advanced algorithms for alignment, including predictive methods.

2 BACKGROUND AND RELATED WORKS

Real walking as a locomotion technique provides several benefits over other techniques including improved navigability [26] and sense of presence [32]. RDW maintains the benefits of real walking, but also introduces subtle manipulations to the user's movement to better use the available physical space. It works by slowly and continuously amplifying or diminishing a component of the user's movement in the virtual environment, and is most commonly implemented using a combination of three self-motion illusions [25]. *Translation gain* techniques measure changes in tracked head position and scale the user's virtual movement in the forward direction, enabling travel over smaller or greater distances in the virtual world. *Rotation gain* techniques measure the change in tracked head orientation and scale the corresponding virtual rotation to reorient the user towards a target location, usually away from physical obstacles and boundaries. *Curvature gain* techniques work by adding small rotations to the user's point of view as they translate forward. Users will subsequently compensate for the offset by walking along a circular arc in the opposite direction of the added rotations. Human sensitivity to self-motion illusions can be measured empirically and RDW is often implemented using the average detection thresholds calculated by Steinicke et al. [27]. In most cases, the user will inevitably enter a collision course with a boundary or obstacle, at which point the system will introduce a reorientation event, or a reset [33]. Resets pause the user's experience and reorient them to a physical direction that is favorable for the RDW system. These events are disrupting to the user's experience, thus it is advantageous to minimize them. For this reason, the number of resets is often a metric used when evaluating RDW systems.

An extensive volume of literature on RDW has developed over the last 15 years; a recent community-authored review can be found in Nilsson et al. [23]. Most research efforts focus on developing and evaluating RDW algorithms. These algorithms choose which gains to apply when, and to what degree. Generally, these algorithms are considered to be either reactive or predictive, with predictive algorithms being further categorized as static or dynamic.

2.1 Reactive Algorithms

Reactive algorithms, such as Steer-to-Center (S2C) and Steer-to-Orbit (S2O) [24], are the simplest RDW algorithms. They have no knowledge of the user's intended trajectory and react to the current system state in order to provide local optimization. Typically, these algorithms work on a single heuristic. S2C always applies gains that attempt to steer the user toward the center of the physical environment, while S2O steers the user along a predefined orbit around the center of the physical environment. Hodgson et al. showed that in most scenarios, S2C outperforms other reactive algorithms [9], however, they posit that S2O might outperform S2C if the virtual path is long and consists of very few turns. Azmandian et al. further compared reactive algorithms in a variety physical environment sizes and aspect ratios [2]. The results of that work reinforce those found by Hodgson et al., showing that S2C outperforms the other reactive algorithms in most practical use cases. However, a new class of reactive algorithms use artificial potential functions rather than a heuristic, and they exhibit more complex behaviors. Thomas et al. and Messinger et al. showed that this class of algorithms performs better than S2C whenever the environment is non-convex or contains obstacles [20, 30], and Bachmann et al. showed that they can be used to effectively implement multi-user RDW [5].

2.2 Predictive Algorithms

Predictive algorithms have some knowledge regarding the user's future movements, and can plan for them accordingly. Unlike reactive algorithms, which optimize the instantaneous state of the system, predictive algorithms can greatly reduce the number of resets. This is done by selecting gains that optimizes for a known future trajectory. In his dissertation, Azmandian broke predictive algorithms into two further categories: static planning and dynamic planning [1]. This work proposes that the simplest way to obtain knowledge about the user's future movement is to have the user follow a pre-defined virtual path. If it is safe to assume that the user will follow the predefined path without deviation, then static planning algorithms can be used. Azmandian provided a static planning algorithm called COPPER; in some scenarios with this algorithm, the user would not encounter a single reset. Static planning algorithms greatly out-perform all other RDW algorithms when they can be employed [1]. For situations where static planning algorithms are not viable, as in free-exploration experiences, dynamic planning algorithms are the ideal alternative. Dynamic planning algorithms, such as FORCE [36] and MPCRed [22], work by attempting to predict the most likely virtual path the user will take and selecting gains to optimize it. Currently, this class of algorithms is limited in its application. They only work reliably when the number of direction options facing a user is fairly low, such as in a maze or a similar corridor system.

2.3 Physical Interactions

There is an extensive body of research on interacting with physical objects in VR. One key interaction technique is passive haptics, where a generic physical object is mapped to a specific virtual object [18]. When the user interacts with the virtual object, they also interact with the physical object. Prior research shows that physical interaction during a virtual experience can enhance the user's sense of presence [6]; furthermore, if the physical object closely resembles the virtual object, the user's sense of presence will also improve [10, 11]. As previously stated, RDW generally eliminates planned physical interactions in a VR experience. However, with careful planning and custom implementation of RDW gains, it is possible to overcome this limitation. Kohli et al. described a method combining the experience narrative with carefully chosen RDW gains to bring the user from one virtual pedestal to another, while physically returning to the same pedestal [14]. The pedestal, which was cylindrical in shape, was chosen for its rotational invariance; the user could approach it from any angle and it would still be rotationally aligned with the virtual pedestal. Steinicke et al. created a scenario where the virtual environment was a larger version of the physical environment [28]. The environment in both spaces was square, and consisted of a single cube-shaped obstacle at its center Because of this unique symmetry, RDW gains could make the virtual environment fit within the physical environment, and the single square obstacle remained aligned to the virtual square object. Unfortunately, the difficulties posed by realigning a user within a physical space are exponentially more difficult if the number of users is more than one. This shortcoming prevents two or more users from physically interacting with each other, even if they are in a shared virtual and physical space. Min et al. proposed a multi-step process to address this problem, where users are instructed to move in ways that will eventually allow them to physically interact [21].

3 ALIGNMENT OF VIRTUAL AND PHYSICAL SPACES

We define **alignment** as the process transforming an arbitrary mapping to a desired mapping by employing RDW techniques. Alignment itself does not provide a constant mapping, but rather attempts to guarantee a specific mapping when certain conditions within the VR experience are met. The set of conditions explored in this work result in **environmental alignment**; this attempts to guarantee that for some pre-defined region of the virtual environment, the user can interact with the physical environment. This region is defined in reference to the virtual environment, as the virtual environment drives the user's experience and decisions.

3.1 Mathematical Foundations

Artificial potential functions are an ideal choice for implementing alignment through RDW algorithms, as the resulting algorithm can leverage an attractive force component. The mathematical framework provided for the Push/Pull Reactive algorithm (P2R) that we presented in an earlier paper [30] includes both an attractive and repulsive component for the artificial potential function. However, in the previous work only the repulsive component is used to keep the user away from boundaries and obstacles. For this work, we define this strategy as **avoidance redirection**. To achieve alignment between the user's physical and virtual mapping, we extended P2R and employed the previously unused attractive component.

Eq. 1 shows the potential function used by P2R for a given set of obstacles, O. This function calculates the associated potential energy at a point, q, within the physical environment.

$$U(q) = \frac{1}{2}||q - q_{goal}|| + \sum_{ob \in O} \frac{1}{||q - q_{ob}||} \quad (1)$$

Eq. 1 can be broken into two components: an attractive force and a repulsive force, shown respectively in Eqs. 2 and 3. The repulsive component will create an artificial force in the direction away from boundaries and obstacles. The attractive component will create an artificial force in the direction towards a goal location.

$$U_{attractive}(q) = \frac{1}{2}||q - q_{goal}|| \quad (2)$$

$$U_{repulsive}(q) = \sum_{ob \in O} \frac{1}{||q - q_{ob}||} \quad (3)$$

For each frame, P2R uses the centered finite difference method to calculate the negative gradient, $-\nabla U(q)$, of the potential function at the user's position. This negative gradient is then used to determine the ideal steering direction and compute RDW gains. In this work, we do not modify how P2R selects RDW gains, and the exact details can be found in the original literature [30].

3.1.1 Configuration Spaces. The domain of the potential function presented in Eq. 1 consists of the physical environment's set of Cartesian coordinates. This is sufficient when the system is only steering the user away from physical obstacles, but alignment requires the system to have awareness of both the physical and virtual environments. To accommodate this, we extended the domain of the potential function to a more general configuration space.

A configuration space, as used in robotics research, is a higher dimensional space where each dimension represents one degree of freedom [19]. All possible configurations of a robot are represented in the configuration space as a single point, known as the configuration. Configuration spaces are an advantageous tool when used for motion planning. Given a starting configuration and a goal configuration, any multitude of path planning algorithms can subsequently find a viable path through the configuration space. This path through the configuration space can then be used to manipulate the robot's actuators until it has reached its goal configuration.

For RDW algorithms, configuration spaces can comprehensively represent the state of the entire system. The user's physical position and orientation, and their virtual position and orientation can be *uniquely* represented as a single point. Paths through such a configuration space can alter the user's physical position and orientation at different rates relative to their virtual position and orientation. This difference can be used to calculate which RDW walking gains to apply, and the levels required to move the user along the most optimal path through the configuration space.

3.1.2 Utility Functions. The potential function P2R originally employed, Eq. 1, assumes that a single goal and a single set of obstacles are being used. This limits both the number and the richness of potential user interactions. To overcome this, we propose generalizing Eq. 1 using utility functions. A utility function takes the form:

$$u(q) = A||q - q_u||^B \quad (4)$$

Here, A and B are variables that are selected to define what the utility function does, and q_u is the point in some associated region of the configuration space that is closest to q. It is possible that the associated region may only be a single point. The new potential function is simply the sum of all the utility functions being used.

$$U(q) = \sum_{i=1}^{n} u_i(q) \quad (5)$$

Notably, from the original potential function in Eq. 1, the attractive component in Eq. 2 and repulsive component in Eq. 3 can both be rewritten using utility functions. For the attractive component, let A be $\frac{1}{2}$, B be 1, and q_u be q_{goal}. For the repulsive component, create a utility function for each obstacle and let A be 1, B be -1, and q_u be q_{ob}. In general, A determines the prevalence of the utility function and B determines if the utility function is attractive ($B > 0$) or repulsive ($B < 0$). In this work we define utility functions to accomplish both avoidance and alignment redirection, though this generalization lends itself to other complex RDW based interactions. Exploring more of these interactions further would prove to be a promising area of future research.

3.2 Alignment Algorithm

A set of alignment conditions and the appropriate utility function were developed to add alignment capabilities to P2R. The set of conditions for environmental alignment is straightforward: when a user's virtual pose is within a pre-defined region of the virtual environment, their mapping should be constant and equal to the offset between the physical and virtual origins. We define such a mapping as an **identity mapping**. Transforming from an arbitrary mapping to an identity mapping can be accomplished by adding an attractive utility function operating on the configuration space C.

$$C = \{q \in \mathbb{R}^6 \mid q = \{x_p, y_p, \theta_p, x_v, y_v, \theta_v\}\} \quad (6)$$

Here, x_p, y_p, θ_p represent the a physical position and heading, and x_v, y_v, θ_v represent a virtual position and heading. Within this configuration space, a region C_a is chosen such that it represents all possible configurations that fulfill our alignment conditions.

It is often useful to talk about parts of this configuration space separately. In this work we will often refer to the virtual user or the physical user. This is a convenient way to reference the virtual pose and physical pose of the user's configuration, respectively. Similarly, we refer to the goal configuration as the alignment target, and the virtual pose and physical pose components as the virtual alignment target and physical alignment target, respectively.

$$\begin{aligned} C_a = \{q \in C \mid \alpha_x \le x_v \le \beta_x \\ \wedge \alpha_y \le y_v \le \beta_y \\ \wedge \{x_p, y_p, \theta_p\} - \{x_v, y_v, \theta_v\} = \{x_o, y_o, \theta_o\}\} \end{aligned} \quad (7)$$

In Equation 7, α and β define a rectangular region within the virtual environment where an identity mapping is desired. α_x and α_y represent the lower boundaries for x_v and y_v respectively, and β_x and β_y represent the upper boundaries for x_v and y_v respectively. $\{x_o, y_o, \theta_o\}$ represent the offset between the physical and virtual origins. If $\alpha_x = \beta_x$ and $\alpha_y = \beta_y$, then the alignment region is a single point. This equation illustrates that if the user is within the virtual region defined by α and β, and the offset between their virtual and physical poses is equal to the offset between the virtual and physical origins, then they are aligned.

Eq. 8 shows the resulting utility function, where q_a is the point within C_a that is closest to the user's configuration. As the utility function needs to be attractive, B_a needs to be greater than 0. We call this utility function the alignment utility function. This function is used in both Experiments 1 and 2. A value of 2 was selected for B_a following informal evaluations, but future research is necessary to determine the exact effect of B_a.

$$u_a(q) = A_a||q - q_a||^{B_a} \tag{8}$$

Eq. 9 shows the avoidance utility function that steers the user to avoid boundaries and obstacles. q_o is the configuration closest to the obstacle region C_o, where C_o contains every configuration resulting in a boundary or obstacle collision. A_o was chosen to be 1 and B_o was chosen to be -1, consistent with the literature [30].

$$u_o(q) = A_o||q - q_o||^{B_o} \tag{9}$$

4 EXPERIMENTS

4.1 Simulation Framework

The experiments reported in this paper were conducted using simulation. A large number of trials testing numerous possible parameters is necessary to comprehensively evaluate the performance of RDW algorithms. This can make live user studies prohibitive and impractical. Consequently, simulation-based evaluation is a common and valid practice in RDW research (e.g. [2, 3, 9, 30, 36]).

The simulations were run on a Dell PowerEdge R815 with 4x AMD Opteron 6220 processor and 192GB of RAM. All simulations were run with a fixed frame rate of 90 fps. Each permutation consisted of 100 trials, and at the start of a trial the simulated user would turn to face the first waypoint, then walk directly towards it. Upon reaching a waypoint the simulated user would stop, turn to face the next waypoint, and again walk directly towards it. This continued until the simulated user reached the final waypoint. The simulated user turned at a constant rate of $\frac{\pi}{2}$ radians per second and translated at a constant speed of 1 meter per second. The physical component of the simulated user would be redirected using the modified P2R algorithm. The simulation parameters were chosen to be the same as previous simulation based RDW studies [2, 30].

Translation and rotation gains were limited to the detection thresholds determined by Steinicke et al. [27]. The maximum curvature was set to a radius of 7.5m, which is a commonly employed threshold value [2, 9]. The simulated physical environment consisted of a 10m x 10m square environment with no obstacles. A reset was triggered upon intersection with one of the boundaries and the simulated user's virtual representation would complete a full rotation while their physical representation would rotate to face the center of the physical environment.

Most RDW algorithms dynamically modify the gain values on a frame-per-frame basis. In a recently published perceptual study, Cogdon et al. explored human sensitivity to the rate of change of rotation gains and suggested that slow changes are harder to detect than sudden ones [7]. However, there is still a lack of understanding for how gain rate change should be modulated by a RDW system, and how specific implementations of gain smoothing may interact with other algorithm parameters. Therefore, the simulations in this paper did not apply temporal smoothing to rotation, translation, or curvature gains. The experiments compared relative performance of different alignment strategies under a consistent set of conditions, and therefore provide generally applicable insight into advantages and disadvantages of these methods. However, we do plan to investigate modulation of dynamic gain changes in the context of redirection and alignment in future work.

4.2 Experiment 1

By definition, reactive algorithms do not make any attempt to predict users' future movements. The proposed approach therefore assumes that the system is continuously seeking an aligned state, and therefore applies avoidance and alignment redirection simultaneously. The combination of avoidance and alignment redirection have not been previously investigated, and therefore the interactions between the variable parameters described in Section 3 are not well understood. This limits our ability to optimize the alignment process and understand the effects of reversing a mapping. To address these gaps in knowledge, Experiment 1 describes an investigation into two of the variables in the alignment algorithm: the weight parameter A_a and the prioritization of the components of the negative gradient. These specific parameters were selected because we expected them to have a particularly strong influence on the interaction between avoidance and alignment redirection.

The experiment used a 2x2 design to examine the impacts of these parameters. For each trial, the simulated user would start at the center of the physical environment with their heading parallel to the positive y-axis. They would then navigate a virtual path of 20 waypoints (this number was chosen as to give enough distance for the effects of RDW to take place given the physical environment size). Each waypoint was generated at a random distance from the previous waypoint using a uniform distribution between 2 and 6 meters. Likewise, the rotation of a new waypoint was generated at a random from the rotation of previous waypoint, using a uniform distribution between $-\pi$ and π radians [2, 30]. The virtual alignment target was located at the final waypoint, oriented in the direction that the simulated user would be facing when walking from the penultimate waypoint. Each permutation used the same ordered set of 100 virtual paths and alignment targets [30].

The utility function shown in Eq. 8 has a weight parameter, A_a, and this parameter increases or decreases the amount of influence the alignment utility function has over other utility functions. The first independent variable was this weight parameter, with two possible conditions: static weighting (SW), or dynamic weighting (DW). DW uses an A_a value of $\frac{1}{d_v}$, where d_v is the distance between the virtual alignment target and the virtual user, and SW uses an A_a value of 1. Both avoidance redirection and alignment redirection were applied at the same time. Therefore, the alignment weight parameter will determine how the alignment utility function interacts with the avoidance utility function.

The second independent variable is the prioritization of the negative gradient's positional components versus its rotational components. Two conditions were considered: position priority (PP) and orientation priority (OP). Aligning a user's mapping requires the user to reach a specific position and orientation, both physically and virtually. An initial solution is to use a potential function to attract the user to the goal position and orientation simultaneously.

$$E_p = \sqrt{(x_{p-user} - x_{p-goal})^2 + (y_{p-user} - y_{p-goal})^2 + (x_{v-user} - x_{v-goal})^2 + (y_{v-user} - y_{v-goal})^2} \quad (10)$$

However, cases can arise where the system may seek to alter the user's location in such a way that repositioning requires a rotation in one direction, while reorienting requires a rotation in the opposite direction. As the user cannot rotate in both directions simultaneously, the system must choose to prioritize the positional steering component, or the rotational steering component.

Experiment 1 measured two dependent variables: the number of resets, and the positional alignment error (E_p). The number of resets provides insight into the effectiveness of the avoidance redirection component as independent variables are changed. The positional alignment error provides insight into the effectiveness of the alignment redirection component. This error, shown in Eq. 10, is defined as the Euclidean distance between the positional components of the user's final configuration and the alignment target. For simplicity, only positional alignment error was considered. This is justified by the fact that once the user is positionally aligned, a simple reset can align their orientation with the desired configuration.

For Experiment 1, we made the following hypotheses:

- **H1**: DW conditions will have a fewer number of resets compared to the SW conditions. Dynamically weighting the alignment utility function will allow the avoidance redirection component to have a greater effect for a greater amount time, which should result in fewer resets.
- **H2**: DW conditions will have greater positional alignment error. Dynamically weighting the alignment utility function will reduce the effect and length of the alignment redirection component, resulting in greater positional alignment error.
- **H3**: OP conditions will have greater positional alignment error compared to PP conditions. This follows from the fact that giving position priority over orientation should result in less positional error.

4.2.1 Results. A Kolmogorov-Smirnov test for normality was conducted for both dependent variables and they were not found to be normally distributed. Because of this, results for Experiment 1 were analyzed using non-parametric techniques and the reported values are medians (Mdn) and inter-quartile ranges (IQR). Results for Experiment 1 were analyzed using a Kruskal-Wallis H-test and the Mann-Whitney U test was used for post-hoc multiple comparison tests. A significance value $\alpha = 0.05$ was used for all tests. P-values reported for post-hoc multiple comparison tests were adjusted using the Bonferonni method.

Analysis of the number of resets found a significant difference $H(5) = 227.13$, $p < 0.001$, $\eta^2 = 0.374$. Post-hoc analysis found significantly fewer resets were encountered with the DW-PP permutation ($Mdn = 5$, $IQR = 2$) than the DW-OP permutation ($Mdn = 9$, $IQR = 2$, $U = 356.0$, $p < 0.001$), the SW-PP permutation ($Mdn = 6$, $IQR = 3$, $U = 3115.0$, $p < 0.001$), and the SW-OP permutation ($Mdn = 9$, $IQR = 3$, $U = 332.0$, $p < 0.001$). Significantly fewer resets were also encountered with the SW-PP permutation ($Mdn = 6$, $IQR = 3$) than the SW-OP permutation ($Mdn = 9$, $IQR = 3$, $U = 1190.0$, $p < 0.001$) and the DW-OP permutation ($Mdn = 9$, $IQR = 2$, $U = 1274.5$, $p < 0.001$).

Analysis of the positional alignment error found a significant difference $H(5) = 24.26$, $p < 0.001$, $\eta^2 = 0.032$. Post-hoc analysis found significantly less error was encountered with the SW-PP permutation ($Mdn = 3.01$, $IQR = 3.09$) than the SW-OP permutation ($Mdn = 4.87$, $IQR = 3.96$, $U = 3370.0$, $p < 0.001$), and the DW-OP permutation ($Mdn = 4.64$, $IQR = 3.65$, $U = 3296.0$, $p < 0.001$). Significantly less error was also encountered with the DW-PP condition ($Mdn = 3.94$, $IQR = 2.74$) than the DW-OP condition ($Mdn = 4.64$, $IQR = 3.65$, $U = 4009.0$, $p = 0.047$).

4.2.2 Discussion. Consistent with H3, Experiment 1 showed that orientation priority performed worse than position priority for both the number of resets and positional alignment error. Additionally, under position priority, H1 and H2 were shown to be correct. Dynamically weighting A_a resulted in fewer resets, but greater positional alignment error. Finally, position priority resulted in significantly lower positional alignment error, confirming H3, but it also resulted in significantly fewer resets, which we did not predict.

In general, our results indicate that alignment using artificial potential functions can reduce the discrepancy of a position between virtual and physical space. However, it is important to note that Experiment 1 was set up to evaluate the relative performance differences between variations in the implementation of alignment. As such, the simulations were not constructed with the intention of reducing the positional alignment error to zero. Because the algorithm was purely reactive and the waypoints and alignment targets were generated randomly, it is unreasonable to expect positional alignment error to disappear. However, in prior research, predictive approaches [22, 36] or custom-built narrative scenarios [35] can significantly improve the effectiveness of RDW algorithms. These results can inform the implementation of alignment within RDW systems that are more sophisticated and complex. This points to the need for further research that can build upon this initial exploration of reactive alignment in a variety of contexts.

4.3 Experiment 2

In Experiment 1, alignment redirection and avoidance redirection were applied simultaneously. As expected, purely reactive approaches leave much room for improvement because they are naive to the user's goals. However, if we relax the restrictions of reactive RDW by providing the system with some knowledge of when alignment is necessary, it would be possible to switch between avoidance and alignment redirection as needed. This capability introduces several interesting use cases. For instance, an experience could begin with a long virtual path that ends at a virtual room with several objects that have physical haptic proxies. Locomotion along the long virtual path would best be implemented with avoidance redirection techniques. However, by the time the user arrives at the room with physical interactions, their mapping would be disrupted. We could then apply alignment-only redirection in order to essentially reverse the mapping discrepancies introduced by RDW and return the system to an aligned state. Experiment 2, therefore, explored the notion of reversing avoidance redirection.

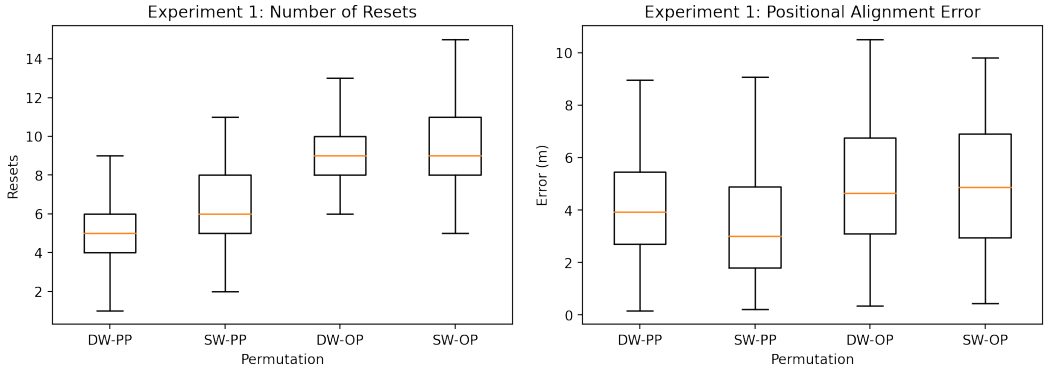

Figure 2: Experiment 1 number of resets (left) and positional alignment error, E_p (right). The bar represents the median value, the box represents the IQR, and the whiskers represent the data spread not including outliers

Using a 4x2 experimental design, a simulated user walked d virtual meters in a random direction, turned around, and walked back to their starting location. The simulated user's starting physical location was randomly chosen within the physical environment (with a 1m buffer from the boundaries) using a uniform distribution. The alignment utility function was defined such that the simulated user's starting configuration was the singular alignment target configuration. Specifically, the simulated user started in an aligned state, and the goal was for the user to return to the same physical and virtual locations at the end of the trial. To help describe the experimental design, we define the point where the virtual user turned around as the *inflection waypoint*. Informed by the results from Experiment 1, position priority was used, and as avoidance redirection and alignment redirection were not being applied at the same time, the weighting had no effect.

The first independent variable considered was the application of alignment after the inflection waypoint. There were two conditions: alignment, or no alignment. In both conditions, only avoidance redirection was applied until the virtual user reached the inflection waypoint. In the alignment condition, the system would switch to only applying alignment redirection upon the virtual user reaching the inflection waypoint. In the no alignment condition, the system would continue to only apply avoidance redirection.

The second independent variable was the virtual distance d between the simulated user and the inflection waypoint. Four possible conditions were considered: 5m, 10m, 20m, and 30m. This condition explores the impact navigated distance has on alignment effectiveness during the application of alignment redirection.

As in Experiment 1, Experiment 2 considered two dependent variables: the number of resets and the positional alignment error (E_p). Our hypotheses for Experiment 2 were as follows:

- **H1**: Conditions with alignment will result in lower positional mapping error than conditions with no alignment. Theoretically, trials with the no-alignment conditions should finish with the simulated user's physical position pseudo-randomly distributed around the physical environment. In trials with alignment, the simulated user is being redirected toward the goal configuration, and theoretically should end closer to the goal configuration.
- **H2**: Conditions with a longer virtual path will result in lower positional mapping error than conditions with a shorter virtual path. This is based on the supposition that the larger the distance the user has to translate over, the more time the system has to move the user into an aligned state.
- **H3**: Conditions with alignment will result in a higher number of resets, as the mode of redirection changes from avoiding boundaries to pursuing a goal position.

4.3.1 Results. A Kolmogorov-Smirnov test for normality was conducted for both dependent variables; they were not found to be normally distributed. Therefore, results for Experiment 2 were analyzed using non-parametric techniques. The reported values are medians (Mdn) and interquartile ranges (IQR). Inflection point distance was not analyzed as a confounding factor. The four conditions were analyzed separately. The Mann-Whitney U test was used for pair-wise testing and a significance value $\alpha = 0.05$ was used.

For the 5m inflection waypoint distance, no significant difference was found regarding final positional mapping error between the alignment condition ($Mdn = 1.44$, $IQR = 0.71$) and the no alignment condition ($Mdn = 1.65$, $IQR = 1.25$). A significant difference was found regarding the number of resets between the alignment condition ($Mdn = 0$, $IQR = 2$) and the no alignment condition ($Mdn = 0$, $IQR = 1$) $U = 4247$, $p = 0.015$.

For the 10m inflection waypoint distance, a significant difference was found regarding final positional mapping error between the alignment condition ($Mdn = 0.23$, $IQR = 1.27$) and the no alignment condition ($Mdn = 2.58$, $IQR = 2.60$) $U = 1469$, $p < 0.001$. No significant difference was found regarding the number of resets between the alignment condition ($Mdn = 2$, $IQR = 2$) and the no alignment condition ($Mdn = 2$, $IQR = 2$).

For the 20m inflection waypoint distance, a significant difference was found regarding final positional mapping error between the alignment condition ($Mdn = 1.67$, $IQR = 2.15$) and the no alignment condition ($Mdn = 2.66$, $IQR = 2.10$) $U = 3185$, $p < 0.001$. No significant difference was found regarding the number of resets between the alignment condition ($Mdn = 3$, $IQR = 1$) and the no alignment condition ($Mdn = 2$, $IQR = 2$).

For the 30m inflection waypoint distance, no significant difference was found regarding final positional mapping error between

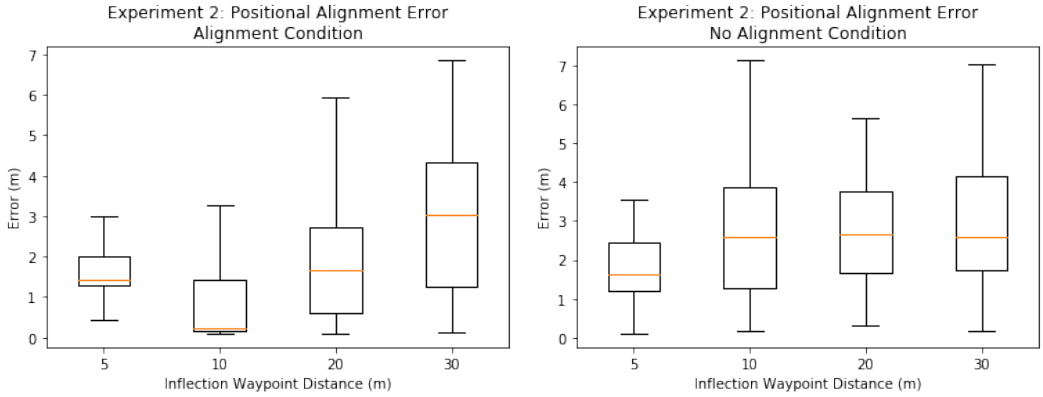

Figure 3: Experiment 2 positional alignment error, E_p, for the alignment condition (top left) and the no alignment condition (top right). The bar represents the median value, the box represents the IQR, and the whiskers represent the data spread not including outliers

the alignment condition ($Mdn = 3.03$, $IQR = 3.06$) and the no alignment condition ($Mdn = 2.61$, $IQR = 2.42$). A significant difference was found regarding the number of resets between the alignment condition ($Mdn = 5$, $IQR = 1$) and the no alignment condition ($Mdn = 4$, $IQR = 0$) $U = 2828$, $p < 0.001$.

4.3.2 Discussion. For Experiment 2, in terms of final positional mapping error, the alignment condition performed significantly better than the no alignment condition for the 10m and 20m inflection waypoint distances. However, no significant results were found for the 5m and 30m inflection waypoint distances.

Regarding the number of resets, the alignment condition performed significantly worse than the no alignment condition for the 5m and 30m inflection waypoint distances, but no significant results were found for the 10m and 20m inflection waypoint distances. This suggests there is an upper and lower inflection waypoint distance threshold during which alignment-only redirection not only ceases to be beneficial, but can actually be harmful. These results were consistent with our hypotheses regarding inflection waypoint distances of 10m and 20m, but were inconsistent with our hypotheses regarding inflection waypoint distances of 5m and 30m.

4.4 Generalizing Alignment

The initial results of Experiment 2 could suggest the existence of some virtual distance between the user and the alignment target where it is ideal to start applying alignment. However, the effects could also be the result of more complex interactions between multiple parameters. While the virtual distance between the inflection waypoint and the alignment target had a significant effect on the number of resets and the positional mapping error, it is also possible that these results were influenced by the physical distance and physical rotational offset between the initial and the desired configuration. As an extension of Experiment 2, we sought a more comprehensive explanation for the observed behavior. Based on prior observations, we chose to examine virtual distance, physical distance, and physical rotational offset to explore the relationship between the user's configuration when alignment-only redirection starts and the alignment target. We targeted these variables as the most likely factors influencing the outcome of the system.

4.4.1 Mathematical Foundations. Using virtual distance, physical distance, and physical rotational offset, we derived a set of mathematical requirements for the relationship between the user's configuration and the alignment target configuration. These requirements can be seen in Eq. 11 and Eq. 12, which correspond to translation gain and curvature gain, respectively. Both equations assume that the virtual user is facing and walking towards the virtual alignment target. Successful environmental alignment occurs when these requirements are satisfied and alignment-only redirection is applied.

$$G_t D_p < D_v < g_t D_p \tag{11}$$

Eq. 11 states that D_v (the virtual distance from the user to the alignment target) needs to be larger than D_p (the physical distance from the user to the alignment target) multiplied by G_t (the maximum translation gain) and less than D_p multiplied by g_t (the minimum translation gain). When this requirement is met, there exists a translation gain between the perceptual thresholds that, when applied, will result in the virtual user and the physical user arriving at the respective alignment targets at the same time.

$$\phi_p < sin^{-1}(\frac{D_p \kappa_{max}}{2}) \tag{12}$$

Eq. 12 sets the required relationship between ϕ_p (the physical rotational offset), D_p, and κ_{max} (the maximum curvature). When this requirement is met, the physical user will be facing a direction such that the application of curvature gain will result in the user moving straight towards the physical alignment target. This will necessarily occur before the virtual representation of the user arrives at the virtual alignment target.

4.4.2 Validation. A simulation-based evaluation was used to validate the mathematical relationships described in the previous section. For each trial, the starting and alignment target configurations were chosen at random such that the virtual user was facing the virtual alignment target, and that their relationship satisfied Eqs.

11 and 12. The simulation conditions were the same as in Experiment 2, except that the user would immediately progress to the virtual alignment target and alignment-only redirection would be applied. 1000 such trials were conducted with the following final positional errors (presented as quartiles): $Q1 = 0.09m$, $Q2 = 0.10m$, $Q3 = 0.12m$. The low alignment error, combined with the narrow inter-quartile range (IQR), provides empirical support for the proposed equations.

4.4.3 Alignment Regions. Applying avoidance and alignment redirection simultaneously to achieve environmental alignment in a purely reactive manner may be inefficient as suggested by the first experiment. However, the second experiment shows that if the system has knowledge of the user's alignment requirements, or can identify when the user needs to align to the physical environment, it can switch from avoidance redirection to alignment redirection. Our results suggest that this is a much more effective strategy. Furthermore, constructing regions in the configuration space such that they satisfy the requirements of Eq. 11 and Eq. 12 can reduce the alignment error for a given goal configuration. When the system identifies that the user is within one of these regions, it can switch to alignment-only redirection to achieve the desired configuration. Additionally, these regions can be used to create virtual waypoints such that once the user reaches that waypoint, their configuration would be located in one of these regions. In summary, we believe that further development and validation of region-based alignment is a promising area for future work.

5 FUTURE WORK

This work presented the mathematical foundations and initial experiments that point towards the value of future research, development, and evaluation of novel alignment methodologies. To this end, we have identified several research vectors to inform future work.

In unconstrained scenarios, we do not expect reactive RDW algorithms to consistently steer the user away from boundaries and obstacles; this is why resets are also employed. When there is positional error remaining after alignment should have completed, a similar "alignment reset" could be applied. For example, rather than pausing an experience to rotate the user towards a more favorable orientation, an alternative intervention would be a *positional reset*, where the user translates to a particular location. Gretchkin et al. implemented a similar concept, called "Rotate and Walk," to achieve contextually relevant resets [8]. Continuous alignment along with positional resets may collectively represent a more generalizable solution, and future work that investigates multiple synergistic techniques would be valuable.

Current RDW techniques cannot steer the user in a specific direction without simultaneously manipulating their rotation. For example, with rotation and curvature gains, there is no way to move users left or right while maintaining a forward-facing orientation. This may impact future avenues for alignment research; particularly those that may try to manipulate the position and rotation of the user independently. You et al. recently introduced a new RDW technique, strafing gain, that has the user strafe to either their left or right as they walk forward [34]. This technique has promise for alignment, as well as traditional RDW, but has yet to be perceptually validated. In general, new redirection techniques that allow for more flexible manipulations of the user's mapping would increase the effectiveness of alignment algorithms.

In practice, it may be difficult to remove alignment error entirely. However, previous research in redirected touching has shown that visual cues will also dominate proprioceptive and haptic feedback [15, 16]. A positional error of approximately 10cm, as shown in some of our experiments, is quite small given the overall scale of the physical environment, and would allow coarse physical interaction with large objects. However, errors of this magnitude may interfere with fine-grained interactions with small or highly detailed physical objects. Potential future work could examine techniques such as haptic retargeting [4] to compensate for alignment error.

The work presented in this paper demonstrates how alignment algorithms can be combined with avoidance algorithms to alter a user's mapping and converge on a specific point. This has significant potential for expanding interaction with the physical environment. However, there are additional locomotion techniques not addressed in this work that do not maintain a constant virtual to physical mapping. For example, multiple applications now combine physical locomotion with virtual locomotion techniques (e.g., controller-based steering, teleportation) to expand the space the user can explore. This is commonly seen in "building-scale" VR, which has emerged as a new experiential avenue following the proliferation of room-scale experiences and HMDs with inside-out tracking. When these application employ physical locomotion, it creates an opening for physical interactions that can enhance the user's sense of presence. However, the use of virtual locomotion techniques will immediately break the mapping between the physical and virtual environment, rendering physical interactions impossible. New locomotion techniques and interfaces could aim to combine continuous alignment during physical movement with subtle positional resets in the virtual environment. These types of advances promise to enhance user experiences by allowing developers to leverage the physical world to a greater degree than is currently possible.

6 CONCLUSION

While RDW provides a tool for navigating virtual environments that exceed the confines of a physical space, its usability has been limited by two major shortcomings: an inability to avoid physical obstacles, and the loss of physical interaction as a consequence of dynamic mapping. The original algorithm and the concept of alignment discussed in this paper provide a solution to both of these problems, particularly relating to environmental alignment.

Our experiments illustrate that the reactive algorithm can use repulsive mechanisms to steer the user away from physical obstacles and attractive techniques to align the user with a given target. The system can reduce the positional discrepancy between the virtual and physical space while minimizing the number of resets that the user experiences. Furthermore, if it has knowledge of when the user needs to align, efficiency can be improved by transitioning between avoidance and alignment redirection. The capability to reverse a dynamic mapping opens up significant new possibilities for virtual environments to employ both natural locomotion and physical interaction techniques, and incorporating both virtual and physical elements in large-scale environments will offer a way to create more complex and sophisticated experiences for users.

REFERENCES

[1] Mahdi Azmandian. 2018. *Design and Evaluation of Adaptive Redirected Walking Systems.* Ph.D. Dissertation. University of Southern California.

[2] Mahdi Azmandian, Timofey Grechkin, Mark T Bolas, and Evan A Suma. 2015. Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance.. In *ICAT-EGVE*. 93–100.

[3] Mahdi Azmandian, Tim Grechkin, and Evan Suma Rosenberg. 2017. An Evaluation of Strategies for Two User Redirected Walking in Shared Physical Spaces. In *IEEE Conference on Virtual Reality and 3D User Interaction*. IEEE.

[4] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. 2016. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *ACM CHI Conference on Human Factors in Computing Systems*. ACM, 1968–1979.

[5] Eric R Bachmann, Eric Hodgson, Cole Hoffbauer, and Justin Messinger. 2019. Multi-User Redirected Walking and Resetting Using Artificial Potential Fields. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 2022–2031.

[6] Lung-Pan Cheng, Thijs Roumen, Hannes Rantzsch, Sven Köhler, Patrick Schmidt, Robert Kovacs, Johannes Jasper, Jonas Kemper, and Patrick Baudisch. 2015. Turkdeck: Physical virtual reality based on people. In *ACM Symposium on User Interface Software & Technology*. ACM, 417–426.

[7] Ben J Congdon and Anthony Steed. 2019. Sensitivity to Rate of Change in Gains Applied by Redirected Walking. In *ACM Symposium on Virtual Reality Software and Technology*. ACM, 3.

[8] Timofey Grechkin, Mahdi Azmandian, Mark Bolas, and Evan Suma. 2015. Towards context-sensitive reorientation for real walking in virtual reality. In *IEEE Conference on Virtual Reality*. IEEE, 185–186.

[9] Eric Hodgson and Eric Bachmann. 2013. Comparing four approaches to generalized redirected walking: Simulation and live user data. *IEEE Transactions on Visualization and Computer Graphics* 19, 4 (2013), 634–643.

[10] Hunter G Hoffman. 1998. Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments. In *IEEE Virtual Reality*. IEEE, 59–63.

[11] Brent Edward Insko, M Meehan, M Whitton, and F Brooks. 2001. *Passive haptics significantly enhances virtual environments.* Ph.D. Dissertation. University of North Carolina at Chapel Hill.

[12] Oussama Khatib. 1986. The potential field approach and operational space formulation in robot control. In *Adaptive and Learning Systems*. Springer, 367–377.

[13] Oussama Khatib. 1986. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*. Springer, 396–404.

[14] Luv Kohli, Eric Burns, Dorian Miller, and Henry Fuchs. 2005. Combining passive haptics with redirected walking. In *Proceedings of the 2005 international conference on Augmented tele-existence*. ACM, 253–254.

[15] Luv Kohli, Mary C Whitton, and Frederick P Brooks. 2012. Redirected Touching: The effect of warping space on task performance. In *IEEE Symposium on 3D User Interfaces*. IEEE, 105–112.

[16] Luv Kohli, Mary C Whitton, and Frederick P Brooks. 2013. Redirected Touching: Training and adaptation in warped virtual spaces. In *IEEE Symposium on 3D User Interfaces*. IEEE, 79–86.

[17] Bruce H Krogh and Timothy J Graettinger. 1985. Maneuverability constraints for supervisory steering control. In *IEEE Conference on Decision and Control*. IEEE, 279–284.

[18] R W Lindeman, J L Sibert, and Hahn J K. 1999. Hand-held windows: towards effective 2D interaction in immersive virtual environments. In *IEEE Conference on Virtual Reality*. 205–212.

[19] Tomas Lozano-Perez. 1990. Spatial planning: A configuration space approach. In *Autonomous robot vehicles*. Springer, 259–271.

[20] Justin Messinger, Eric Hodgson, and Eric R Bachmann. 2019. Effects of Tracking Area Shape and Size on Artificial Potential Field Redirected Walking. In *IEEE Conference on Virtual Reality and 3D User Interfaces*.

[21] Dae-Hong Min, Dong-Yong Lee, Yong-Hun Cho, and In-Kwon Lee. 2020. Shaking Hands in Virtual Space: Recovery in Redirected Walking for Direct Interaction between Two Users. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 164–173.

[22] Thomas Nescher, Ying-Yin Huang, and Andreas Kunz. 2014. Planning redirection techniques for optimal free walking experience using model predictive control. In *IEEE Symposium on 3D User Interfaces*. IEEE, 111–118.

[23] Niels Christian Nilsson, Tabitha Peck, Gerd Bruder, Eri Hodgson, Stefania Serafin, Mary Whitton, Frank Steinicke, and Evan Suma Rosenberg. 2018. 15 Years of Research on Redirected Walking in Immersive Virtual Environments. *IEEE Computer Graphics and Applications* 38, 2 (2018), 44–56.

[24] Sharif Razzaque. 2005. *Redirected walking*. University of North Carolina at Chapel Hill.

[25] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. 2001. Redirected walking. In *EUROGRAPHICS*, Vol. 9. Citeseer, 105–106.

[26] Roy A Ruddle. 2013. The effect of translational and rotational body-based information on navigation. In *Human walking in virtual environments*. Springer, 99–112.

[27] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. 2010. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics* 16, 1 (2010), 17–27.

[28] Frank Steinicke, Gerd Bruder, Timo Ropinski, and Klaus Hinrichs. 2008. Moving towards generally applicable redirected walking. In *Proceedings of the Virtual Reality International Conference (VRIC)*. IEEE Press, 15–24.

[29] Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan McGuire, David Luebke, and Arie Kaufman. 2018. Towards virtual reality infinite walking: dynamic saccadic redirection. *ACM Transactions on Graphics* 37, 4 (2018), 67.

[30] J. Thomas and E. Suma Rosenberg. 2019. A General Reactive Algorithm for Redirected Walking using Artificial Potential Functions. In *IEEE Conference on Virtual Reality and 3D User Interfaces*.

[31] J. Thomas and E. Suma Rosenberg. 2020. 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)*. IEEE, 317–323.

[32] Martin Usoh, Kevin Arthur, Mary C Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P Brooks Jr. 1999. Walking> walking-in-place> flying, in virtual environments. In *ACM SIGGRAPH*. ACM Press/Addison-Wesley Publishing Co., 359–364.

[33] Betsy Williams, Gayathri Narasimham, Bjoern Rump, Timothy P McNamara, Thomas H Carr, John Rieser, and Bobby Bodenheimer. 2007. Exploring large virtual environments with an HMD when physical space is limited. In *ACM Symposium on Applied Perception*. 41–48.

[34] Christopher You, Evan Suma Rosenberg, and Jerald Thomas. 2019. Strafing Gain: A Novel Redirected Walking Technique. In *ACM Symposium on Spatial User Interaction*. ACM, 26.

[35] Run Yu, Zachary Duer, Todd Ogle, Doug A Bowman, Thomas Tucker, David Hicks, Dongsoo Choi, Zach Bush, Huy Ngo, Phat Nguyen, et al. 2018. Experiencing an Invisible World War I Battlefield Through Narrative-Driven Redirected Walking in Virtual Reality. In *IEEE Conference on Virtual Reality and 3D User Interfaces*. IEEE, 313–319.

[36] Michael A Zmuda, Joshua L Wonser, Eric R Bachmann, and Eric Hodgson. 2013. Optimizing constrained-environment redirected walking instructions using search techniques. *IEEE Transactions on Visualization and Computer Graphics* 19, 11 (2013), 1872–1884.