2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)

# Reactive Alignment of Virtual and Physical Environments Using Redirected Walking

Jerald Thomas* Evan Suma Rosenberg†

University of Minnesota

## Abstract

Virtual reality applications may accomplish richer and more immersive experiences by incorporating physical interactions such as passive haptic feedback. These interactions require that the coordinate mapping between the physical and virtual environment remains fixed. However, a static relationship is not maintained by many common locomotion techniques, including redirected walking, resulting in a state of misalignment. In this work, we address this limitation by proposing a novel reactive algorithm that uses redirected walking techniques to transition the system from a misaligned state to an aligned state, thereby enabling the user to interact with the physical environment. Traditionally, redirected walking algorithms primarily optimize for avoiding collisions with the boundaries of the physical space, whereas the proposed method leverages redirection techniques to achieve a desired system configuration. Simulation-based experiments demonstrate an effective use of this strategy when combined with redirected walking using artificial potential functions. In the future, reactive environment alignment can enhance the interactivity of virtual reality applications and inform new research vectors that combine redirected walking and passive haptics.

**Index Terms:** Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality;

## 1 Introduction

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

A particularly compelling use of the physical space is to provide the user with a VR experience that leverages physical interaction. Research has shown this can significantly enhance the user's experience [8, 9]. 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. As a result, it cannot be modified by the user's interactions with the virtual world (e.g. translating, rotating, interfacing). Alternatively, a variable mapping allows for transformations between the user's physical and virtual poses. These could be the result of either user interactions or system manipulations.

Most locomotion techniques for navigation in VR, including common strategies such teleportation and flying, require variable mappings. With these techniques, the user translates and rotates their virtual pose without altering their physical pose. This introduces a variable offset between the two poses. Conversely, if the user navigates by walking, they have to physically move in order to virtually move. Traditionally, this movement in the virtual environment is equal in magnitude and direction to the movement in the physical environment, and because of this, a constant mapping is maintained. However, redirected walking (RDW) is a locomotion technique that does not retain a constant mapping. Rather, RDW changes the user's mapping on a frame-per-frame basis in a unpredictable way to maximize the physical space. This prevents VR experiences that use RDW algorithms from incorporating physical interactions. To address this deficiency, we 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 [23] and Artificial Potential Field RDW (APF-RDW) [4, 15]. Although they differ in some implementation details, the two algorithms both employ artificial potential functions, a concept adapted from the field of robotics [10, 11, 13]. The range of an artificial potential function represents an abstract energy and the domain is comprised of the set of all possible system states. More ideal system states have a lower energy, and less ideal system states have a higher energy. Generally, the goal of the system is to be in a state that has the lowest corresponding energy. The P2R and APF-RDW algorithms calculate the energy at a location within the physical environment by using the euclidean distances from the location to obstacles and boundaries. This results in the minima of the artificial potential function being the physical environment locations that are the furthest away from obstacles and boundaries. By then calculating the negative gradient of the potential function at the user's position, the algorithms can choose RDW gains to steer the user in the most ideal direction. Artificial potential functions typically have an attractive component and a repulsive component, however, the algorithms presented for RDW only featured the repulsive component.

In this work, we investigate the use of the attractive force component of artificial potential functions to achieve system configurations that would support interaction with the physical environment, a process we define as **alignment**. To the best of our knowledge, previous research in computational approaches for RDW have exclusively focused on avoidance of physical obstacles [15, 23] or collisions between multiple users [4]. Therefore, this project represents a fundamentally new direction that can address one of the major usability limitations of current VR applications. Major contributions of this paper include:

- The introduction of alignment, a novel use of RDW that addresses a prominent problem with VR locomotion techniques.
- The extension of an existing RDW algorithm that supports both obstacle avoidance and alignment redirection.
- An experiment that evaluates the effects of manipulating reactive alignment implementation variables.
- An experiment that evaluates the capability of reactive alignment to reverse the mapping offsets introduced when using conventional RDW algorithms.

These contributions lay the foundation for future research into more

*e-mail: thoma891@umn.edu
†e-mail: suma@umn.edu

978-1-7281-6532-5/20/$31.00 ©2020 IEEE
DOI 10.1109/VRW50115.2020.00071

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 [20] and sense of presence [24]. RDW maintains the benefits of real walking, but 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 [19]. *Translation gain* techniques measure changes in tracked head position and scale the user's 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. *Curvature gain* techniques work by adding offsets to real world movements, for example, by slowly rotating the virtual world while walking forward. Users will subsequently compensate for the offset by walking along a circular arc. 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. [21]. 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 [25]. 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.

Over the past 15 years, there has been an extensive volume of literature on RDW; a recent community-authored review can be found in Nilsson et al. [17]. A large amount of research effort focuses on developing and evaluating RDW algorithms. These algorithms choose which gain 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.

**Reactive Algorithms** Reactive algorithms, such as Steer-to-Center (S2C) and Steer-to-Orbit (S2O) [18], 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 steer the user toward the center of the physical environment, while S2O steers the user to a predefined circle around the center of the physical environment. Hodgson et al. showed that in most scenarios, S2C outperforms other reactive algorithms [7], 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 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 using artificial potential functions instead of a heuristic 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 [15, 23], and Bachmann et al. showed that they can be used to effectively implement multi-user RDW [4].

**Predictive Algorithms** Predictive algorithms have some sort of knowledge regarding the user's future movements, and can plan for them accordingly. Unlike reactive algorithms, which optimize for the instantaneous state of the system, predictive algorithms can greatly reduce the number of resets by selecting gains that optimizes for a known future trajectory. In his dissertation, Mahdi Azmandian broke predictive algorithms into two further categories: static planning and dynamic planning [1]. 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 and in some scenarios the user would not encounter a single reset. When they can be applied, static planning algorithms far out-perform all other RDW algorithms [1]. If the system cannot rely on a static planning algorithm, dynamic planning algorithms are the ideal alternative. Dynamic planning algorithms, such as FORCE [28] and MPCRed [16], work by attempting to predict the most likely virtual path the user will take and selecting gains that optimize for it. Currently, this class of algorithms only works when the number of movement direction options of a user is fairly low (e.g. in a maze or an indoor system of corridors).

**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. When the user attempts to interact with the virtual object, they also interact with the physical object. It has been shown that if the physical object closely resembles the virtual object, an enhanced level of presence can be achieved [8, 9]. 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 deficiency. Kohli et al. described a method in which the experience narrative, in conjunction with carefully chosen RDW gains, would bring the user from one virtual pedestal to another while physically returning to the same physical pedestal [12]. The pedestal, which was cylindrical in shape, was chosen due to 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 in which the virtual environment was a larger version of the physical environment, which was square and consisted of a single square obstacle in the center [22]. Thus, RDW gains could make the virtual environment fit within the physical environment, and the single square obstacle, which the users could interact with, remained aligned to the virtual square object.

## 3 Alignment of Virtual and Physical Spaces

We define **alignment** as the process of transforming an arbitrary mapping to a desired mapping using RDW techniques. Alignment does not provide a constant mapping, but rather attempts to guarantee a specific mapping when certain conditions within the VR experience are met. An example set of conditions that we explored in this work result in **environment alignment**, which attempts to guarantee that within 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 because the virtual environment is what is driving the user's experience and decisions.

### 3.1 Mathematical Foundations

Artificial potential function based RDW algorithms are an ideal foundation for implementing alignment as they have an attractive force component that can be leveraged for alignment. The mathematical framework provided for P2R includes both an attractive and repulsive component for the artificial potential function, although the authors stated that only the repulsive component is used to keep the user away from boundaries and obstacles (which we define as avoidance redirection). To implement alignment, we extended P2R and made use of the previously unused attractive component.

Equation 1 shows the potential function used by P2R given the set of obstacles $O$. The value of the potential function at a user's position $q$ is one half the distance to the goal position plus the sum of one over the distance to the obstacle's nearest point for all obstacles.

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

Equation 1 can be broken into two components: an attractive force (Equation 2) and a repulsive force (Equation 3).

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

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

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, $q$. $-\nabla U(q)$ is then used to determine the ideal steering direction and compute RDW gains.

#### 3.1.1 Configuration Spaces

Equation 1 has a domain that consists of the physical environment's Cartesian coordinates. This is sufficient when the system is only steering the user away from physical obstacles, but alignment requires that the system has knowledge of both the physical and virtual environments. To accommodate for this we extended the potential function's domain to work on a more general configuration space.

Configuration spaces are another concept from the field of robotics. The configuration space of a robot is a higher dimensional space where each dimension represents one degree of freedom [14]. Any possible configuration of a robot is represented in the configuration space as a single point, referred to as a configuration. In robotics, configuration spaces are desirable because given a starting configuration and a goal configuration, any multitude of path planning algorithms can be used to generate a viable path through the configuration space.

In the case of RDW, configuration spaces are desirable because they can represent the state of the entire system. For example, the user's physical position and orientation as well as their virtual position and orientation can be uniquely represented as a single point. Paths through such a configuration space could result in changing the user's physical position and orientation at different rates than their virtual position and orientation. We can use this difference to calculate which RDW gains to apply, and at what levels, to get the user to navigate the most optimal path through the configuration space.

#### 3.1.2 Utility Functions

This potential function (Equation 1) assumes that a single goal (which is itself a single point in the configuration space) and a single set of obstacles are being used. This is limiting in both the number and richness of user interactions it can provide. To overcome this we propose generalizing the potential function previously used with 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 an associated region (which itself can be a single point) of the configuration space that is closest to $q$. The new potential function is simply the sum of all the utility functions.

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

It should be pointed out here that the attractive component (Equation 2) and repulsive component (Equation 3) of the original potential function can both be represented using utility functions. To represent the attractive component, select $A$ to be $\frac{1}{2}$, $B$ to be 1, and $q_u$ to be $q_{goal}$. To represent the repulsive component, create a utility function for each obstacle and select $A$ to be 1, $B$ to be $-1$, and $q_u$ to be $q_{ob}$. In general, $A$ is set to determine the prevalence of the utility function and $B$ is set to determine if the utility function is attractive ($B > 0$) or repulsive ($B < 0$).

### 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 environment alignment is fairly straight forward: when a user's virtual pose is located 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, which will operate on the configuration space $C$.

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

The variables $x_p, y_p, \theta_p$ represent the user's physical position and heading, and the variables $x_v, y_v, \theta_v$ represent the user's virtual position and heading. Within this configuration space a region, $C_a$, is defined such that it represents all possible configurations that meet our alignment conditions.

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

The variables $\alpha$ and $\beta$ define a rectangular region within the virtual environment where an identity mapping is desired. $\alpha_x$ and $\alpha_y$ represent the lower boundaries for $x_v$ and $y_v$ respectively, and $\beta_x$ and $\beta_y$ represent the upper boundaries for $x_v$ and $y_v$ respectively. $\{x_o, y_o, \theta_o\}$ represents 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 essentially states that if the user is within the virtual region defined by $\alpha$ and $\beta$, and the offset between their virtual and physical poses is the same as the offset between the virtual and physical origins, then they are aligned.

Equation 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, and will be used in both Experiments 1 and 2. A value of 2 was selected for $B_a$ after informal experimentation, but future research is necessary to determine the exact effect $B_a$ has.

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

Equation 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 $C_o$, which contains every configuration that would result in a boundary or obstacle collision. $A_o$ was chosen to be 1 and $B_o$ was chosen to be $-1$, as those were the values used in the original literature [23].

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

## 4 Experiments

### 4.1 Simulation Framework

The two experiments reported in this paper were conducted using simulation. It is necessary to run a very large number of trials and test numerous possible parameters to comprehensively evaluate the performance of RDW algorithms, which is impractical for live user studies. For this reason, simulation-based evaluation is a common

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

practice in the RDW literature (e.g. [2, 3, 7, 23, 28]). RDW gain thresholds used when evaluating RDW algorithms via simulation do not deviate for those determined in previous literature as being unnoticeable by human users. As such, we can assume that the findings from simulation based experiments will translate to a human user population. The work presented in this paper is modifying an existing RDW algorithm, and though we are using RDW techniques in a new way, we maintain that the same arguments could be made for why simulation is a valid evaluation technique for reactive alignment as we are using gain thresholds within the determined perceptual limits.

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 framerate 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 and 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 would continue 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. Translation and rotation gains were limited to the detection thresholds determined by Steinicke et al. [21]. The maximum curvature was set to a radius of 7.5m, which is a commonly employed threshold value [2, 7]. 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 and 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 [5]. However, there is currently a lack of concrete understanding of how the rate of gain changes 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 were conducted to compare relative performance of different alignment strategies under a consistent set of conditions, and will therefore provide generally applicable insight into their 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

The purpose of Experiment 1 was to explore a couple of the variables that can be manipulated when applying alignment in a reactive manner. Two such variables were tested in a 2x2 experiment design. For each trial, the simulated user would start at the center of the physical environment facing along the positive $y$ axis, and navigate a virtual path consisting of 20 waypoints. Each waypoint was generated at a random distance from the previous waypoint using a uniform distribution between 2 and 6 meters. Similarly, each waypoint was generated at a random rotation from the previous waypoint using a uniform distribution between $-\pi$ and $\pi$ radians. The virtual alignment target was located at the final waypoint, facing the direction the simulated user would be facing when walking from the second to last waypoint. Each permutation used the same ordered set of 100 virtual paths and alignment targets.

The first independent variable was the alignment utility function weight parameter and had two conditions: statically weighted (SW), dynamically weighted (DW). The utility function shown in Equation 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.DW will use a $A_a$ value of $\frac{1}{d_v}$, where $d_v$ is the distance between the virtual component of $C_a$ and the virtual pose of the user, and SW will use a $A_a$ value of 1. Both avoidance redirection and alignment redirection were being applied at the same time, so 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 and had two conditions: 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. The first method that comes to mind is to use a potential function that attracts the user to the goal positions and orientations simultaneously.However, it may be the case that the system wants to alter the user's position in such a way that would result in rotating one direction, and their orientation in such a way that would result in rotating in the other direction. As the user cannot rotate in both directions at once, the system must choose to prioritize the positional steering component, or the rotational steering component.

Experiment 1 had two dependent variables: the number of resets, and the positional alignment error ($E_p$). The number of resets will provide insight into the effectiveness of the avoidance redirection component as independent variables are changed, and the positional alignment error will provide insight into the alignment redirection component. Positional alignment error, shown in Equation 10, is the the Euclidean distance between the positional components of the user's final configuration and the alignment goal configuration. To simplify the study, we chose to focus on positional alignment error and not explore the effects on rotational alignment error. Once the user's mapping is aligned with respect to rotation, a simple reset will align the rotational component of the user's mapping.

For Experiment 1 we had 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. Similar to the reasoning behind H1, dynamically weighting the alignment utility function will force the alignment redirection component to have a lesser effect and for not as long, which should result in greater positional alignment error.
- **H3**: OP conditions will have greater positional alignment error compared to PP conditions.

### 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

Experiment 1 shows some interesting results. First, orientation priority performed worse than position priority for both the number of resets and positional alignment error. Additionally, when considering 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 positional discrepancy between the 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, such an expectation would be unreasonable. However, in the prior literature RDW can achieve greater effectiveness using predictive approaches [16, 28] or custom-built narrative scenarios [27]. In addition to reactive 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 our initial exploration of reactive alignment in a variety of contexts.

### 4.3 Experiment 2

For Experiment 1, alignment redirection and avoidance redirection were both being applied at the same time. However, it is possible to use only avoidance redirection and then switch to only applying alignment redirection when the system would like the user to become aligned. The purpose of Experiment 2 was to explore this concept and determine to what degree reactive alignment can "undo" the mapping discrepancies introduced by using RDW to avoid the physical boundaries. To accomplish this, a 4x2 experimental design was implemented. The task consisted of a simulated user walking $d$ virtual meters in a random direction, turning around, and walking back to their starting location. The simulated user's starting physical location was randomly chosen within the physical environment (with a 1m buffer) using a uniform distribution. We defined the alignment utility function such that the simulated user's starting configuration was the singular alignment goal configuration. This means that 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 was whether or not alignment was applied after the inflection waypoint and had two possible conditions: alignment and no alignment. In both conditions only avoidance redirection was applied until the virtual user reached inflection waypoint. The alignment condition would switch to only applying alignment redirection at the inflection waypoint and the no alignment condition would continue applying only avoidance redirection. The second independent variable was the virtual distance between the simulated user and the inflection waypoint ($d$) and had four possible conditions: 5m, 10m, 20m, and 30m. This condition allows us to explore what effect the amount of distance navigated while employing alignment redirection has on alignment effectiveness.

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

- **H1**: Conditions with alignment would result in lower positional mapping error than the conditions with no alignment. Trials with the no alignment conditions should theoretically finish with the simulated user's physical position pseudorandomly distributed around the physical environment. In trials with alignment, the simulated user is being redirected toward the goal configuration and should theoretically end closer to the goal configuration.
- **H2**: Conditions with a longer virtual path would also result in lower positional mapping error than conditions with a shorter virtual path. The longer the user translates the more time the system has to get the user in an aligned state.
- **H3**: Conditions with alignment would result in a greater number of resets as the 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 and they were not found to be normally distributed. Because of this, results for Experiment 2 were analyzed using non-parametric techniques and the reported values are medians ($Mdn$) and inter-quartile ranges ($IQR$). Inflection point distance was not analyzed as a confounding factor, and 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 ($Mdn = 2.66$, $IQR = 2.10$) $U = 3185$, $p < 0.001$.

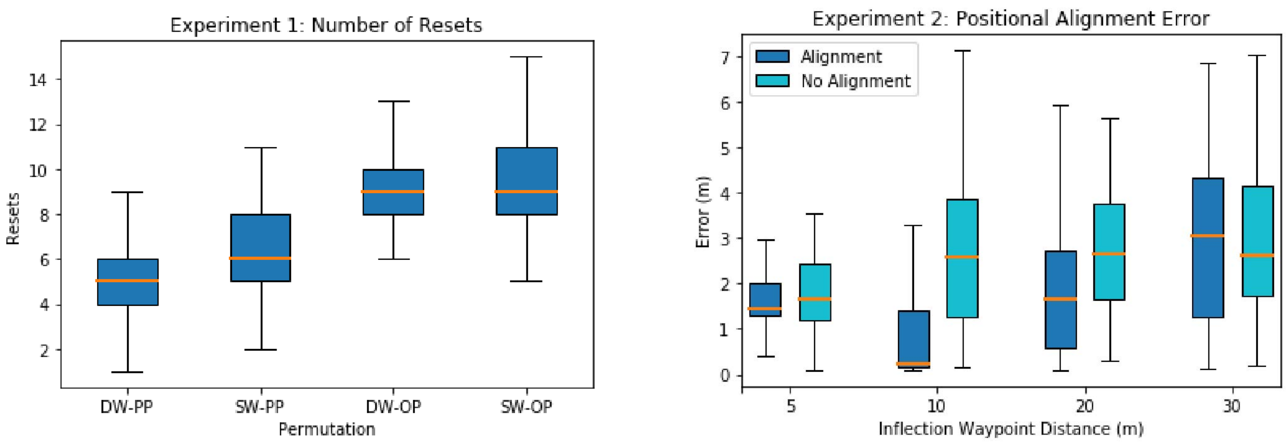


Figure 1: Experiment 1 number of resets (left) and, Experiment 2 positional alignment error, $E_p$ (right). The bar represents the median value, the box represents the IQR, and the whiskers represent the data spread not including outliers

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 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, the alignment condition performed significantly better than the no alignment condition regarding final positional mapping error for the 10m and 20m inflection waypoint distances, but no significant results were found for the 5m and 30m inflection waypoint distances. Conversely, the alignment condition performed significantly worse than the no alignment condition regarding the number of resets for the 5m and 30m inflection waypoint distances, but no significant results were found for the 10m and 20m inflection waypoint distances. This points to the fact that 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. One potential interpretation of these results is that there is an ideal virtual distance from the goal configuration for the RDW algorithm to switch to alignment only redirection. Other factors that were not studied in this experiment may have an influence such as physical environment size, or the ratio of total avoidance redirection with respect to alignment redirection. These factors would be interesting to investigate in future experiments.

## 5 Conclusion and Future Work

In this paper, we introduced the concept of alignment, a novel mechanic that uses RDW techniques to align the user's virtual and physical worlds, thereby addressing a problem that exists with several VR locomotion techniques. By enabling VR experiences to leverage physical interactivity, alignment has the ability to increase the level of immersion and sense of presence afforded by the medium. This work presented the mathematical foundations and initial experiments that point towards the value of future research, development, and evaluation of new alignment techniques. To this end, we have identified several interesting research vectors to inform future work.

In unconstrained scenarios, we do not expect RDW algorithms to be able to perfectly keep the user away from boundaries and obstacles, which is why resets are employed. We think that a similar concept can be utilized to correct for the positional alignment error remaining after alignment should be completed. One example would be a positional reset, where instead of pausing the experience to have the user rotate towards a more favorable orientation, we employ an intervention that has the user translate to a particular location. Gretchkin et al. implemented a similar concept, which they called "Rotate and Walk" to achieve contextual relevant resets [6]. We suggest that continuous alignment along with positional resets may collectively represent a more generalizable solution, and more research into combined techniques and algorithms would be valuable.

One major problem posed by using RDW techniques to achieve a desired mapping involves difficulties in steering the user in a specific direction without also inducing rotation. For example, with rotation and curvature gains, there is no way to steer users towards their right while having them continue to face forward. In recently published work, You et al. introduced a new RDW gain that would be able to address this specific situation; however, this technique has yet to be perceptually validated [26]. In general, new redirection techniques that allow for more flexible manipulations of the user's mapping would increase the effectiveness of alignment algorithms.

We demonstrated how alignment can be used in conjunction with avoidance RDW to converge a user's mapping, which could potentially be used to enable interactions with the physical environment. However, as previously stated, there are several other locomotion techniques that do not maintain a constant mapping. A growing number of VR applications are designing for "room-scale VR," and with the recent advent of head mounted displays that support inside-out tracking, the concept of "building-scale VR" is also emerging. These applications commonly use locomotion techniques that involve a combination of limited physical walking for navigation within the bounds of the physical environment, along with virtual locomotion for navigating over greater distances. Because these experiences allow for physical movement, they have the ability to employ physical interactions to enhance the user's experience. However, as soon as the virtual locomotion mechanism is utilized, the user's mapping diverges and physical interactions will break. It would be interesting to develop new locomotion interfaces that combine continuous alignment during physical movement with subtle positional resets integrated into virtual movement. In general, there is great potential in algorithms and interfaces based on alignment, and exploring them will be a promising line of future research.

## REFERENCES


[1] M. Azmandian. *Design and Evaluation of Adaptive Redirected Walking Systems*. PhD thesis, University of Southern California, 2018.

[2] M. Azmandian, T. Grechkin, M. T. Bolas, and E. A. Suma. Physical space requirements for redirected walking: How size and shape affect performance. In *ICAT-EGVE*, pp. 93–100, 2015.

[3] M. Azmandian, T. Grechkin, and E. Suma Rosenberg. An evaluation of strategies for two user redirected walking in shared physical spaces. In *IEEE Conference on Virtual Reality and 3D User Interaction*. IEEE, 2017.

[4] E. R. Bachmann, E. Hodgson, C. Hoffbauer, and J. Messinger. Multi-user redirected walking and resetting using artificial potential fields. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2022–2031, 2019.

[5] B. J. Congdon and A. Steed. Sensitivity to rate of change in gains applied by redirected walking. In *ACM Symposium on Virtual Reality Software and Technology*, p. 3. ACM, 2019.

[6] T. Grechkin, M. Azmandian, M. Bolas, and E. Suma. Towards context-sensitive reorientation for real walking in virtual reality. In *IEEE Conference on Virtual Reality*, pp. 185–186. IEEE, 2015.

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

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

[9] B. E. Insko, M. Meehan, M. Whitton, and F. Brooks. *Passive haptics significantly enhances virtual environments*. PhD thesis, University of North Carolina at Chapel Hill, 2001.

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

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

[12] L. Kohli, E. Burns, D. Miller, and H. Fuchs. Combining passive haptics with redirected walking. In *Proceedings of the 2005 international conference on Augmented tele-existence*, pp. 253–254. ACM, 2005.

[13] B. H. Krogh and T. J. Graettinger. Maneuverability constraints for supervisory steering control. In *IEEE Conference on Decision and Control*, pp. 279–284. IEEE, 1985.

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

[15] J. Messinger, E. Hodgson, and E. R. Bachmann. Effects of tracking area shape and size on artificial potential field redirected walking. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2019.

[16] T. Nescher, Y.-Y. Huang, and A. Kunz. Planning redirection techniques for optimal free walking experience using model predictive control. In *IEEE Symposium on 3D User Interfaces*, pp. 111–118. IEEE, 2014.

[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, 2018.

[18] S. Razzaque. *Redirected walking*. University of North Carolina at Chapel Hill, 2005.

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

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

[21] 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, 2010.

[22] 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.

[23] J. Thomas and E. S. Rosenberg. A general reactive algorithm for redirected walking using artificial potential functions. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2019.

[24] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr. Walking¿ walking-in-place¿ flying, in virtual environments. In *ACM SIGGRAPH*, pp. 359–364. ACM Press/Addison-Wesley Publishing Co., 1999.

[25] 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 *ACM Symposium on Applied Perception*, pp. 41–48, 2007.

[26] C. You, E. Suma Rosenberg, and J. Thomas. Strafing gain: A novel redirected walking technique. In *ACM Symposium on Spatial User Interaction*, p. 26. ACM, 2019.

[27] R. Yu, Z. Duer, T. Ogle, D. A. Bowman, T. Tucker, D. Hicks, D. Choi, Z. Bush, H. Ngo, P. Nguyen, et al. 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*, pp. 313–319. IEEE, 2018.

[28] 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.