The Chaotic Behavior of Redirection – Revisiting Simulations in Redirected Walking

Christian Hirt*+
Innovation Center Virtual Reality
ETH Zurich, Switzerland

Yves Kompis†+
Vision for Robotics Lab
ETH Zurich, Switzerland

Christian Holz‡
Department of Computer Science
ETH Zurich, Switzerland

Andreas Kunz§
Innovation Center Virtual Reality
ETH Zurich, Switzerland

Abstract

Redirected Walking (RDW) is a common technique leveraged to allow real walking for exploring large virtual environments in constrained physical tracking spaces. Effective RDW is challenging due to its complexity and disturbance factors (e.g., spontaneous user behavior). Existing techniques range from combinations of simple motion scaling to more elaborate curvature injections and reactive, predictive, or scripted steering concepts. However, many of these approaches were evaluated in simulation only, and researchers argued that the findings would translate to real scenarios to motivate the effectiveness of their algorithms. Using the Redirected Walking Toolkit and its virtual path generator, a randomized waypoint-based path generator has been common practice, although its built-in simplifications assume sequential user behavior regarding translation and rotation.

In this paper, we argue that pure simulation-based evaluations employing such simplified path generators require critical reflection. We demonstrate RDW simulations that show the chaotic process fundamental to RDW, in which altering the initial user's position by mere millimeters can drastically change the resulting steering behavior. This insight suggests that RDW is more sensitive to the underlying data than previously assumed. Thus, we rigorously analyze the influence of commonly used synthetically generated paths on multiple state-of-the-art steering concepts and compare them against previously recorded real paths.

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

1 Introduction

Real walking is considered the most natural and intuitive kind of locomotion in virtual reality [34]. However, it still remains challenging to explore infinite virtual environments (VEs) in limited physical spaces while allowing users to really walk.

Guiding physical walking through redirected walking (RDW), introduced by Razzaque et al. [27, 28], mitigates the challenges by exploiting human reliance on visual stimuli over others. Developing a general solution, however, is subject to active research. RDW induces certain, mostly visual, manipulations to lead a user to subconsciously adjust their real, physical motion to compensate for potential visuo-vestibular mismatches. In ideal cases, users can be infinitely "redirected" onto a circle while they believe they are walking a straight path.

RDW approaches fall into three categories: reactive [14, 28], predictive [21, 40], and scripted [3, 37]. These categories describe how the respective steering algorithm, also called RDW controller in other literature, takes the VE into consideration. A steering algorithm follows a predefined steering concept and determines how the redirection is applied to the individual user in each frame. For example in the case of a scripted RDW approach, these steering algorithms can be optimized since the user's path through the VE is given, i.e. *scripted*. As long as the user strictly follows this predefined path, the optimized steering algorithm safely guides the user through the limited physical space. Reactive RDW approaches do not evaluate the VE, but determine the redirection only from the user's current state (i.e., position and heading) in the physical environment. In predictive RDW approaches, the user's behavior and virtual states are used to estimate a prediction of where the user might travel to. This prediction (i.e., a predicted state or even a complete path) is then overlaid to the physical space to adjust the redirection to guarantee a safe, collision-free exploration. Whereas scripted RDW approaches can be optimized for each virtual environment, both reactive and predictive concepts are still subject to active and fundamental development.

Novel steering concepts are being proposed in rapid succession, but many of these have been evaluated solely in simulation. While simulation was not initially intended to be representative of real user performance, but rather in a comparative manner to efficiently measure relative performances when using different steering concepts, it has become one of the default validation procedure for RDW techniques. However, evaluations on simulation performance by default omit any kind of subjective perspective such as simulator sickness [16], user exhaustion [12], age [15], or gender implications [23]. Therefore, even when RDW protocols are closely followed during evaluation (e.g., using imperceptible redirection gains [29]), the steering concept that performs best in simulation may still be rejected by real users in the worst case, which diminishes the algorithms practical value.

The simulations used for evaluation in related efforts are often derived from the Redirected Walking Toolkit (RDWT [4]) due to its availability and versatility. The RDWT is a comprehensive simulation environment for RDW with many built-in steering algorithms and includes a simulated walker that follows specific exploration patterns (e.g., building navigation) created by a virtual path generator. However, the implementation of this randomized waypoint-based path generator simplifies human locomotion to sequential translations and rotations. These simplifications, therefore, neglect important human traits of navigating in new environments, resulting in unnatural polygonal walking trajectories.

In this paper, we address the discrepancy between commonly used simulation-based validation using synthetic paths and recorded paths from real users. We first investigate the effect of just slight varia-

*e-mail: hirtc@ethz.ch
†e-mail: ykompis@ethz.ch
‡e-mail: christian.holz@inf.ethz.ch
§e-mail: kunz@iwf.mavt.ethz.ch

+These authors contributed equally.

tions in starting position or heading on simulations and demonstrate that these can result in seemingly random simulation outcomes of a single path, even though all other conditions remained identical and steering algorithm and gain restrictions remained unchanged. We conclude that the behavior of such simulations qualifies as *chaotic*, which describes dynamic systems with seemingly random states of irregularities and disorder which are actually governed by underlying patterns, highly sensitive to initial conditions [6]. Based on this insight, we quantify the difference in outcomes of simulated performance between the synthetically generated paths and the paths recorded in real user studies. We show the sometimes drastic differences between the synthetic paths and the paths recorded in real user studies in terms of common RDW evaluation criteria such as meters per reset (also called distance between resets in other literature).

Following these simulations, we lay out why simulation results that purely rely on synthetic user paths should be interpreted with caution. At the example of several kinds of user paths, synthetically generated or previously recorded in user studies, we discuss the impact of RDW steering algorithms and tracking space sizes on the outcome. We further highlight the steering concepts that are particularly prone to misleading results in simulation when compared to real user paths and conclude our findings with recommendations on handling RDW evaluations in the future.

We note that the goal of our work in this paper is in no way to question other contributions or approaches in the field. Our intention is to add to the growing body of work and analysis of how delicate and sensitive RDW is to effects that occur during real user tests. Our second goal is to show that validation based on simulation alone may result in a loss of real-world relevance to a certain extent. Simulation-based validation has a clear importance and effectively supports smooth development, rapid prototyping or the collection of large-scale data. For these reasons, simulation will stay relevant as the field of RDW matures, however deriving better and more realistic locomotion models that grasp humans walking nature will greatly enhance the weight such simulation are able to generate. With our results, we hope to encourage researchers to extend their evaluation of new steering concepts to user studies in order to gain a more comprehensive understanding of an algorithm's performance in real conditions with real users.

2 Related Work

Following Suma et al.'s taxonomy [31], manipulations applied in RDW are broadly classified into discrete and continuous, overt and subtle, and reorientation and repositioning. In most RDW approaches, these so-called redirection techniques are chosen from two specific classes: continuous, subtle reorientation and -positioning in normal situations, and continuous, overt reorientation for collision avoidance. Here, a normal situation infers a safe physical environment for the user, meaning that there is no danger of collisions of any kind. Accordingly, this first class of continuous, subtle reorientation and -positioning is applied for the major part of the exploration of a VE, usually consisting of *curvature*, *rotational* and *translational* gains as base redirection techniques. Whereas rotational and translational gains scale the respective motion of a user, the curvature gain induces a slight rotation around the user's yaw while they are walking, to nudge them towards a real-world curve while following a straight path in the VE. More details on redirection techniques can be found in the comprehensive reviews by Langbehn & Steinicke [17] and by Nilsson et al. [24].

In case users walk too close to a boundary of the physical environment or are endangered of colliding with physical objects or other users, collision avoidance is activated and users are instructed to follow a *reset* procedure [5,20,25,32,35]. Resets stop a user's continuous motion by instructing them to follow certain motion patterns to reorient or -position them in safe conditions in the real environment. None of the three initially introduced approaches are in common use nowadays due to their situational inflexibility, but the *2:1-reset* (R21) remains most notable. Instead of its original definition, researchers now use simple implementations such as *reset to center* (R2C), *reset to gradient* (R2G), or the more complex *modified reset to center* or *step forward reset to gradient* [32], which also take non-convex tracking spaces into consideration. Since we only simulated convex tracking spaces, we limited our considerations to R21, R2C, and R2G, which ask the users to stop and spin in place. When the reset is triggered, the user's real state is taken as an input and a reset vector is determined. For R21, the reset vector is defined by the inverse of the current heading direction. For R2C, the vector points towards the center of the tracking space. For R2G, the vector used for the reset is determined by the concept of artificial potential fields (APFs), which is discussed more in detail later on. While the user follows the spinning instructions, a rotational gain is applied reorienting the user such that they spin 360° in the VE, but are aligned with the reset vector in reality after the reset is completed.

Steering Algorithms

Scripted steering algorithms in RDW require a predefined virtual path and are not applicable more generally [3]. For this reason, we do not consider them in this work.

In contrast, reactive steering algorithms have become popular due to their versatility and most recent capability of handling multiple users in the same real tracking space [5]. The three approaches initially introduced are *steer to center* (S2C), *steer to orbit* (S2O), and *steer to changing targets* [27]. S2C, similar to R2C, always aims at steering the user towards the center of the tracking space. S2C comes with an unwelcome oscillation behavior of the steering algorithm whenever a user faces straight towards the center but slightly veers around the redirection vector (i.e. the vector connecting the user to their redirection target) provoking a strong shift between the gains. Additionally, passing straight through the center of the tracking space requires massive gains to readjust the user back towards a heading direction aligned with the redirection vector. S2O mitigates these issues and redirects the user on a circular orbit with the origin in the center. Alternatively, steer to changing targets directs the user to multiple different, strategically distributed targets in the physical room. While both S2O and steer to changing targets address parts of S2C's issues, studies show that they could not outperform S2C over longer time periods [14, 27]. As a result, S2C has become a generally accepted reference for steering algorithm validation.

More recent reactive approaches follow the concept of APFs, such as *P2R* [32] and *APF RDW* [5]. They represent the real tracking space by a potential field, assigning dangerous areas a high potential (e.g., close to a boundary), while free open space is considered safe and thus holds a low potential. Since APFs cover the whole tracking space, the user always moves within this representation. By design, APFs are differentiable at all positions, such that regardless of the user's position, the gradient points to an area with a lower potential, thus providing an easy-to-follow redirection vector. APFs bypass the previously prevalent simplifications of convex tracking spaces and single-user operation for RDW, scale to any tracking space shape, and can also superimpose dynamic obstacles (e.g., a second user). For consistency purposes, we refer to APF redirection using steer to gradient (S2G) in the remainder of this paper.

Predictive steering algorithms are rarer in literature than reactive steering algorithms due to the complexity of their implementation. Early predictive approaches (e.g., *RFED* [26]) only relied on predicting the user's future path based on a skeleton graph and their walking direction. Later approaches have pursued more complex core concepts. For example, *FORCE* uses constrained VEs and follows a skeleton graph consisting of edges for straight paths and nodes in crossings [40]. Each possible pathway is then considered as a prediction candidate and a discrete set of redirection gains is employed for distortion. For each node and edge, the set of gains

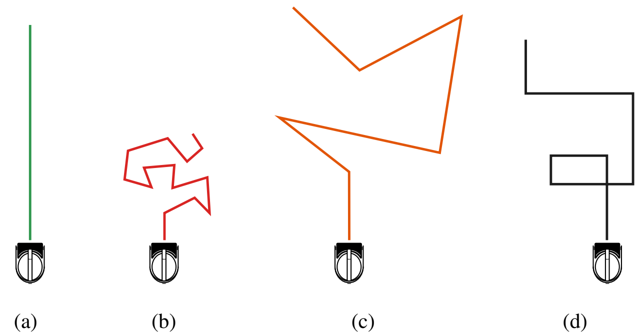


Figure 1: The RDWT can generate different synthetic path types: (a) Straight, (b) Small Exploration, (c) Large Exploration, and (d) Building Navigation. Synthetic paths are built from way-points with straight connections, and rotation-in-place at each way-point.

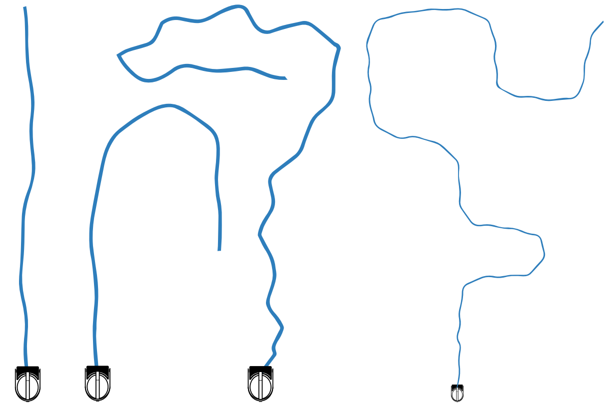


Figure 2: A selection of real walking trajectories from different user studies shown at different scales. Users completed different tasks e.g. traversing a maze, unguided search, exploration, and more.

is applied resulting in a multitude of final states, which can be evaluated to identify a score. This score is determined by the distance a user could continue to walk in a straight line without colliding with any kind of physical boundary. Finally, the best score then defines the redirection to be applied to the moving user. On the other hand, *MPCRed* follows Bellman's principle of optimality and recursively evaluates the optimal redirection gain based on a cost function consisting of reset, redirection and collision costs [21]. However, MPCRed also relies on constricted VEs to simplify calculations.

So far, most predictive approaches have been restricted to single-user experiences while exploring constrained, static VEs. Few approaches combined predictive RDW with multiple users in the same tracking space. For example, Hirt et al.'s proposed concept builds on predictive APFs and optimality [13]. Dong et al. presented dynamic APFs that superimpose users' assumed-to-be constant motion to influence the redirection vector [8]. This approach creates predicted safe zones in the tracking space, towards which users can be redirected. Lastly, Lee et al. introduced *steer-to-optimal-target* [18] as an algorithm comprising a simple steer-to-target steering algorithm (i.e. similar to S2C) and a deep Q learning network. They integrated real user paths to derive model parameters of their artificial neural network, which they then applied to identify where a predicted user target optimally lies. They later expanded on their findings and presented *multi-user-steer-to-optimal-target* [19], applying again a steer to target algorithm but enhanced with a D3QN artificial neural network.

More recently, researchers have proposed other learning-based approaches (e.g., *S2L* [30]) or *alignment* concepts (e.g., [33,36]). Both have not demonstrated a predictive nature yet, but show potential for development towards this direction.

3 Methodology

We now outline the simulations that we investigate and compare to empirical results. We first provide an overview of the RDWT and the diversity of synthetically generated paths and then introduce our modifications to enable simulating real user paths. Finally, we discuss performance measures that are common in literature.

3.1 Redirected Walking Toolkit

Our simulation tool builds on Azmandian et al.'s RDWT [4], containing template classes for *redirectors* and *resetters* that implement steering algorithms and resets, respectively. Additionally, the toolkit provides popular steering algorithms such as S2C and S2O that can be used as baselines. Both steering algorithms (i.e. S2C and S2O) are used without alteration to the initial implementation.

During deployment, redirectors and resetters get access to the current physical state in the tracking space and act as an abstraction layer that converts physical coordinates into virtual coordinates. During development, this physical location is usually not readily available. Thus, the RDWT also provides a simulated walker acting as a replacement of a real user.

The simulated walker follows a procedurally generated random synthetic path that emulates different types of virtual walking trajectories. These paths are generated at runtime and consist of a set of virtual way-points connected by straight paths. The toolkit offers multiple types of locomotion behavior as seen in Fig. 1. The four path types—*Straight*, *Small Exploration*, *Large Exploration*, and *Building Navigation*—consist of a section of straight path with a constant velocity, followed by a rotation in place with a constant angular velocity. The simulated walker's heading is thus always facing in the current walking direction and there is no explicit interaction between the simulated path and the redirector (except for an automated rotation inserted when a reset procedure is in progress). Depending on the type, the length of straight paths and each rotation angle are differently randomized.

3.2 Simulating Real Trajectories: A Collected Dataset

Although the different synthetic path types offer some variety, all greatly differ from the natural behavior of a real user walking in a VE. Such differences include acceleration and deceleration, independent heading and walking directions, perturbations of paths caused by the natural veer of the center of gravity while walking, strafing behavior, curved trajectories, and others. Instead of modelling these and other nuances as part of the path generation process, we make use of the vast amount of already available data in the form of real user paths recorded in previous user studies. These paths are inherently more realistic than fully synthetic paths and, thus, hold promise for improving the accuracy of RDW simulations.

We collected walking trajectories from more than 20 different user studies on RDW involving more than 800 participants. VEs and tasks varied widely in these studies and subsequently the trajectories represent a variety of path types, but the VEs in which the trajectories were recorded in can still be roughly allocated to the four synthetic path types previously introduced, thus ensuring comparability. Overall, our collected dataset contains more than 30 000 individual path segments, where multiple segments can originate from the same user in different parts of a study. To get meaningful redirection results, we prune this raw dataset to only contain path segments of sufficient duration and path length. In our case, we filtered for a minimum

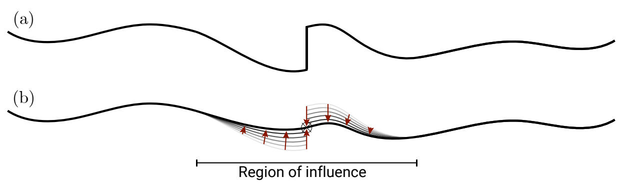


Figure 3: Redirection artefacts from subtle discrete redirection techniques as shown in (a) are removed by linear interpolation in a small region around the discontinuity (b).

Path Type	Segment Length	Rotation Angle
Small Exploration	$U(0.2\,\text{m}, 1\,\text{m})$	$U(-\pi, \pi)$
Large Exploration	$U(1\,\text{m}, 3\,\text{m})$	$U(-\pi, \pi)$
Building Navigation	$U(1\,\text{m}, 3\,\text{m})$	$-\pi/2$ or $\pi/2$

Table 1: Parameters used for synthetic path generation where $U(a,b)$ is the uniform distribution between a and b.

duration of 60 s and a minimum path length of 40 m, leaving more than 32 km of real and unique walking trajectory segments. Each trajectory segment is treated as an independent experiment without any chaining of segments. This ensures that no additional path artefacts emerge from stitching multiple path segments together. A selection of different path segments is shown in Fig. 2.

We sourced virtual user paths instead of the physical paths where possible. Thus, paths are not bound by the study's physical tracking space, but the raw paths also contain redirection artefacts, such as resets and discontinuities from subtle discrete redirection techniques. We addressed this through filtering and removed all resets and other artefacts. Discontinuities in position and yaw are detected by a threshold on the instantaneous linear and angular velocity. The disjoint path segments are then joined at the discontinuities by linearly interpolating the tail and head of the two disjoint segments as shown in Fig. 3. Resets are detected by finding full virtual rotations limited to a small area. The removal procedure of these resets is the same as for instantaneous discontinuities, but with a slightly larger region of influence for the interpolation.

For each path segment in the collected dataset, we generate a set of equivalent synthetic path segments. Equivalent path segments are characterized by an equal duration and equal average walking velocity. These metrics are calculated on a per-path basis on the dataset of real path segments.In essence, we generate for each synthetic path type a total of 32 km of equivalent path segments. Other parameters used during the path generation step are listed in Table 1.

3.3 Real Paths in the Redirected Walking Toolkit

In order to simulate RDW with real paths in our analysis, we replaced RDWT's *runtime* synthetic path generation with a more versatile movement simulator. This simulator can input any prerecorded path drawing from our collection above into the simulation. Our approach therefore does not only allow simulating real trajectories from previous studies, but also maintains the compatibility with the synthetically generated path types as described above. It also allows future research to compare observed trajectories with synthetic paths in simulation.

3.4 Performance Measures

To be able to compare different steering algorithms, we define measures that reflect an algorithm's performance. We model our measures based on the primary goals of RDW: *ensuring user safety* and *increasing user presence*, as RDW's purpose is supporting natural locomotion for increased presence in a VE [27]. For presence, we use the mean virtual distance between resets to quantify a steering

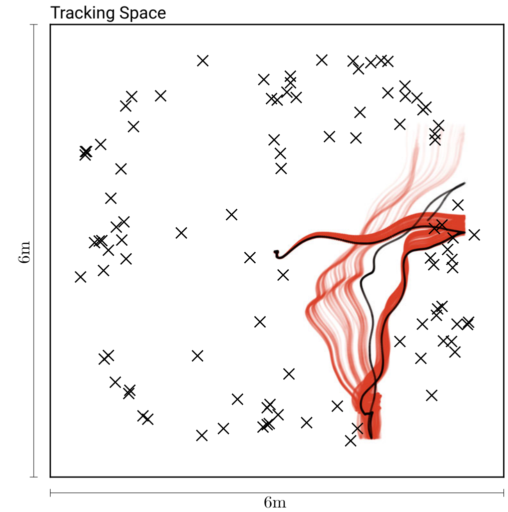


Figure 4: Exemplary result: 100 simulations of the same path with slightly different initial conditions. The starting position and orientation in the center of the tracking space are randomly offset by just $\pm 0.5\,\text{cm}$ and $\pm 0.5°$. The initial few meters of the diverging paths are shown in red, crosses mark the final position after the simulating the full 180 m path, a single path is highlighted in black.

algorithm's performance. For user safety, we use the mean distance to the closest wall to quantify its performance, similar to Engel et al. [9] in their gain identification. Using the mean distance between resets over the number of resets has the benefit of normalizing over the path length. Additionally, the mean distance between resets also provides a more intuitive overview of the performance of steering algorithms compared to mean time between resets, since it accounts for the velocity of a user as well (e.g., when a user is standing still).

4 Results & Discussion

We run extensive simulations for an in-depth examination of RDW behavior and performance comparison of individual steering algorithms under diverse conditions.

4.1 Chaotic Behavior of Redirected Walking

In a pre-study simulation, we test the influence an initial state has on the outcome of a simulation. Whereas we acknowledge the micro-scale experiment conducted by Azmandian [3], in which he analyzed the variance of end positions for different users with different unique redirection techniques, we wish to delve deeper and highlight the RDW's high sensitivity to slight deviations while freezing all boundary conditions except for the initial state. Accordingly, we take a real 180 m path from our dataset, use a combination of S2G and R2G for redirection and simulate in a 6m × 6m tracking space. We simulate the path 100 times, where the only difference between simulations is the initial position and orientation which were chosen uniformly at random between $\pm 0.5\,\text{cm}$ and $\pm 0.5°$ from the center of the tracking space, respectively. The start of all redirected paths is shown in the center of Fig. 4 where we can observe that the paths start to diverge from each other already in the first few meters. The final positions of the simulations (marked by "×") are spread over the entire tracking space. One reason for this diversity in the final state are situations where one path narrowly misses the reset boundary while another path does not. Such a situation is highlighted in black in Fig. 4 where the black path requires to reset on the right wall, leading to a completely independent course of redirection actions from that point on-wards. Notably, these discrepancies between simulations occur

Steering Algorithms	Reset Techniques	Tracking Spaces	Path Types
NULL S2C S2O S2G	R21 R2C R2G	$4\,m \times 4\,m$ $4\,m \times 6\,m$ $4\,m \times 8\,m$ $6\,m \times 6\,m$ $6\,m \times 8\,m$ $8\,m \times 8\,m$ $8\,m \times 10\,m$ $8\,m \times 12\,m$ $10\,m \times 10\,m$	Real Straight Small Exploration Large Exploration Building Navigation

Table 2: Conditions used for the main simulation study. Each combination of these conditions is simulated. NULL denotes no redirection.

while redirection gains are always limited within the imperceptibility thresholds as proposed by Steinicke et al. [29].

We repeated this simulation for multiple paths of similar lengths, resulting in the same characteristics, therefore these results suggest that RDW is a chaotic process which is very susceptible to even slight changes in the input data. We do understand that such final positions of paths are by no means pertinent performance criteria for RDW, but we wish to emphasize how prone RDW is to such a butterfly effect, resulting in vastly different paths based on simple and small deviations of data. So far, we only showed how altering initial conditions can influence the redirection's behavior, but it is not clear how entirely different path types might influence the performance of different steering algorithms.

4.2 Simulation

Given these observations, we run an extensive simulation in which we simulate different steering algorithms, with different real and synthetic paths, in a selection of different tracking spaces. The full list of simulation conditions is listed in Table 2.

Each unique combination of conditions is simulated on all 32 km of every path type. Since we observed chaotic behavior in the pre-study, we randomized the initial positions and simulated two different starting points for every path. The randomization was carefully crafted with a fixed seed, such that each steering algorithm had the exact same initial conditions for every path. To ensure the subtlety of the redirection techniques, we again followed the gain values proposed by Steinicke et al. [29]. In total, we performed approximately 200 000 individual simulations which equates to simulating approximately 35 000 km of RDW.

Reset Performance To present the simulation results, we jointly plot the mean distance between resets d_{reset} against the mean distance to the closest wall d_{wall}, showing marginal distributions of each variable as seen in Fig. 5 in which we visualize the performance of reset techniques. In this plot, each individual simulation is represented by a single dot. Isolines and the marginal distributions help to distinguish the relative density of dots. Since a good RDW algorithm is characterized by large distances between resets as well as large distances to walls, the top right corner of this plot signifies good performance. Thus, Fig. 5 shows that R21 performs worse than R2C or R2G, which both seem to be equivalent. This is to be expected, since R2C and R2G ought to behave similarly in the case of a square, convex tracking space. To verify these findings we perform a pair-wise multivariate Wald-Wolfowitz test [11] which reveals that R21 performs significantly different from both R2C ($W = -11.214$, $p < 0.05$) and R2G ($W = -10.592$, $p < 0.05$), while R2C and R2G appear to have a similar distribution ($W = 0.701$, $p = 0.7585$).

Tracking Space Performance As another verification of our simulations, we plot the performance of S2G paired with R2G on real paths for each tracking space in Fig. 6. In essence, this plot is equivalent to the top marginal distribution from the representation in Fig. 5. As expected, d_{reset} increases together with the size of the tracking spaces. Additionally, the standard deviation of the distribution grows with the size of the tracking space. This means that the spread of d_{reset} between individual simulations increases with the tracking space size.

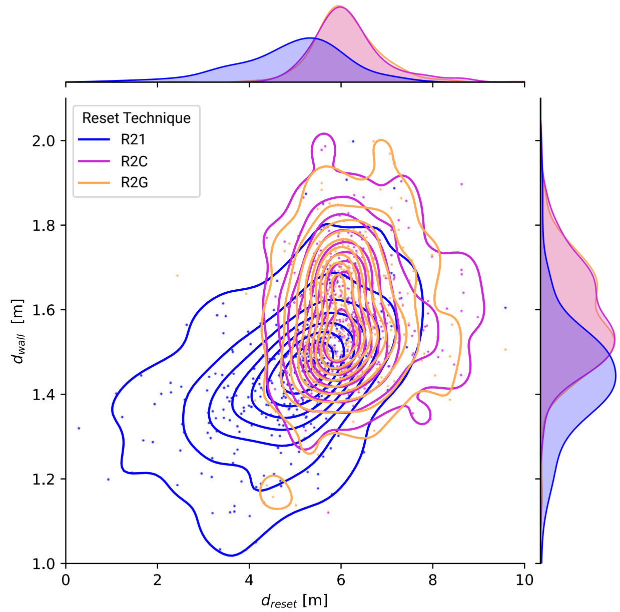


Figure 5: Performance comparison between reset techniques. Each dot represents a single simulated path. Isolines and marginal distributions aid in showing the relative density of dots. The top right corner of this plot represents good performance: a large distance between resets and a large distance from walls. All simulations shown here use **S2G** and were simulated with real paths in a $6\,m \times 6\,m$ tracking space. While R2C and R2G are very similar in terms of performance, R21 is noticeably worse.

Steering Algorithm Performance Comparing the performance of different steering algorithms on real paths in a $6\,m \times 6\,m$ tracking space reveals that S2O only slightly improves d_{reset} over no redirection (i.e. NULL) as seen in Fig. 7. Apparently, S2O achieves this by sacrificing d_{wall} which is somewhat reduced. S2C and S2G perform similarly and both clearly outperform NULL in both performance measures. A pair-wise multivariate Wald-Wolfowitz test confirms these findings, showing that NULL and S2O are significantly different from all other steering algorithms (all $p < 0.05$) while S2C and S2G appear to have similar distributions ($W = -1.252$, $p = 0.1052$).

Ideally, the performance distributions for each steering algorithm on the different synthetic path types should closely match the result from Fig. 7. If this were the case, we could safely use generated paths to estimate a steering algorithm's performance on real paths.

But if we plot the same data for each synthetic path type, as shown in Fig. 8, we observe that these distributions are severely distorted compared to Fig. 7. This means that not all steering algorithms perform equally well on real or any of the synthetic paths. Please note that these plots do not share axes, which needs to be considered when interpreting the results.

Not only are there notable differences from real to synthetic, but also large differences among the synthetic path types. Steering algorithms seem to struggle with Straight paths, where d_{reset} is relatively small compared to real paths, while they perform much better on Small Exploration paths. For example, d_{reset} for S2G/R2G

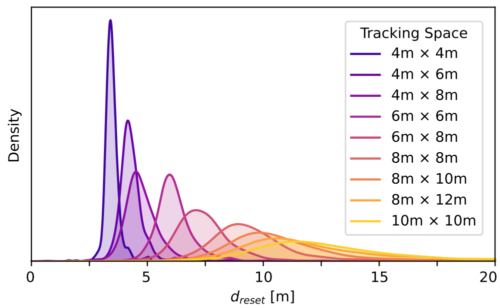


Figure 6: Performance comparison between tracking spaces. As the tracking spaces increases, so does the distance between resets. Shown are results from the subset of simulations using **S2G** and **R2G** on real paths.

nearly doubles when simulating Small Exploration paths compared to real paths.

Real vs. Synthetic Paths To further investigate the influence a given path type has on performance metrics, we take a closer look at S2G in various tracking space sizes. In Fig. 9, we plot the performance broken down to the different path types for multiple tracking spaces using S2G and R2G. As expected, none of the synthetically generated paths behaves similarly to the real paths. Even though the differences between path types are less distinct for larger tracking spaces, the distribution of performance measures of each group is still clearly distinguishable. A multivariate Wald-Wolfowitz test confirms that all pair-wise comparisons are significantly different (all $p < 0.05$) in all tracking spaces shown. Interestingly, in most cases, the steering seems to be more effective on synthetic paths. Especially d_{reset} is often much larger for synthetic paths. This might be due to the fact that these synthetic paths contain a lot of rotations in place which are practically completely missing in the real dataset. During these rotations, S2G can apply rotation gains which are more effective than curvature gains within the detection thresholds. Although not shown here, these differences are not limited to this specific reset technique and steering algorithm.

Real vs. Accumulated Synthetic Paths Following the tendencies shown in Fig. 8 and Fig. 9, we further emphasize the discrepancy between real and synthetically generated user paths by summarizing all synthetic paths into one representation (see Fig. 10). Again, if the two path types were similar enough to represent each other, the resulting distributions should closely match. However, even though some peaks may seem close, the overall distributions vary considerably and the result thus yields a misleading performance comparison. This observation is further confirmed by performing a pair-wise multivariate Wald-Wolfowitz test that shows that the distributions are significantly different ($W = -27.635$, $p < 0.05$).

Distance between Resets Fig. 11 further accentuates the relative performance differences of steering algorithms on different path types using d_{reset}. The figure shows the mean and standard deviation of the reset distance for every combination of steering algorithm and path type, which confirms a strong influence of path types on performance. The first two columns again underline the main inconsistencies between real and synthetic user paths, divided into the different synthetic path types towards the right. Without redirection, d_{reset} is fairly similar for all path types. But especially for S2G and S2C, the performance varies considerably. As a result, simulations on synthetic paths inflate S2C's and S2G's capabilities severely.

Finally, we show a complete overview of our simulation results in Fig. 12. It visualizes redirection performance of each steering

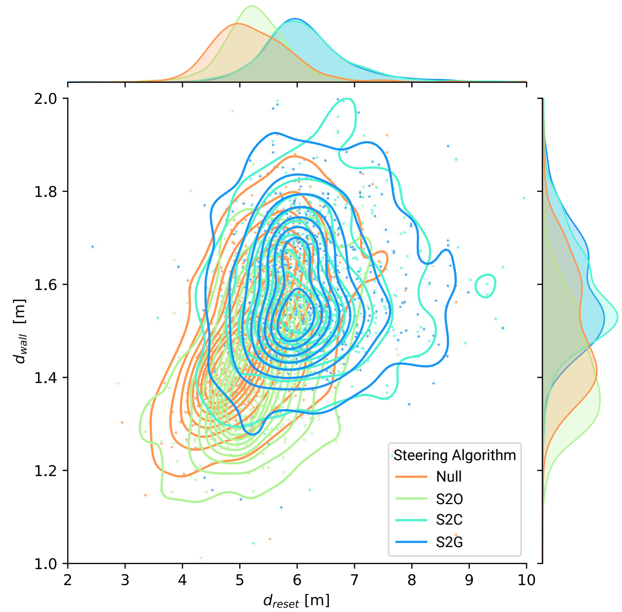


Figure 7: Performance comparison between steering algorithms. Each dot represents a single simulation. All simulations shown here use **R2G** and were simulated with real paths in a $6\,\mathrm{m} \times 6\,\mathrm{m}$ tracking space. The top marginal distribution indicates lower average distance between resets for S2O and NULL compared to S2C and S2G. Similarly, the average distance to the closest wall is lowest for S2O and NULL, visible in the right marginal distribution. Overall, S2O only slightly outperforms NULL while S2C and S2G manage to outperform NULL and S2O in both metrics. S2C and S2G perform similarly.

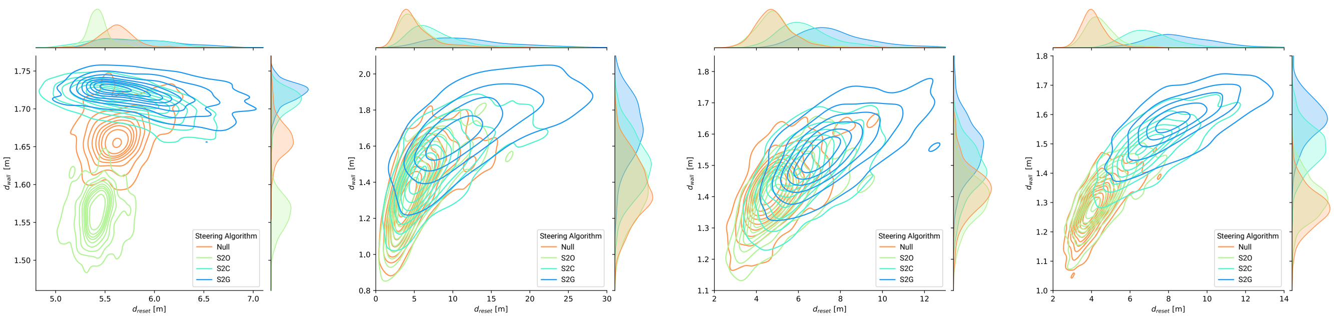
algorithm, with every path type for every tracking space. Once again, the synthetic paths are unable to accurately predict redirection performance for real paths and leaves room for misjudgement and misinterpretation.

5 Conclusion & Future Work

In this paper, we investigated simulation-based approaches in RDW steering algorithm validation. In a pre-study simulation, we demonstrated that RDW is a chaotic process and that just marginal alterations to initial conditions can already have drastic effects on the redirection and the resulting user path. Furthermore, we evaluated a total of approximately 35,000 km paths in more than 200,000 individual simulations to investigate multiple implicit assumptions that have been part of related approaches.

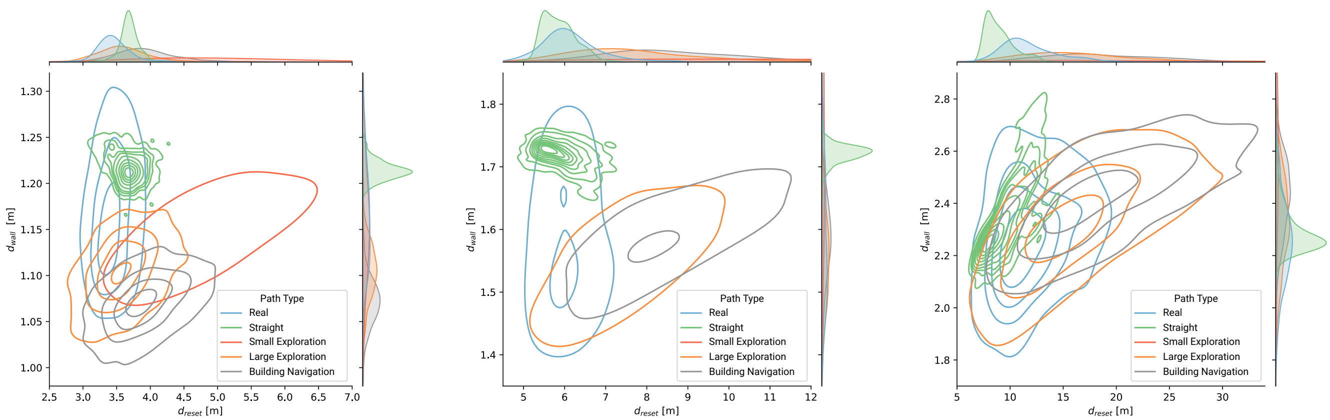
Our simulations in this paper confirm established facts in the RDW literature, which also validates our simulation environment. Among others, we showed that the resetters R2C and R2G outperform R21, that increasing the tracking space size improves the steering algorithms' performance, and that S2C and S2G slightly outperform S2O and NULL redirectors.

However, comparing the performance distribution of the steering algorithms between real paths (Fig. 7) and synthetic paths (Fig. 8), we revealed notable distinctions in the performance of the redirection. Pursuing these observations, we simulated the exemplary pair of S2G/R2G in multiple tracking spaces on all five path types and demonstrate that the different properties of the path types create a larger discrepancy in the results than previously assumed (Fig. 9). The performance of the pair S2G/R2G should be consistent over the real and synthetic paths to ensure comparability, however, none of the synthetic paths manages to grasp the real path's traits and thus



(a) Straight (b) Small Exploration (c) Large Exploration (d) Building Navigation

Figure 8: Performance of different steering algorithms on synthetic data broken down by path type. Shown are simulation results from all steering algorithms using **R2G** in a $6\,\mathrm{m} \times 6\,\mathrm{m}$ tracking space. Please note the different axis scales between plots.



(a) $4\,\mathrm{m} \times 4\,\mathrm{m}$ tracking space (b) $6\,\mathrm{m} \times 6\,\mathrm{m}$ tracking space (c) $8\,\mathrm{m} \times 12\,\mathrm{m}$ tracking space

Figure 9: Performance of different path types in three different tracking spaces. Shown are simulation results from the subset of simulations that used **S2G** and **R2G**. In all tracking spaces, we can clearly see the performance difference between path types shown by the differences in the distributions. In general, d_{reset} is typically overestimated when simulating with synthetic paths. The same holds true for d_{wall}. Especially for small tracking spaces, Straight paths lead to a much higher average d_{wall}. The low standard deviation from Straight path stems from the fact that all simulated paths simulate the exact same path with the only difference being the initial configuration and the overall path length. Since the resulting distributions for Small Exploration paths in the two larger tracking spaces are extremely flat, we cannot draw any meaningful isolines.

all result in significantly different performance distributions. Furthermore, evaluating the steering algorithms solely on such synthetic paths may potentially create misleading or even faulty results since these fail to grasp the true nature of human locomotion. For example, as emphasized in Fig. 11, S2G outperforms S2C significantly in most synthetic settings, but performs similarly on real paths.

With the presented work, we confirmed the usefulness of testing RDW approaches on real user data. While validations through such studies have waned in recent years in favor of simulation through powerful tools, our paper offers evidence that simulated paths are not always fully representative of real human paths. Indeed, even just small discrepancies may distort RDW results to a large extent. Even if prerecorded user paths are used as input, simulation alone may result in a loss of reality to a certain extent, because simulation cannot conclusively model users' reactions to RDW.

With our results, we hope to inspire future research on the sensitivity of RDW to slightly altered data and the effect of differences between synthetically generated paths and real user recordings. We also highlight that human behavior specifically evoked by RDW needs more investigation to enhance simulation environments. First attempts in this direction exist. For example, Nguyen et al. studied gain compensation [22], which essentially describes humans adjusting to redirection gains after some exposure time. Finally, we note that previously recorded user paths may not sufficiently qualify for pure simulation-based validation, because of the influence of online RDW on a user's perception and behavior. Since this potential change in behavior may lead to different performance distributions due to the sensitivity of RDW approaches, further research needs to investigate the conditions under which real paths can be adopted for simulation.

Alternatively, the set of commonly used synthetically generated user paths should be extended and reorganized considering already established human locomotion models and more realistic path segments should be added. For this goal, Fink et al. already derived second-order differential equations describing human walking behavior [10], while Arechavaleta et al. discussed potential approaches employing a system of differential equations with a cost function [1, 2]. Later, Cirio et al. [7] introduced Euler integration for continuous positional updates, and Zank & Kunz [38, 39] created a set of expected virtual trajectories based on points of interests in the VE. But eventually, the question remains, how well simulation-based results, even with drastically improved input data (may it be synthetic or

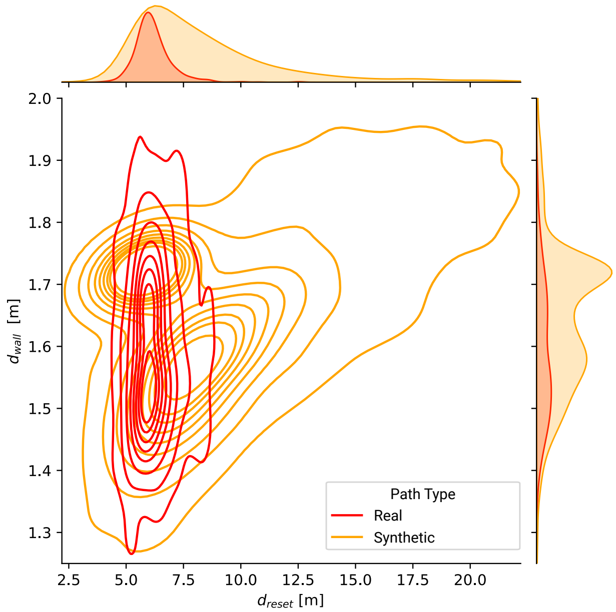


Figure 10: Performance comparison between path types. We show the results over the exemplary subset of simulations with **R2G** in a $6\,\mathrm{m} \times 6\,\mathrm{m}$ tracking space using **S2G** for steering. Synthetic represents all types of synthetically generated paths combined.

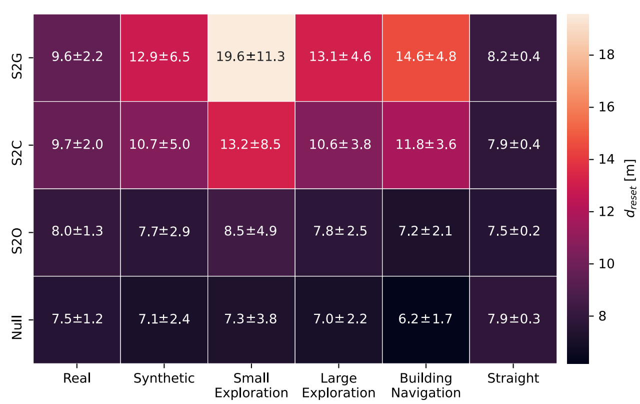


Figure 11: Mean distance between resets d_{reset} broken down by steering algorithm and path type. The first two columns describe the direct comparison between real and synthetic user paths, whereas synthetic paths are further differentiated towards the right. We show the mean over the subset of simulations with R2G or R2C resets in the $8\,\mathrm{m} \times 8\,\mathrm{m}$ tracking space.

real), can be extrapolated to real users without rigorous testing.

Additionally, the effect of a reset procedure requires more in-depth investigation in future work. In our simulations, we assumed resets to be conducted perfectly and without veering, strafing or other disturbances. In reality, however, we observed users' tendency to reset uncleanly in various studies. Some users even stumbled slightly, which may cause another considerable source of influence on the redirection. This raises further questions on how such unclean behavior can be modeled more or less realistically in simulation, for example by inducing random perturbations during resets.

Finally, we hope that through our analysis in this paper, we convincingly demonstrated the importance of rigorous testing of newly proposed steering algorithms since RDW is such a sensitive process. Even though simulations may seem promising at first, inferring that steering algorithms' performances translate to real scenarios should be considered critically and an empirical validation should always be contained in the development pipeline for new steering algorithms.

References

[1] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. The nonholonomic nature of human locomotion: a modeling study. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pp. 158–163. IEEE, 2006.

[2] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. An optimality principle governing human walking. *IEEE Transactions on Robotics*, 24(1):5–14, 2008.

[3] M. Azmandian. *Design and evaluation of adaptive redirected walking systems*. PhD thesis, University of Southern California, 2018.

[4] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma. The redirected walking toolkit: a unified development platform for exploring large virtual environments. In *Everyday Virtual Reality (WEVR), 2016 IEEE 2nd Workshop on*, pp. 9–14. IEEE, 2016.

[5] 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.

[6] Britannica The Editors of Encyclopaedia. Chaos theory. Encyclopedia Britannica. Online, accessed: 28-May-2021.

[7] G. Cirio, A.-H. Olivier, M. Marchal, and J. Pettre. Kinematic evaluation of virtual walking trajectories. *IEEE transactions on visualization and computer graphics*, 19(4):671–680, 2013.

[8] T. Dong, X. Chen, Y. Song, W. Ying, and J. Fan. Dynamic artificial potential fields for multi-user redirected walking. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 146–154. IEEE, 2020.

[9] D. Engel, C. Curio, L. Tcheang, B. Mohler, and H. H. Bülthoff. A psychophysically calibrated controller for navigating through large environments in a limited free-walking space. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pp. 157–164. ACM, 2008.

[10] P. W. Fink, P. S. Foo, and W. H. Warren. Obstacle avoidance during walking in real and virtual environments. *ACM Transactions on Applied Perception (TAP)*, 4(1):2–es, 2007.

[11] J. H. Friedman and L. C. Rafsky. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics*, pp. 697–717, 1979.

[12] D. Harris, M. Wilson, and S. Vine. Development and validation of a simulation workload measure: the simulation task load index (sim-tlx). *Virtual Reality*, 24(4):557–566, 2020.

[13] C. Hirt, M. Zank, and A. Kunz. Prewap: Predictive redirected walking using artificial potential fields. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 976–977. IEEE, 2019.

[14] 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.

[15] O. Janeh, G. Bruder, F. Steinicke, A. Gulberti, and M. Poetter-Nerger. Analyses of gait parameters of younger & older adults during (non-) isometric virtual walking. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[16] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[17] E. Langbehn and F. Steinicke. Redirected walking in virtual reality, 2018.

[18] D.-Y. Lee, Y.-H. Cho, and I.-K. Lee. Real-time optimal planning for redirected walking using deep q-learning. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 63–71. IEEE, 2019.

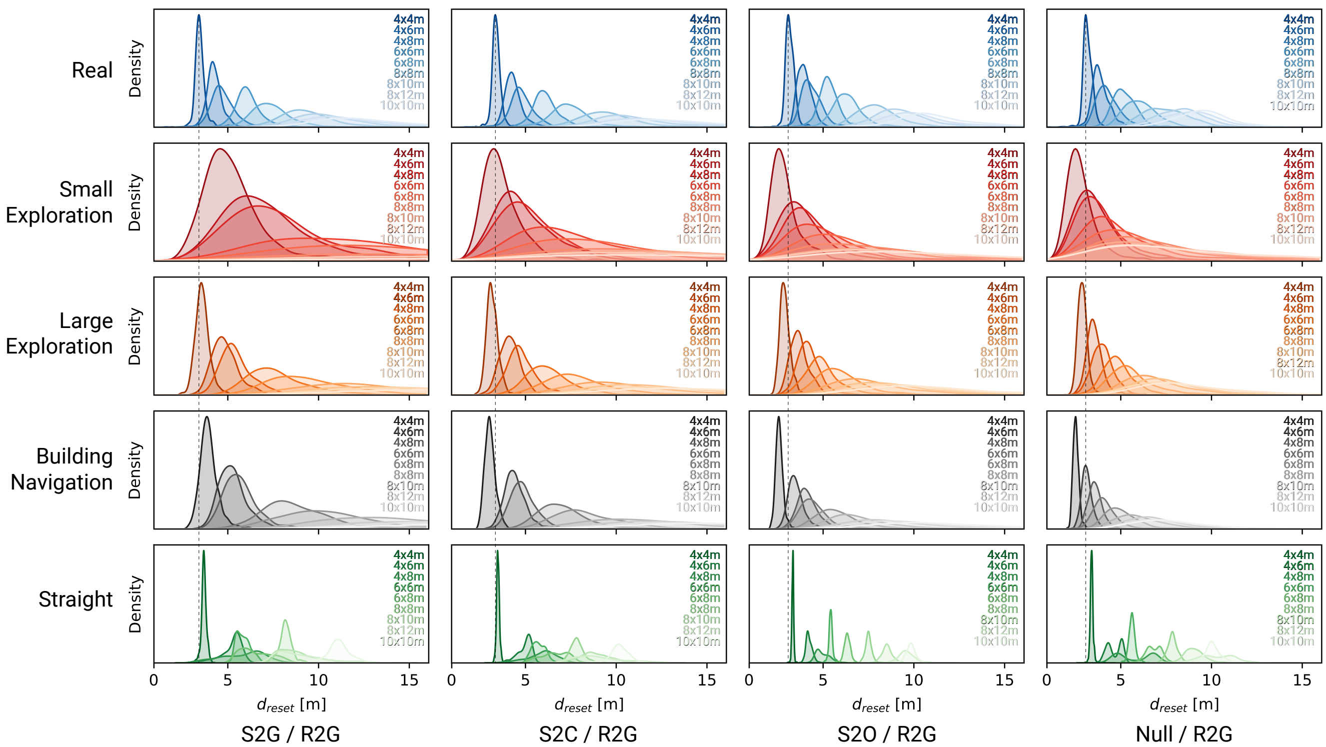


Figure 12: Simulation results for all steering algorithms, path types, and tracking spaces. Even though all synthetic types show the expected characteristic of improved redirection performance with increasing tracking space, they still do not strictly adhere to the real user paths.

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

[20] J. Messinger, E. Hodgson, and E. R. Bachmann. Effects of tracking area shape and size on artificial potential field redirected walking. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 72–80. IEEE, 2019.

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

[22] A. Nguyen, F. Cervellati, and A. Kunz. Gain compensation in redirected walking. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, p. 20. ACM, 2017.

[23] A. Nguyen, Y. Rothacher, B. Lenggenhager, P. Brugger, and A. Kunz. Individual differences and impact of gender on curvature redirection thresholds. In *Proceedings of the 15th ACM Symposium on Applied Perception*, p. 5. ACM, 2018.

[24] 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.

[25] T. C. Peck, H. Fuchs, and M. C. Whitton. Evaluation of reorientation techniques and distractors for walking in large virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):383–394, 2009.

[26] T. C. Peck, H. Fuchs, and M. C. Whitton. The design and evaluation of a large-scale real-walking locomotion interface. *IEEE transactions on visualization and computer graphics*, 18(7):1053–1067, 2012.

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

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

[29] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe. Analyses of human sensitivity to redirected walking. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pp. 149–156. ACM, 2008.

[30] R. R. Strauss, R. Ramanujan, A. Becker, and T. C. Peck. A steering algorithm for redirected walking using reinforcement learning. *IEEE transactions on visualization and computer graphics*, 26(5):1955–1963, 2020.

[31] E. A. Suma, G. Bruder, F. Steinicke, D. M. Krum, and M. Bolas. A taxonomy for deploying redirection techniques in immersive virtual environments. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pp. 43–46. IEEE, 2012.

[32] J. Thomas and E. S. Rosenberg. A general reactive algorithm for redirected walking using artificial potential functions. pp. 56–62. IEEE, Osaka, Japan, 2019. doi: 10.1109/VR.2019.8797983

[33] J. Thomas and E. S. Rosenberg. Reactive alignment of virtual and physical environments using redirected walking. pp. 317–323. IEEE, Atlanta, GA, USA, 2020. doi: 10.1109/VRW50115.2020.00071

[34] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr. Walking¿ walking-in-place¿ flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 359–364. ACM Press/Addison-Wesley Publishing Co., 1999.

[35] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. Rieser, and B. Bodenheimer. Exploring large virtual environments with an hmd when physical space is limited. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization*, APGV '07, pp. 41–48. ACM, New York, NY, USA, 2007. doi: 10.1145/1272582.1272590

[36] N. L. Williams, A. Bera, and D. Manocha. Arc: Alignment-based redirection controller for redirected walking in complex environments. *arXiv preprint arXiv:2101.04912*, 2021.

[37] 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 *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 313–319. IEEE, 2018.

[38] M. Zank and A. Kunz. Using locomotion models for estimating walking targets in immersive virtual environments. In *Cyberworlds (CW), 2015 International Conference on*, pp. 229–236. IEEE, 2015.

[39] M. Zank and A. Kunz. Where are you going? using human locomotion models for target estimation. *The Visual Computer*, 32(10):1323–1335, 2016.

[40] 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.