Comparing Four Approaches to Generalized Redirected Walking: Simulation and Live User Data

Eric Hodgson and Eric Bachmann, *Member, IEEE*

Abstract—Redirected walking algorithms imperceptibly rotate a virtual scene and scale movements to guide users of immersive virtual environment systems away from tracking area boundaries. These distortions ideally permit users to explore large and potentially unbounded virtual worlds while walking naturally through a physically limited space. Estimates of the physical space required to perform effective redirected walking have been based largely on the ability of humans to perceive the distortions introduced by redirected walking and have not examined the impact the overall steering strategy used. This work compares four generalized redirected walking algorithms, including *Steer-to-Center*, *Steer-to-Orbit*, *Steer-to-Multiple-Targets* and *Steer-to-Multiple+Center*. Two experiments are presented based on simulated navigation as well as live-user navigation carried out in a large immersive virtual environment facility. Simulations were conducted with both synthetic paths and previously-logged user data. Primary comparison metrics include mean and maximum distances from the tracking area center for each algorithm, number of wall contacts, and mean rates of redirection. Results indicated that *Steer-to-Center* out-performed all other algorithms relative to these metrics. *Steer-to-Orbit* also performed well in some circumstances.

Index Terms—Redirected walking, virtual environments, navigation, human computer interaction, live users, simulation

1 Introduction

Immersive virtual environments (VEs) that incorporate wearable rendering and display systems allow users to navigate through virtual worlds in a natural manner. In such an immersive VE, a user's real world position and orientation are tracked within a specified area and then sent to the rendering system, which updates the view displayed in the user's Head Mounted Display (HMD). Normally, movement in the real world will result in identical movement within the VE; changing one's physical direction or speed of travel result in identical changes within the visual, virtual world. This correspondence between real and virtual movement allows the body of the user to function as an input device while supplying users with rich spatial-sensory feedback. Unlike navigation using arbitrary controls or intermediaries, virtual walking produces the same proprioceptive, inertial, and somatosensory cues that users experience while navigating in the real world. The result is often a greater sense of immersion or presence [1] and a lessened chance of becoming disoriented within the VE [2].

The obvious limitation to allowing immersed users to walk naturally is that the area over which user position and orientation can be tracked within the real world is limited. It will not be possible to travel naturally through VEs larger than the tracking area with a 1:1 correspondence between real and virtual movements. Thus, users must be restricted to very small VEs or must use an alternative method of navigation, such as using a joystick, walking in place [3], using an omni-directional treadmill [4], or employing similarly specialized hardware [5]. While these types of methods can enable large-scale virtual navigation in small physical spaces, they also decrease the fidelity of spatial-perceptual feedback to the user and may require a large financial investment.

In recent years the technique of redirected walking (RDW) [6] [7] has begun to emerge as a viable means of allowing users to travel naturally through virtual worlds of unlimited size. When using RDW, navigation through unbounded virtual worlds is accomplished by subtly distorting the correspondence between user movements and visual consequences in order to steer the user within the physical space. While the real and virtual movements are no longer mapped with 1:1 fidelity, the differences are ideally small enough to fall below the level of human perceptual sensitivity. For example, it is well known that people can and will walk in circles while attempting to walk in a straight line in sparse (e.g., a desert) or confusing environments (e.g., a forest) [8]. Steering users with RDW is accomplished by imperceptibly rotating the virtual scene about the viewpoint of the user such that she will unconsciously veer in the real world while pursuing a visual goal. In contrast to work such as that of Interrante et al. [9] in which user movement in the direction of travel is scaled to the point where one step in the real world may result in a virtual movement equivalent to up to seven steps, in RDW, both translations and rotations within the VE are scaled imperceptibly to induce smaller or larger physical changes. When used in a systematic manner, these techniques can be used to guide user's physical movements into open space and away from tracking area boundaries, obstacles, or even other users.

The magnitudes of the discrepancies between real and virtual motion introduced by RDW may vary, but should generally be limited by the abilities of humans to perceive them. The introduction of perceptible discrepancies can be distracting and break the sense of presence, or lead to increased cyber sickness. Thus, a considerable amount of research has focused on determining the threshold at which humans will detect discrepancies associated with RDW [6] [10] [11] [12]. These thresholds have then been extrapolated and used as a basis for estimating the smallest possible area in which RDW could effectively be used to simulate VEs of arbitrary size. These estimates are discussed more fully in Section 2, but the minimum room dimensions typically range from 40m [12] to 90m [6].

Whereas the limits of human perception are undoubtedly an important factor in determining the space required to use RDW effectively, there are a number of other relevant factors that have received less attention. For example, the required tracking area and the maximum rates of imperceptible steering are likely to be influenced by the nature of the VE (e.g., proximity of objects, amount of optic flow), the attention demands of the user task (e.g., focus on navigation or some other goal), perceptual adaptation (e.g., duration of sessions, repeated sessions), individual differences

• *Eric Hodgson is the director of the Smale Interactive Visualization Center at Miami University, Ohio. E-mail: eric.hodgson@miamiOH.edu.*
• *Eric Bachmann is an associate professor in Computer Science and Software Engineering and Director of the HIVE at Miami University, Ohio. E-mail: eric.bachmann@miamiOH.edu.*

Manuscript received 13 September 2012; accepted 10 January 2013; posted online 16 March 2013; mailed on 16 May 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

among users, and of course the specific redirected walking algorithm or strategy in use. This last factor is the focus of the present work.

In his original treatise on RDW [6], Razzaque's proposed three general methods for steering users, dubbed *Steer-to-Center*, *Steer-to-Orbit*, and *Steer-to-Multiple Targets*. Although some other specialized algorithms exist, these three adequately represent the approaches commonly seen in the RDW literature.

While nearly all RDW algorithms use the same steering techniques (e.g., bending straight paths, altering user rotations, etc.), they differ on the high-level strategy of *where* a user is to be steered. These differences would be expected to require different amounts of physical space to function well, thus making some algorithms more effective than others. The present work examines how the choice of algorithms alters the effectiveness of RDW and, by extension, the comparative space requirements.

To this end, the performance of Razzaque's three algorithms was evaluated in both simulation and with human subjects engaged in live navigation. Algorithms were evaluated on the typical distances (mean, max) within which they constrained users, the mean rate of redirection, and the number of times live users came near the laboratory walls. Additionally, some simulation studies used synthetic paths to investigate best- and worst-case scenarios for RDW. Because *Steer-to-Multiple Targets* could be implemented in many different ways, two variations were evaluated.

Contributions of this work include the following:

- Implementation of multiple generalized RDW algorithms with live users
- Quantitative and qualitative comparisons of four generalized RDW algorithms
- Realistic evaluation of the space requirements of each algorithm for users engaged in a simple search task
- Theoretical results that demonstrate when generalized RDW algorithms can be expected to fail or work poorly

The remainder of this paper is organized as follows. Section 2 describes related work with a focus on studies that describe or compare RDW algorithms, and on those that have attempted to establish the minimum required area for RDW. Section 3 provides a detailed description of the algorithms implemented. Section 4 continues with a description of the simulations and real world experiments for which algorithm comparison data are presented. The final section summarizes conclusions that can be drawn from the work.

2 BACKGROUND

In [6] [7], Razzaque presented seminal work in which the perceptual underpinnings of RDW were examined, and basic steering techniques were described. Razzaque did not have a tracking area of sufficient size to fully implement or study generalized RDW, but he did provide a basic proof-of-concept implementation for Steer-to-Center. With this implementation, the dampening meant to prevent sudden changes in steering direction inadvertently disabled steering for users traveling away from the center of the tracking area – a period when RDW is needed most. Although this limitation can and has been corrected [14], it led Razzaque to propose two alternative strategies, including steering users to multiple waypoints or onto an orbit around the center.

Razzaque performed a series of experiments in order to determine the threshold of imperceptible rotation rates. The data from these experiments indicated that the worst case detection threshold was a rate of 1.0°/s. Based on this value and assuming a normal walking speed along a long straight path, it was conjectured that the area required to perform RDW would have a radius of at least 45m. Since the detection threshold of 1.0°/s was established for walks that started from a stop and were relatively short in duration, it was hypothesized that it might be possible to increase the rate of rotation over time due to podokinetic after-rotation effects (PKAR), with the rate eventually reaching 13 - 18°/s. Using these values it was estimated that an area roughly 30m x 30m (15m radius) would be sufficient to perform imperceptible rotations associated with RDW once a user adapted.

Steinicke and his colleagues have done a great deal of work related to determining perception thresholds for undistracted users [9] [11] [12]. Like Razzaque, they have estimated minimum tracking area sizes by determining and extrapolating these thresholds. In their work, they have differentiated the amount of rotation distortion that can be applied while users are walking and while they are turning their head while standing in place, and the amount of translational distortion that can be applied while walking. Importantly, they have also identified asymmetries in the amount of steering that can be implemented at a given moment depending on whether the redirection complements or counteracts the user's current movements. In [12], it was determined that distances could be decreased in the physical world by up to 14% or increased by 26% relative to the visual distance travelled. Likewise, turns in place (head yaws) can be compressed by 20% and dilated by 49%. Finally, data indicated that users walking in a straight line could be made to follow a circular arc with a radius of 22 meters. Based on these results, it was estimated that a tracking area would have to measure 40m x 40m in order to simulate arbitrarily large VEs.

The studies by Razzaque, Steinicke, and others are helpful in estimating the size of a tracking space required to implement RDW and for providing reasonable steering parameters. It may be best to consider their findings as worst-case baselines for detectable thresholds and size requirements. None consider the cases of users who were actively engaged in a primary task in a VE, were naive to the idea of RDW, or who might stop, start, look around, or change their direction and speed of travel at any time. Participants in such studies were generally told where and how far to walk, and were explicitly tasked with detecting the presence of RDW.

There has been some work done to evaluate the relative performance of different steering algorithms. For example, Field and Vamplew [13] presented a simulation study comparing Razzaque's [6] Steer-to-Center algorithm and two algorithms of their own design called the *Large Circle* and *Small Circle*. Whereas Steer-to-Center is generalized and requires no knowledge of the VE or the task being performed by the user, the Large- and Small-Circle algorithms were designed to steer based on foreknowledge of when, where, and in what direction users would turn. In the study, realistic variations in travel, such as occasionally pausing or varying the speed of one's translations or rotations were not modeled. Algorithm comparisons were made based on the furthest distance from the tracking area center that the user travelled and the percent of time the simulated users were within a tracking area of arbitrary size. In the study, it was found that Steer-to-Center outperformed the Large-Circle algorithm when no knowledge of the user's future movement was available. When provided with forecasted movements, both the Large- and Small-Circle algorithms outperformed Steer-to-Center. Maximum distances from the center of the tracking area ranged from a worst case of 31.8m to a best case of 17.3m.

In [15], the authors presented what is believed to be the first demonstration of generalized RDW in which real users navigated through a large virtual world while physically walking in a moderately sized tracking area, measuring 25m x 45m. The primary focus of the work was to study the impact of RDW on users' spatial memory of the virtual world. Data were obtained through experiments in which users were tasked with a dual search task in a large virtual forest. Users performed these tasks under both normal and RDW conditions, using a Steer-to-Center algorithm. Though the maximum steering rates used were higher than those suggested by Steinicke et al. [12] or Field and Vamplew [13], data indicated that the use of RDW had no impact on spatial memory, did not result in an increase of simulator sickness, and was generally unnoticed. While redirected users did reach the bounds of the tracking space occasionally, this work suggested that RDW might be reasonably effective in spaces that are notably smaller than originally estimated and that users who are cognitively engaged in a primary task can be redirected at higher rates.

3 Generalized Redirected Walking Algorithms

Four algorithms are compared in this work: *Steer-to-Center*, *Steer-to-Orbit*, and *Steer-to-Multiple Targets*, and *Steer-to-Multiple+Center*. All are based on methods suggested in Razzaque's [6] foundational study, with the latter two algorithms being variations of the Steer-to-Multiple that do or do not include the center of the tracking area as a steering target. Other potential approaches that require specific user movements or some knowledge of a user's likely future path, are not considered. Instead, all algorithms herein are generalized and merely react to changing user movements in a way that tends to direct the user towards a specific steering target or path.

The parameters used to implement the steering and guide users to the current steering target were the same for all algorithms, and used the rotational and translational gains suggested by [12] and arced straight paths onto a minimal radius of 7.5m as used in [15]. These parameters were previously shown not to adversely impact the user or make RDW noticeable. The steering parameters were constant across all algorithms, as was the task for real users and the user-agent trajectories underlying the simulation. Thus, the only differences among the four algorithms lay in *where* the user was being steered. Using different steering parameters will obviously change the size requirements of RDW, with stronger or weaker steering needing smaller or larger spaces, respectively. To illustrate how the results below might be expected to differ with alternative RDW parameters, simulated data are also presented using a turn radius of 22m instead of 7.5m, which is the turn radius suggested to be optimal in [12].

The basic operation of each algorithm is illustrated in Fig. 1. Algorithms consider the current physical location, linear velocity, and the angular velocity of the user. Based on this input, the algorithms determine whether or not the current steering target should be updated and, if necessary, generates or selects a new target.

Once the current steering target has been evaluated and possibly updated, the maximum rotation that can be applied is determined based on the state of the user. As suggested by Razzaque [6], three different rates of rotation are considered, with the highest available rate prevailing. The three rates considered are a baseline rotation rate, a linear movement rotation rate, and an angular rotation rate.

The baseline rotation rate is associated with a user state in which there is little or no movement presently occurring. Given that humans have a certain amount of natural postural sway and are incapable of standing completely still, the application of baseline redirection is intended to inject small amounts of steering during postural sway. Here, the maximum value for this rate is 0.5º/s. Thus, the maximum rotation that can be applied from baseline redirection in any given rendering frame is

$$baselineRotationRate \times \Delta t$$

where Δt is the sampling interval.

If a user's linear movement exceeds a threshold of 0.2m/s, they are considered to be no longer standing still, and a linear movement rotation rate is calculated. It is intended that the application of linear movement rotation rate would cause a user following a long straight visual path to travel on an arc with a specified radius. The value for the specified radius used in this work is 7.5m. The additional linear movement rotation that could be applied during an update is

$$360 \times \frac{linearVelocity}{2 \times \pi \times 7.5} \times \Delta t$$

Because the linear movement rotation rate is based on a fixed radius, the visual rotation (in º/s) varies with the speed of walking. Unlike [16], the above calculation will cause the rate of rotation due to linear movement to increase until a limit of 15º/s is reached. The

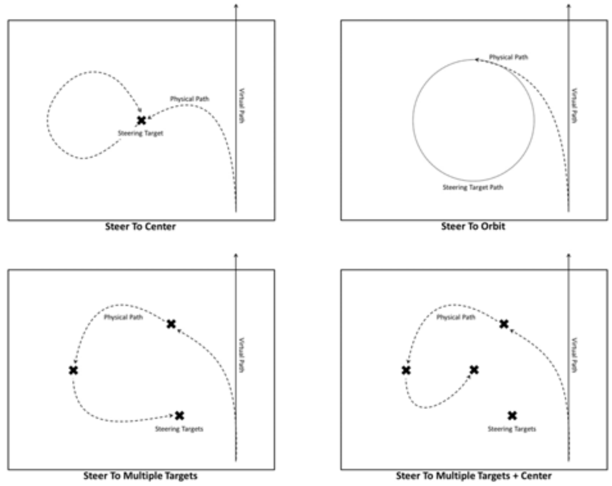

Fig. 1. Illustrations of the four RDW algorithms tested. Users were steered towards particular targets or onto a target path.

incorporation of the refinement described in [16] could be expected to produce steering instructions which are less perceptible and will be considered in future implementations.

The angular rotation rate is associated with user head yaws that are greater than a threshold value of 1.5º/s. It functions by scaling the physical head rotation that is taking place using gains. Gain values less than one cause the user to turn further in the physical world than in the VE, while gains greater than one result in smaller physical turns. As suggested by Steinicke, et al [12], the gain magnitude depends on whether the steering rotation is with or against the rotation of the head, and is set to either 1.30 or 0.85, respectively. As with linear movement rotations, angular rotation rates are limited so as not to exceed 30º/s. The additional angular rotation that could be applied during an update is given by

$$angularVelocity \times angularRateScalingFactor \times \Delta t$$

After the rotations given by equations above are calculated, the largest rotation is selected for application. In all algorithms, however, the selected rotation may be dampened based on a number of factors. When the current steering target is within a pre-defined dampening range relative to the heading of the user, the rotation to inject is dampened by

$$selectedRotation \times \sin\left(bearingToTarget \times \frac{90}{dampeningRange}\right)$$

Similarly, if the user is within a 1.25m distance threshold of the steering target, the selected rotation to inject is dampened by

$$selectedRotation \times \frac{distanceToTarget}{distanceThreshold}$$

Both of these operations are performed in order to suppress rapid directional changes to the applied steering when the users are walking directly towards the steering target or have just passed it. Once the appropriate magnitude of steering has been determined for a given moment, the desired direction of steering (+/-) is determined.

Finally, before applying the calculated changes to a VE, a smoothing function is applied. This prevents abrupt changes in the applied rotation, which could be perceived by users. The final applied rate is

$$(1-s) \times previousAppliedrotation + s \times rotationToInject$$

where s is a smoothing factor set to a value between 0 and 1. The value used for s in the experimental work presented below is 0.125.

3.1 Algorithms

As noted above, the algorithms differ only in how and when steering targets are selected. Details of how each algorithm selects the current target are given below.

3.1.1 Steer-to-Center

The Steer-to-Center algorithm used in the present experiments was identical to that used in [15]. It is a modified version of Razzaque's basic algorithm [6], but designed to overcome limitations of the original. Specifically, Razzaque avoided oscillations in the direction of steering when users were walking directly towards or away from the center steering target by multiplying the maximum desired steering value by the sine of a user's heading in relation to the target. This caused maximal steering to be injected at ±90º and increasingly dampened steering as the bearing-to-center approached 0º and 180º. While oscillations in steering direction are potentially noticeable and distracting to users, it is disadvantageous to dampen steering for users who are traveling away from the center and towards a wall – the precise time RDW is needed most. In the present implementation, this problem is alleviated by generating a temporary steering target whenever the bearing-to-center is greater than 160º. Temporary targets are set at a distance of 4m from the user, 90º from her current heading, in whichever direction will return the user to the tracking area center soonest. The temporary target is abandoned as soon as the user's bearing-to-center is again less than 160º, and steering instructions again guide the user exclusively towards the center.

3.1.2 Steer-to-Multiple

The Steer-to-Multiple algorithm utilizes three predefined steering targets that were equally distributed around the room's center. Each target was 5m from the center of the room, and separated by 120° (i.e., at headings of 90°, 210°, 330°). Users are steered toward the steering target that is closest to being in front of them, so as to complement their naturally-chosen course. The current steering target is evaluated on every update cycle in order to determine if a new steering target should be selected. That is, users are not left to continue towards a prior steering target if an alternative is determined to be more appropriate.

3.1.3 Steer-to-Multiple+Center

The Steer-to-Multiple+Center algorithm uses the same three targets as the Steer-to-Multiple, but also includes the tracking area's center as a fourth target. The implementation described in the previous paragraph used three equally distributed waypoints at a fixed radius. This could be conceived as a very basic form of the Steer-to-Orbit, as more points chosen at the same radius would eventually generate a circle around the center. In particular, users walking a long, straight line may circle around to each of the waypoints in order. The inclusion of the room's central point differentiates this algorithm from the Steer-to-Orbit approach described below by allowing users to be directed through the interior. The selection of steering targets follows the same logic as above; users are steered toward whichever target is closest to being in front of them.

3.1.4 Steer-to-Orbit

The previous algorithms rely on steering to specific, pre-defined points in space. Steer-to-orbit, on the other hand, continuously calculates new steering targets that are ahead of the user on a desired path through space. For the present study, an ideal orbit of 5m is used, and a point along the orbit is selected based on the user's current position and orientation. For users outside the ideal orbit, two lines are generated that intersect the user and are tangent to the orbit. The two tangent points on the orbit are then considered as potential steering targets. For users inside the ideal orbit, two potential steering targets are also generated on the orbit, 60º to either side of a line that passes from the center of the orbit through the user. In either case, the candidate steering target requiring the smaller bearing change is then selected as the new steering point.

In this work, the radius of the orbit (5m) is smaller than the maximum curvature used for redirection (7.5m). Thus, it is not possible for users to actually follow the target orbit exactly, but more likely they will follow an orbit equal to the maximum curvature. This difference becomes more important if users are walking in long, straight paths that will tend to keep users on an orbit. For users who may be spontaneously changing course, the smaller ideal orbit allows wayward users to be returned closer to the tracking space's center before falling back into the orbit.

4 Experimental Evaluation

We conducted two experiments designed to gauge the performance characteristics of the various steering algorithms detailed above. The first experiment consisted of a series of simulations. Input data for the simulations came from two sources. The first source was a pool of actual navigation data logged from the paths of participants in a previous behavioral study. The second source was a set of procedurally generated paths that represented specific patterns, such as a long straight path, zigzag, or figure-eight. These latter paths were not intended to be realistic, but rather tested specific idealized scenarios. Each logged and synthetic path was fed through all four candidate RDW algorithms to determine which algorithm would constrain users to the smallest physical area.

The second experiment extended beyond simulations and compared the performance of each algorithm with live users. In the experiment, participants completed a series of search tasks in a large VE while being steered by each algorithm. The primary goal of these real world experiments was to verify that the results obtained in simulation had external validity.

4.1 Simulations

4.1.1 Method

The simulations were designed to take sets of visual paths through a VE and compute where a user would travel physically while being redirected by each of the four algorithms being considered. Visual paths were drawn from a bank of previously recorded data from the movement of live users in [15] and from a set of synthetic paths representing certain idealized patterns, as described below. In sum, there were 7 synthetic paths and 71 paths recorded from previous users. The latter represented 4 hours and 44 min of actual navigation.

Live-user data was captured from users who were participating in a free-exploration search task in a large virtual forest, as in [15]. These users were physically walking within Miami University's 25m x 45m HIVE facility [17] either under normal conditions or while being redirected to allow for large scale navigation. Thus, one group was restricted to travel no more than 25.74m both physically and virtually from the center of the tracking space (n=35), while the other group could wander throughout much larger virtual area (n=36). Initial analyses of the data presented below considered these groups separately, but because there were no meaningful effects or interactions with the data source, they were considered together for the final analysis. For logged paths, users' visual position and orientation within the VE had been recorded at 30 - 60 Hz along with a time stamp to ensure accurate velocity estimation during simulation.

Synthetic trajectories were simulated to move at speeds typical of an immersed user, specifically 1m/s. Turns were set to 90º/s. The generated paths were not random or designed to mimic human walking, but rather conformed to specific shapes that provide ideal conditions for redirection (e.g., a long, straight path) or that were expected to provide difficult conditions for redirection (e.g., a sinusoidal path).

These two data sets were intended to address different questions. While synthetic paths are excellent for testing certain scenarios, it is difficult to ascertain the overall performance of an algorithm with such paths. Indeed, it is difficult or impossible to procedurally capture the pseudo-random nature of human navigation or the individual differences among users. Users may turn in different directions at different times and velocities, either pivoting or following a gentle arc; users may vary their walking speed, stop and start, walk backwards, side-step, or even stand in place and gaze about for an extended period of time. All of these behaviors can be captured in the logged data set, while a pseudo-random synthetic path might only approximate "realistic" navigation for some prototypical user. As such, the analyses below evaluate algorithm performance based solely upon simulations using logged paths, while synthetic paths were used as case studies to illuminate specific places where algorithms excelled or struggled.

4.1.2 Results and Discussion

Logged user data. A set of univariate analysis of variance (ANOVAs) were conducted to determine the effect of redirection algorithm (Steer-to-Center, Steer-to-Orbit, Steer-to-Multiple, or Steer-to-Multiple+Center) on three separate metrics. Specifically, algorithms were compared on the *mean physical distance* from the center of the tracking space at which they kept users, the *maximum physical distance* from the center that an average user reached, and the *mean unsigned rate of redirection* during navigation.

It should be noted that the mean and maximal distances reported below are directly related to specific settings of the redirection algorithms (e.g., a minimum turn radius of 7.5 m). Because these settings were constant across all four algorithms, differences among the algorithms are valid and useful in comparing their relative performance. However, maximum distances observed here would necessarily vary with changing steering parameters; RDW settings can easily be adjusted to constrain users to larger or smaller spaces. To make a more comprehensive assessment of the space requirements of each algorithm, simulations were rerun using a much larger minimum turn radius of 22m, as suggested by [12]. These results are not discussed in detail as the pattern of results is unchanged, but they are displayed in data figures for reference.

Mean Physical Distance. Mean distances for each algorithm are shown in Fig. 2, and exhibited a significant main effect of redirection algorithm ($F(3, 280) = 12.18$, $p < .001$). A series of planned contrasts indicated that the omnibus effect of algorithm was primarily driven by the difference between Steer-to-Center and the other three algorithms ($F(1, 280) = 36.24$, $p < .001$). There were no differences between the other conditions (all F's < 1). Steer-to-Center kept users, on average, 6.41 m ± 0.44 (for a 95% confidence interval) from the center of the tracking space. Steer to multiple targets (8.10 m ± 0.48), multiple targets including the center (8.00 m ± 0.51), and orbit (7.93 m ± 0.39) all allowed users to stray consistently farther. These patterns are clearly visible in Fig. 2, and do not differ if a larger turn radius is used. It is also noteworthy that a 22m turn radius (293.33% larger) results in a relatively small increase in mean distance. This speaks to the relatively strong influence of rotational gains that are implemented as users spontaneously turned or looked around. Because participants rarely walked in straight lines for long periods, the veering rate had only a modest influence.

While Steer-to-Center was superior in terms of simple averages, it is informative to consider the entire distribution of physical distances for each algorithm. These distributions are shown overlaid in Fig. 3. The advantage of Steer-to-Center is readily apparent, as it boasts a large area of its distribution less than 4 m from the center – a feature not shared by any other algorithm. However, Steer-to-Center also yielded the broadest distribution of distances. Perhaps more noteworthy is that Steer-to-Orbit generally contained users to a much narrower region of space, and peaked between 5m (the orbit defined in the algorithm) and 7.5 m (the maximum turn radius allowed by the algorithm). Thus, Steer-to-Orbit was highly successful at keeping users in the targeted region, and may be a worthy alternative to Steer-to-Center despite its slightly higher average. Neither variation of Steer-to-Multiple-Targets exhibited any such advantage.

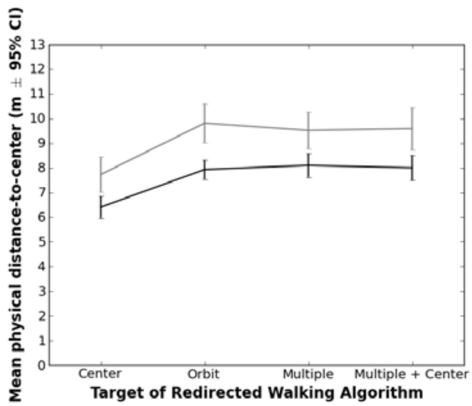

Fig. 2. Mean physical distance of simulated users steered with each algorithm. The darker line shows the results when using a 7.5m turn radius; the lighter line shows a 22m turn radius.

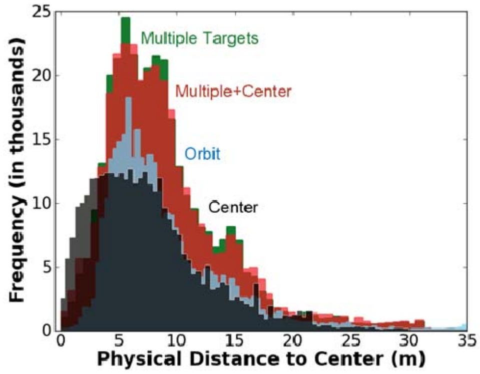

Fig. 3. Distribution of physical distances for simulated users steered with each algorithm.

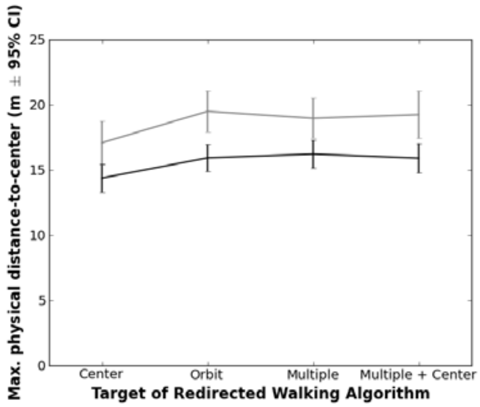

Fig. 4. Maximum physical distance of simulated users with each algorithm. The darker line shows the results when using a 7.5m turn radius; the lighter line shows a 22m turn radius.

Maximum physical distance. Maximum distances are shown in Fig. 4. The influence of algorithm type was similar, but somewhat weaker as the main effect was marginally non-significant ($F(3, 280) = 2.34$, $p = .073$). Planned contrasts, however, again showed a significant difference between Steer-to-Center and the other three algorithms ($F(1, 280) = 6.83$, $p < .01$), which did not differ from each other (all F's < 1). Steer-to-Center contained simulated users within an average maximal radius of 14.38 m ± 1.09, while Steer-to-Orbit (15.93 m ± 1.04), Steer-to-Multiple-Targets (16.22 m ± 1.08), and Steer-to-Multiple+Center (15.91 m ± 1.14) all allowed users to

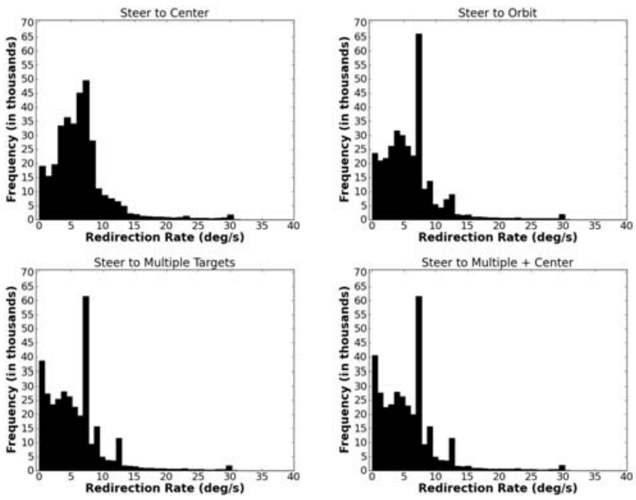

Fig. 5. Distribution of absolute rates of redirection for simulated users steered with each algorithm. Lower rates indicate less perceptual distortion, but less-effective redirection. Peaks near zero indicate higher incidence of dampened steering.

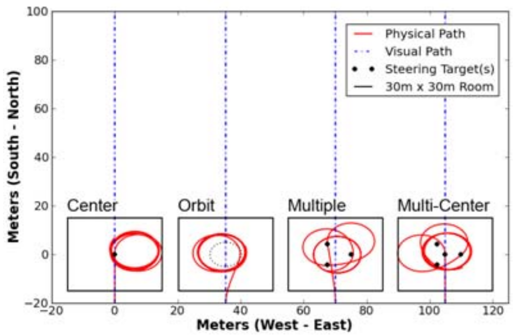

Fig. 6. Simulated physical paths redirected from long, straight paths by each algorithm.

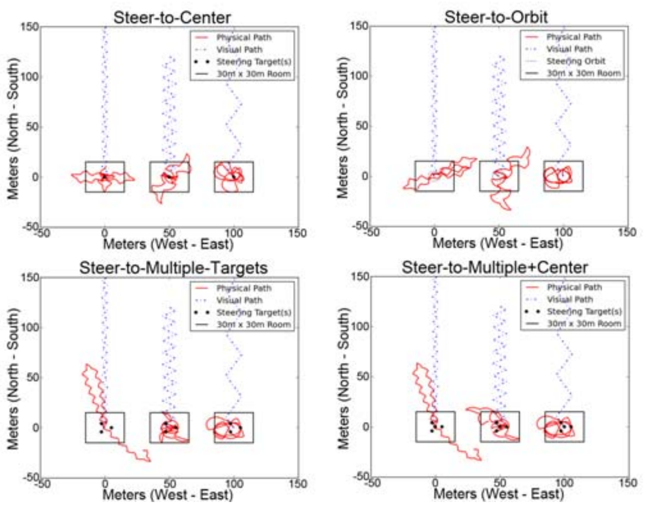

Fig. 7. Simulated physical paths redirected from small, medium, and large zig-zagged paths by each algorithm.

stray further from the center of the tracking space. It is interesting that Steer-to-Orbit did not perform better on this metric, given the relatively tighter distribution of distances noted above.

Assuming the same RDW settings (i.e., a turn radius of 7.5 m) and gains used in this simulation, a tracking space of roughly 30 m diameter would be necessary to completely contain users with Steer-to-Center while enabling infinite virtual navigation. As shown in Fig. 4, increasing the turn radius to 22m increases the minimally required area, although the increase is smaller than might be expected given the nearly three-fold increase in turn radius: a mere 5.42m of additional space would suffice.

Mean absolute steering rate. The distributions of absolute steering rates are illustrated in Fig. 5. The pattern of means generally showed the inverse pattern as the two distance metrics. That is, higher average rates led to lower average distances. The rate of redirection showed a significant main effect of steering algorithm ($F(3, 280) = 7.78$, $p < .001$). Steer-to-Center exhibited the highest average rate of redirection (6.11º/s ± 0.35). Planned contrasts indicated that Steer-to-Orbit was significantly lower than Steer-to-Center (5.60º/s ± 0.33; $F(1, 280) = 4.79$, $p < .05$), but also significantly higher than both of the Steer-to-Multiple-Targets variations ($F(1, 280) = 5.08$, $p < .05$). Steer-to-Multiple-Targets (5.18º/s ± 0.31) and Steer-to-Multiple+Center (5.12º/s ± 0.32) showed the lowest rates of redirection and did not differ from one another ($F < 1$). For reference, an immersed user walking a continual straight line at a typical speed of 0.8 – 1.0m/s while being redirected at the same rates used in these simulations would be turned at a rate of 6.11 – 7.64º/s, excluding any dampening.

Periods of dampening were not equally likely to occur under all algorithms. Indeed, the two variations of Steer-to-Multiple-Targets were constructed to always steer users toward whichever target was nearest their front, which increases the chance that dampening will occur. An evaluation of the distribution of redirection rates for each algorithm (see Fig. 5) reveals that these two yielded substantially more periods of near-zero steering than Steer-to-Center or Steer-to-Orbit, as evidenced by a large, secondary spike in the distribution around 0º/s. Steer-to-Center and Steer-to-Orbit steer users more or less continuously, while the Multiple-Targets algorithms often provided users with little course correction. It is possible that a different method could be employed for choosing a steering target that minimized these periods of dampening, such as selecting a steering target nearest 90º instead of in front of the user. The implementations here, however, were notably less effective than Steer-to-Center or Steer-to-Orbit, which are conceptually simpler.

Synthetic data. Synthetic paths were intended to provide a more qualitative source of data to compare the candidate algorithms and to identify specific situations that were problematic. Quantitative trends were in line with those reported above – albeit with less statistical power – and will not be discussed here for the sake of brevity. Instead, three case studies are presented that represent one relatively ideal scenario and two problematic situations.

Ideal scenario: A long, straight line. Data for this scenario were constructed by simulating a user starting 20m due South of the tracking space's center and traveling Northward in the VE at 0.8m/s for a distance of 120m. Upon reaching the end of the path, the simulated user turned around and returned to the starting point. The redirected physical paths calculated by each algorithm are shown in Fig. 6, and conform to what one might expect from each algorithm.

One interesting point to note is that Steer-to-Orbit required the least physical space of any of the algorithms in this scenario. In VEs that feature long, straight segments punctuated by 90º turns (e.g., hallways, roads), this algorithm may be an optimal choice. Users can be easily contained on a desired orbit while navigating straight segments and can be induced to simply change directions along the orbit with visually turns of 90º amplified to near 180º. In the logged data above, this was not the case; the VE permitted users to travel along arced paths and change direction at will. Practically speaking, it may be the case that different algorithms will be better suited to specific types of environments. It may even be advantageous to switch between algorithms dynamically depending on what the user is doing or what part of a VE she is in at the moment.

Problematic case 1: Frequent direction changes. Three sets of simulations were run in which the synthetic user followed a zigzagged path of varying segment lengths and turn angles. Specifically, the paths were either 6m with 90º turns, 12m with 120º turns, or 24m with 60º turns. The selection of these lengths and turn angles is somewhat arbitrary, and the particulars here are less important than the overall pattern of a user making large, alternating changes in directions with varying frequency.

The redirected physical path for each zigzag pattern and algorithm are shown in Fig. 7. While some algorithms fared somewhat better or worse, all four struggled to contain the small and medium-sized zigzagged paths. Because the synthetic user changed directions so frequently and with such magnitude (i.e., 90º or more in the two problematic conditions), redirection had little opportunity to adjust the user's path before promptly changing. For example, the user may have been steered leftward, rightward, leftward, rightward in rapid succession, with the net effect being little or no redirection. As the segments of the zigzagged paths lengthened, all algorithms were able to inject progressively more course correction and physical paths were contained to progressively smaller areas. This simulation suggests that frequent turning – especially in alternating directions – may actually decrease the effectiveness of steering by causing subsequent steering instructions to counteract earlier ones.

Problematic case 2: Arced visual paths. Fig. 8 illustrates three simulated virtual paths that follow a curved figure-eight of varying sizes – either 6m, 12m, or 24m in radius. Again, the particular paths chosen are of less importance than the overall pattern – users walking along arced paths of various curvatures. Pilot testing of live-user RDW had shown this to be a problematic scenario that merited a more detailed examination.

The medium-sized figure-eight clearly produced the worst redirected paths for all algorithms. Smaller turns were sometimes tight enough to overcome the curvature of redirection and thus change the direction of the user – though not always in the intended direction. Likewise, the largest curves were shallow enough that RDW could still exert some influence and generate a relatively typical physical path. A closer examination of the small and medium figure-eight paths, however, shows two distinct phases. When users are walking along a path that arcs in the same direction as the steering corrections, the resulting path is tighter and more efficient than typical redirection, as one might expect. However, when the user begins to arc in the opposite direction, the conflicting curvatures largely cancel each other out. The resulting physical path approaches a straight line, which allows the user to travel over a surprisingly large area – seemingly without correction.

In many cases, it is presumed that users will be walking along moderately straight visual path segments connected by turns of varying degree. In fact, the logic of many redirection approaches [6] [7] [11] [14], including the present implementation, separates walking and turning into discrete categories that are affected by different sets of gains and curvature settings. Additionally, many studies ask participants to walk in squares [10], zigzags [7], or other artificially straight paths [18], while in reality people often walk along curved, organic paths, make broad turns, and look around while still walking (see, e.g., Fig. 10.). Because redirection seeks to direct people onto an arc of a certain magnitude and direction, a user who is simultaneously attempting to walk an arced path of similar magnitude in the opposite direction can effectively counteract all steering. In this case, the small (6m) and medium (12m) figure-eights were too similar to the targeted turn radius of 7.5m, to the users' detriment. A useful feature of generalized RDW would be to detect whether a user is conforming to injected steering corrections or opposing them, and then react accordingly.

4.2 Live User Experiment

It is important to evaluate the quality of each redirection algorithm with real, live users who are actually being steered. Methodologically, it is also preferable to have a clean source of visual path data. In simulation, logged data was gleaned from users who were either not being redirected – and thus could not explore a large area – or were being redirected using a Steer-to-Center algorithm, which coincidentally fared best in simulation. Given that both sources of data produced the same results, and that the visual trajectory should be a function of an individual's intentions rather

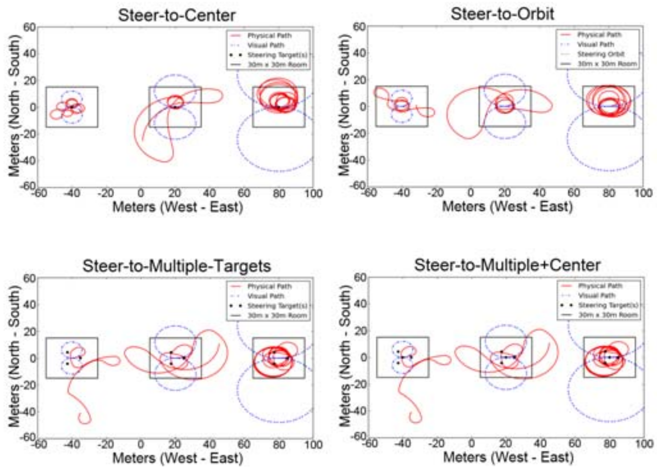

Fig. 8. Simulated physical paths redirected from small, medium, and large figure-eight paths by each algorithm.

than a by-product of the steering algorithm originally used, it is unlikely that the previous steering instructions would have confounded the simulation results. Nonetheless, this less-than-ideal shortcoming is avoided by having live users explore a VE separately with each algorithm.

In this experiment, we asked people to complete the same search task used to generate the logged data sets used in simulation. Participants wandered through a large virtual forest at will and collected as many colored posts as possible within 4 min, with different colored posts being worth different numbers of points. Participants completed four rounds of collection in an attempt to get higher point totals, while unknowingly being steered by a different redirection algorithm during each round.

The search task was chosen as an ideal test for generalized RDW due to its unpredictable and relatively unrestricted nature. It was not possible to anticipate what path a participant would take through the VE, how widely they might explore, or which parts of the VE they might choose to visit or ignore.

4.2.1 Method

Participants. 21 participants (12 female, 9 male) completed the experiment in exchange for credit in an introductory psychology course. The mean age of participants was 19.38 ± 0.49. Three participants reported cyber sickness after completing several blocks of testing and thus provided only partial data; they were allowed to recover and then excused from the remaining portion of the experiment. It is unclear whether the sickness was related to the use of redirected walking or to the general nature of being in an immersive VE simulation. All participants were naïve as to the aims of the experiment or the use of redirected walking.

Materials. Participants were tested in Miami University's 25m x 45m HIVE facility while wearing a backpack rendering system (see Fig. 9). The HIVE's 1,125 m^2 tracking area was crucial to implementing relatively imperceptible rates of redirection while maintaining relatively few wall collisions. Likewise, the backpack rendering system eliminated the need for extremely long HMD tethers. The backpack system consisted of a laptop (Alienware Aurora M9700) and video control unit that were affixed to an aluminium pack frame, and an HMD (Nvis SX60; 60º diagonal FOV, 100% binocular overlap) to display visual and auditory information to the wearer. Head orientation was tracked via an InertiaCube 2+ mounted on the HMD. The participant's position within the HIVE was tracked using an active infrared position tracking system (WorldViz PPT X-12), with an infrared LED marker affixed to the HMD. Position and orientation information were supplied to the redirected walking algorithms and used to update the participant's virtual viewpoint in real-time. The laptop was responsible for rendering the VE, generating RDW steering

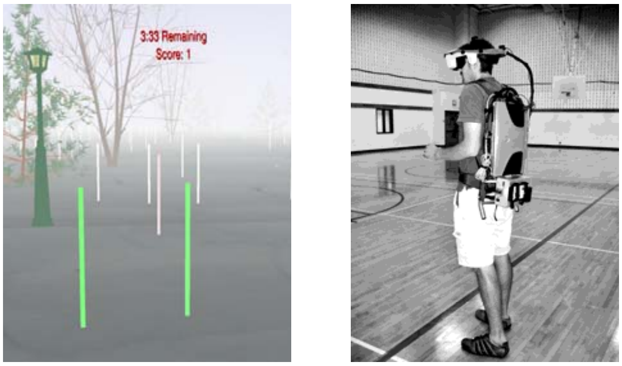

Fig. 9. Left panel – screen capture of the VE used in the live user experiment. Participants gathered posts for points. Right panel – an immersed user wears the HIVE's backpack rendering

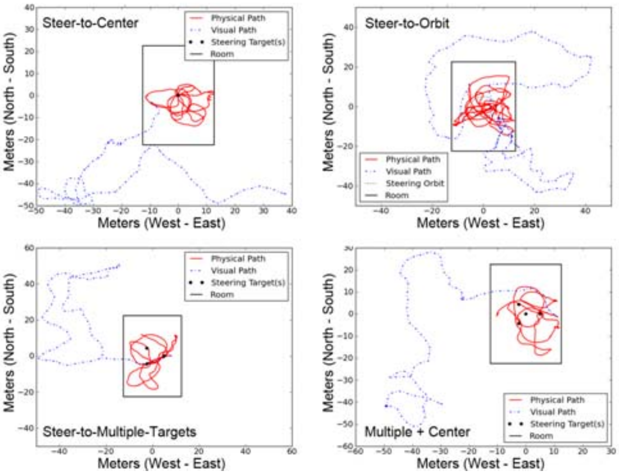

Fig. 10. Representative paths taken by one participant while being redirected by each of the candidate algorithms.

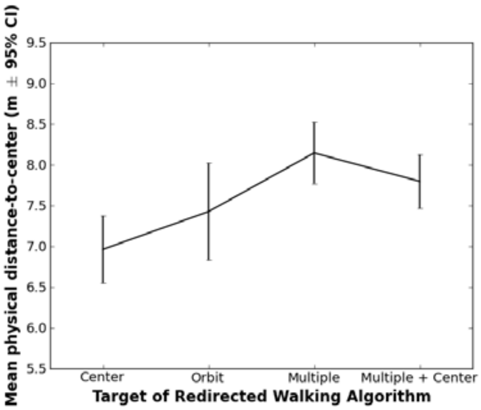

Fig. 11. Mean physical distance-to-center for live users being redirected each of the candidate algorithms.

adjustments, and processing orientation data from the orientation tracker. Position information was aggregated on a separate PC and transmitted to the user-worn laptop via a WiFi (802.11x) signal.

The VE was a lightly-wooded forest with snow and fog that obscured distant objects and masked the far reaches of the environment (see Fig. 9). The forest was simulated to be 100m x 160m (16,000 m^2) and thus was more than 14 times the size of the physical tracking area (1,125m^2). In the simulated forest, 500 colored posts (1.5m tall) were pseudo-randomly positioned. The posts were colored either red, green, or blue. Posts would disappear and generate an auditory cue (a coin sound) if the participant came within 0.5m, and would reappear after 20s in a location that was randomly displaced up to 2m from its previous location. Thus, participants had access to an infinite number of posts in their collection task.

Procedure. All participants were greeted in the lab and offered a brief introduction to the experiment before giving informed consent. The search task was described, and participants were able to view a desktop version of the forest VE but not interact with it. When a participant indicated that they understood the task, they were asked to don the backpack and HMD and to begin.

Four rounds of searching were completed, with each round lasting 4 min and using a different steering algorithm. The order of the algorithms was randomized. The first search began from the center of the tracking space, and each subsequent round was begun from the endpoint of the last, with the exception that participants who finished near a wall were directed to face away from it prior to beginning the next round.

4.2.2 Results and Discussion

Candidate algorithms were evaluated on several dimensions to determine how effective each was at containing users to the HIVE's tracking space. As with the simulated results above, the mean physical distance-to-center and mean absolute rate of redirection were examined. Maximum physical distance was not a viable metric in the live user experiment because of physical walls that constrained the maximum achievable physical distance was in all conditions. Notably, the nearest walls (12.5m) were closer than the average maximum distances observed in simulations (~15m). Instead, the mean number of wall contacts was used to evaluate the upper bounds of navigation in each condition. Participants' mean visual distance-to-center and search-task score were also used as manipulation checks. Because different visual paths were pursued by participants in each condition, it was important to assess whether participants tended to explore an equally large virtual area and to complete their primary task effectively regardless of the steering method. For each measure, 95% confidence intervals (CI) were calculated, which excluded between-participant variability [19]. Differences between algorithms were assessed using a repeated-measures ANOVA with RDW algorithm as the only factor.

Mean physical distance to center. For each recorded position along a participant's traversed path, the linear distance to the center of the tracking space was calculated. Samples of one participant's paths for each condition are shown in Fig. 10. Mean physical distances for each condition are plotted in Fig. 11. As with the simulated results above, there was a significant main effect of algorithm type ($F(2,20) = 4.39$, $p < 0.05$). Steer-to-Center kept users significantly closer to the tracking area center (6.95m ± 0.41) than either of the Steer-to-Multiple Targets algorithms (Multi: 8.08m ± 0.38; Multi + Center: 7.94m ± 0.33). Steer-to-Orbit fell in the middle (7.37m ± 0.59) and exhibited the highest variability. Planned contrasts indicated that there was no significant difference between Steer-to-Center and Steer-to-Orbit ($F<1$), or between the two variations of Steer-to-Multiple Targets ($F<1$). The difference between these two sets was significant, however ($F(1,11) = 28.51$, $p < 0.001$).

Number of wall contacts. It was anticipated that less effective steering would lead to an increased need for the experimenter to manually stop and redirect participants who approached a wall. This turned out to be the case. A chi-square test revealed that the frequency of collisions differed across algorithms ($\chi2$ (3, N = 96) = 11.67, $p < .05$). This effect was driven primarily by the high number of collisions seen with Steer-to-Multiple targets (38 total) relative to the other algorithms (16 total for Steer-to-Center; 20 total for Steer-to-Orbit; 22 total for Steer-to-Multiple+Center). Trends of wall collisions were consistent with the results of mean physical distance above and with the general trends of the simulation. Specifically, Steer-to-Center yielded the fewest wall collisions with Steer-to-Orbit a close second, while the two variations of Steer-to-Multiple-Targets performed the worst. In practice, however, wall contacts occurred relatively infrequently in all conditions. Wall contacts occurred at an average rate of 1.13 times ± 0.23 per 4 min of navigation across all conditions. For comparison, un-redirected participants completing the same search task in [15] contacted the walls at a rate of 4.56 times in 4 min of navigation.

Mean Rate of redirection. The mean absolute rate of redirection exhibited a pattern opposite that seen for mean physical distance as expected. There was a significant main effect of steering algorithm ($F(3,29) = 16.34$, $p < 0.001$) driven mainly by the difference between the two Steer-to-Multiple Target algorithms and the remaining methods (Center and Orbit; $F(1,11) = 34.84$, $p < 0.001$). There was no significant difference between the Steer-to-Center and Steer-to-Orbit ($F < 1$), which showed relatively higher rates of steering. Nor were there significant differences between the two Steer-to-Multiple Target algorithms ($F(1,11) = 1.15$, $p = 0.31$). This was consistent with the simulated results above.

An evaluation of the distribution of redirection rates again revealed that each algorithms showed a main peak around the target steering rate (global mean: 7.87°/s ± 0.10), but the two Steer-to-Multiple algorithms showed a secondary peak near 0°/s. This also was consistent with the simulated results of the simulation and reflected an increased tendency to dampen the rate of redirection – and thus its effectiveness.

Visual distance to center. One goal of RDW is to enable users to explore areas larger than the physically available tracking space. However, it would be undesirable for the RDW algorithm to impact the *visually* chosen path while manipulating a user's physical path. Importantly, there were no significant differences between conditions for the mean visual distance-from-center ($F < 1$) or the average maximum visual distance achieved ($F < 1$). Participants were equally prone to wander in all conditions. Thus, any differences in physical distances or wall contacts were a result of the steering algorithm and not an artifact of the underlying visual paths. Mean visual distance-to-center collapsed across all users and conditions was 31.94m ± 0.58, or 152% further than the HIVE's near walls. The average maximum distance across all conditions was 55.65m ± 0.82, which is more than double the largest physical distance attainable in the HIVE. In other words, participants exploration of the VE would have been impossible without RDW.

Search Task Score. As with visual distance, there were no significant differences in participants' task scores across conditions ($F<1$). Participants were able to complete their assigned task with equivalent success in each condition.

5 General Discussion

Several general trends are apparent across the results of the simulated and live-user redirection comparisons. First, Steer-to-Center tended to outperform all other algorithms at containing users to the smallest possible area. This pattern held true both in simulations that shared a common set of visual trajectories, and with live users who selected unique paths in each condition. Importantly, all four RDW algorithms shared the same gains, base rates, settings, and codebase to implement steering. The only difference among these algorithms was *where* the user was being redirected. Thus, it seems that if one wants to constrain users to the smallest possible area, it is most logical to steer them to the center.

Steer-to-Orbit was often a close second in terms of its performance, and showed some potential in specific areas. Indeed, it outperformed Steer-to-Center for long, straight-line navigation. One potential limitation of the current study is that the only VE considered was designed to be open and not to constrain user's navigation. This was intentional, so that RDW algorithms had to react to unpredictably changing routes, but also limits the generalizability of the present results to other types of VEs. It may be that Steer-to-Orbit might outperform Steer-to-Center in a VE where users must walk in long straight paths and make orthogonal turns (e.g., a series of hallways or streets). Follow-up studies are underway to examine this possibility.

Second, the space requirements for RDW seen here were smaller than has been posited in the literature. To some degree, this is a function of the tighter turn radius shared by the steering algorithms. However, increasing the turn radius nearly three-fold yielded only a relatively modest increase in the required space. In the case of Steer-to-Center, increasing the turn radius from 7.5m to 22m (293.33%) increased the maximum distance from 14.83m to 17.09m (2.71m, or 18.85%). Steinicke and his colleagues [12] had estimated that a 40m x 40m tracking space would be required to redirect users at a 22m turn radius. The present results indicate that less than 35m x 35m would suffice, due to the availability of large steering corrections when users turn or look around, and due to the relatively high prevalence of such behaviors.

Third, the two variations of Steer-to-Multiple-Targets exhibited an increased tendency to dampen steering as users approached a steering target, which lessened the effectiveness of their steering. This limitation could be overcome by redesigning the target-selection logic of the algorithms, but in the end, the simpler Steer-to-Center algorithm may be a better choice. Again, it seems unwise to steer users towards waypoints that are displaced from the center of the tracking space if the goal is to constrain them to the smallest possible area.

Fourth, all RDW algorithms appeared to have difficulty with frequent, and especially alternating, changes in direction and with curved paths that counteracted steering corrections. While much focus has been given to the perceptual thresholds of RDW components, the higher-level behavioral patterns of users need to be considered when evaluating how and where to steer users. It may be optimal to temporarily redirect a user *towards* a wall, for example, if he is already naturally veering in that direction. The natural curvature in the user's path can then be amplified to correct his physical path more efficiently.

Finally, all four algorithms performed reasonably well. Despite the measureable differences among algorithms in terms of distances and wall contacts, all four algorithms were reasonably effective at constraining users to a relatively small area and minimizing the number of times they reached the boundaries of the physical space. Any could be put into practice with a reasonable degree of success. By the same token, none truly "succeeded" in the strict sense of eliminating *all* wall contacts in the live user study. While a larger facility or improved steering logic may further reduce the number of times users reach the outer bounds of the tracking space, there are likely to always be extreme cases in which users cannot be redirected successfully. This speaks to the need for fail-safe resetting methods, even for the best generalized RDW algorithms [20].

Going forward, the most useful tact may be to intelligently switch between different RDW algorithms depending on factors that include the current behaviour of the user and the nature of the VE. For example, if a user is traversing a long, straight portion of the VE and turns are unlikely in the near future, then temporarily switching to Steer-to-Orbit may be advantageous. Likewise, a user who is arcing against Steer-to-Center steering instructions may be temporarily steered towards an alternate waypoint in the spirit of Steer-to-Multiple-Targets. Finally, VEs that contain rooms and doorways could augment basic RDW with the recently described change-blindness techniques [14]. This method imperceptibly moves a doorway from one wall to another to induce a large, 90 ° change in the user's direction, but may inhibit spatial learning as it alters the global spatial arrangement of the VE [14]. While Steer-to-Center performed best under general circumstances, each of these other approaches has value in at least some circumstances and should not be discounted. The sum of their efforts, then, may be more effective than any technique in isolation.

Acknowledgments

Thanks to David Waller for helpful comments on early drafts of this manuscripts. This work was supported in part by grants from the Army Research Office and the National Science Foundation.

REFERENCES

[1] M. Slater and S. Wilbur, "Through the looking glass world of presence: A framework for immersive virtual enviornments.," in *FIVE '95 Framework for Immersive Virtual Environments.*, M. Slater, Ed., QMW University of London, 1995.

[2] R. L. Klatzky, J. M. Loomis, A. C. Beall, S. S. Chance and R. G. Golledge, "Updating an egocentric representation during real, imagined, and virtual locomotion.," *Psychological Science,* vol. 9, pp. 293 - 298, 1998.

[3] S. Razzaque, D. Swapp, M. Slater, M. C. Whitton and A. Steed, "Redirected walking in place.," in *Proceedings of the Eurographics Workshop on Virutal Environments.*, 2002.

[4] R. P. Darken, W. R. Cockayne and D. Carmein, "The omni-directional treadmill: A locomotion device for virtual worlds.," in *Proceedings of ACM Symposium on User Interface Software and Technology*, 1997.

[5] J. M. Hollerbach, "Locomotion interfaces," in *Handbook of Virtual Environments: Design, Implementation, and Applications*, NJ, Erlbaum, 2002, pp. 239 - 254.

[6] S. Razzaque, "Redirected walking," doctoral dissertation, University of North Carolina, Chapel Hill, https://www.cs.unc.edu/cms/publications/dissertations/razzaque.pdf, 2005.

[7] S. Razzaque, Z. Kohn and M. C. Whitton, "Redirected walking," in *Proceedings of Eurographics 2001*, 2001.

[8] J. L. Souman, I. Frissen, M. N. Sreenivasa and M. O. Earnst, "Walking straight into circles," *Current Biology,* vol. 19, pp. 1538 - 1542, 2009.

[9] V. Interrante, B. Ries, amd L. Anderson; "Seven League Boots: A New Metaphor for Augmented Locomotion through Moderately Large Scale Immersive Virtual Environments," *IEEE Symposium on 3D User Interfaces, 2007. 3DUI '07.*, 10-11 March 2007.

[10] F. Steinicke, G. Bruder, J. Jerald, H. Frenz and M. Lappe, "Analyses of human sensitivity to redirected walking," in *15th ACM Symposium on Virtual Reality Software and Technology*, New York, 2008.

[11] F. Steinicke, G. Bruder, J. Jerald, H. Frenz and M. Lappe, "Real walking through virtual enviornments by redirection techniques," *Journal of Virtual Reality and Broadcasting,* vol. 6, p. no. 2, 2009.

[12] 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,* vol. 16, pp. 17 - 27, 2010.

[13] T. Field and P. Vamplew, "Generalised algorithms for redirected walking in virtual environments," in *AISAT2004: International Conference on Artificial Intelligence in Science and Technology*, Hobart, Tasmania, 2004.

[14] E. A. Suma, D. Krum, S. Finkelstein, and M. Bolas. "Effects of Redirection on Spatial Orientation in Real and Virtual Environments," *IEEE Symposium on 3D User Interfaces, 2011*, pp. 35-38, 2011.

[15] E. Hodgson, E. Bachmann and D. Waller, "Steering immersed users of virtual environments: Assessing the potential for spatial interference," *ACM: Transactions on Applied Perception,* vol. 8, pp. 1 - 22, 2011.

[16] C. T. Neth, J. L. Souman, D. Engel, U. Kloos, H. H. Bulthoff, and B. J. Mohler, "Velocity-dependent dynamic curvature gain for redirected walking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18(7), pp. 1041-1052, 2012.

[17] D. Waller, E. Bachmann, E. Hodgson and A. C. Beall, "The HIVE: A huge immersive virtual environment for research in spatial cognition.," *Behavior Research Methods,* vol. 39, pp. 835 - 843, 2007.

[18] D. Engel, C. Curio, L. Tcheang, B. Mohler and H. H. Bulthoff, "A psychophysically calibrated controller for navigating through large environments in a limited free-walking space," in *VRST*, Bordeaux, France, 2008.

[19] D. Cousineau, "Confidence intervals in within-subject designs: A simpler solution to Loftus and Masson's method," *Tutorials in Quantitative Methods for Psychology,* vol. 1, pp. 42 - 45, 2007.

[20] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. Rieser, and B. Bodenheimer: "Exploring large virtual enviornments using scaled translational gain," in *APGV 2007*, Tübingen, Germany, 2007.