

# Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance

Mahdi Azmandian<sup>1</sup>, Timofey Grechkin<sup>1</sup>, Mark Bolas<sup>1,2</sup>, and Evan Suma<sup>1</sup>

<sup>1</sup>USC Institute for Creative Technologies, Los Angeles, CA

<sup>2</sup>USC School of Cinematic Arts, Los Angeles, CA

---

## Abstract

*Redirected walking provides a compelling solution to explore large virtual environments in a natural way. However, research literature provides few guidelines regarding trade-offs involved in selecting size and layout for physical tracked space. We designed a rigorously controlled benchmarking framework and conducted two simulated user experiments to systematically investigate how the total area and dimensions of the tracked space affect performance of steer-to-center and steer-to-orbit algorithms. The results indicate that minimum viable size of physical tracked space for these redirected walking algorithms is approximately  $6m \times 6m$  with performance continuously improving in larger tracked spaces. At the same time, no “optimal” tracked space size can guarantee the absence of contacts with the boundary. We also found that square tracked spaces enabled best overall performance with steer-to-center algorithm also performing well in moderately elongated rectangular spaces. Finally, we demonstrate that introducing translation gains can provide a useful boost in performance, particularly when physical space is constrained. We conclude with the discussion of potential applications of our benchmarking toolkit to other problems related to performance of redirected walking platforms.*

Categories and Subject Descriptors (according to ACM CCS): Information Interfaces and Presentation [H.5.1]: Multimedia Information Systems—Artificial, augmented, and virtual realities; Computer Graphics [I.3.6]: Methodology and Techniques—Interaction techniques; Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Virtual reality

---

## 1. Introduction

Creating a compelling experience of exploring arbitrarily large virtual environments using limited physical space is a critical challenge for virtual reality systems. The most promising solution to this problem is Redirected Walking (RDW) [RKW01], which decouples user’s virtual path from the real-world trajectory (Figure 1). The key principal behind this approach is to leverage unnoticeable perceptual manipulations such as imperceptibly small visual rotations (*rotation* and *curvature gains*) and translations (*translation gains*) to redirect a physically walking user away from the boundaries of the tracked space. RDW provides the benefits of unconstrained physical walking in virtual environments such as enhanced sense of presence [UAW\*99], efficient navigation [RL09] [SCFW10], and better cognitive maps of the environment [RVB11] at a limited cost of some cognitive load on the user [BLS15] and without interfering with navigation and spatial cognition [HBW11] [SKFB11].

In order to apply redirected walking in practice, one needs to understand how the performance of RDW algorithms may be influenced by external parameters such as size and shape of the tracked physical space. It is also important to compare the performance of various algorithms relative to each other. These tasks are non-trivial, because the performance of the RDW algorithm depends on a variety of interacting factors including user behavior, tracked space dimensions, structure of the virtual environment and type of the virtual path, as well as internal parameters of the algorithm such as perceptual thresholds and wall-contact resolution (*reset*) mechanisms.

A common approach to defining physical space requirements for RDW setup is to focus on estimating the area of the smallest physical tracked space that enables users to walk along an infinite straight virtual path without ever reaching the boundaries. However, this approach relies on a special case scenario (long straight path) that may not accu-

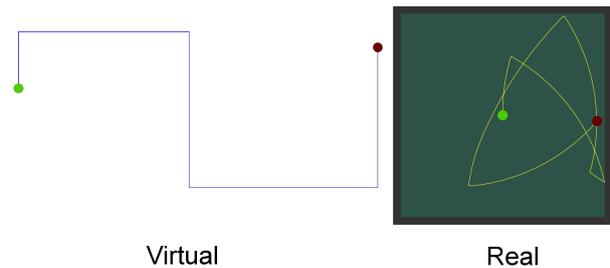


Figure 1: Principles behind redirected walking: example of disassociation between user's path in virtual world (left) and tracked space (right) moving from the green point to red. In this example steer-to-center algorithm is combined with resets that occur at the boundaries of tracked space.

rately represent general RDW behavior. For the same type of algorithms it produced a wide range of estimates for optimal size of tracked space from  $30m \times 30m$  [RKW01], to  $35m \times 35m$  [HB13], to  $40m \times 40m$  [SBJ\*10], to as much as  $60m \times 60m$  [FBV04]. Finally, it also does not help to understand effectiveness of redirected walking in tracked spaces that are smaller than the optimal size or have non-square shape.

The goal of this paper is to systematically evaluate the physical space requirements for redirected walking while controlling for the most salient factors that may affect RDW performance; this is to shed light on the optimal tradeoff between cost and performance for tracked space requirements.

## 2. Related Work

Since the concept of redirected walking was first introduced by Razzaque et al. [RKW01], there have been a lot of interest in its modification and improvement. Today RDW algorithms can be broadly classified as either *reactive* or *predictive*. Reactive algorithms assume no knowledge of the virtual environment or future user direction of travel and rely only on the current travel direction, speed, and position relative to the physical tracked space to steer the user. Examples include widely used Steer-To-Center (S2C) and Steer-To-Orbit (S2O) algorithms initially proposed by Razzaque et al. and later improved by Hodgson and Bachmann [HB13]. Predictive algorithms [ZWBH13, NHK14, AYBS14] attempt to take advantage of the information about the past behavior of the user and the structure of the virtual environment to forecast user movement into the future and leverage this prediction to improve effectiveness of redirection. These algorithms typically optimize the choice of steering action by searching among possible outcomes of applying various options in a near-term time horizon.

The algorithms above rely on injecting small visual rotations when the user is moving forward (curvature gains) or rotating (rotation gains) to achieve redirection. Another

mechanism for decoupling visual and physical movement is scaling the lateral locomotion (translation gains). Translation gains have been successfully used to explore large scaled virtual environments [IRA07, XLW\*10], though never formally coupled with other gains as a general reactive algorithm.

In general it is very difficult to guarantee that walking user will not reach the boundary of the tracked space. One approach is to allow redirection gains to increase as the user approaches the boundary [NHS04, Su07]. This is likely to require increasing gain levels so much that redirection will become obvious. Instead most RDW approaches rely on stopping and reorienting the user as necessary using a *reset* procedure. Reset procedures can be administered manually by calling out instructions for the user to stop moving and turn away from the boundary [HB13] or in a variety of automated methods [WNR\*07, PFW11, NHK14]. Reset strategies include fixed 360 degrees visual rotation corresponding to a 180 degrees physical one [WNR\*07], aligning the user perpendicular to the virtual walls or to face the center of the tracked space [PFW11] and orientation towards the farthest corner of the tracked space [ZBHW13]. Predictive algorithms can also calculate specific rotation angles to fit their optimized strategy [NHK14] [AYBS14]. Regardless of implementation details, resets drastically alter real-world user trajectories and therefore have significant impact on overall performance. This calls for reset methods to be considered an integral part of the RDW algorithms. Furthermore, the work of Peck et al. [PFW11] demonstrates that reset-like periodic reorientation may be sufficient to explore large virtual spaces without any other RDW techniques.

In addition to internal factors discussed so far performance of RDW algorithms depends on a number of external factors. One of the most obvious external parameters affecting the effectiveness of redirected walking algorithms is the layout of physical tracked space. This not only directly influences end users, but also is arguably the primary practical issue in a RDW setup. Performance of RDW algorithms has been studied in a variety of settings. Simulated studies considered square-shaped tracked spaces of  $5m \times 5m$  [Su07],  $20m \times 20m$  [ZWBH13],  $35m \times 35m$  [HB13] as well as a circular tracked space of diameter  $60m$  [FBV04]. In addition, user studies were conducted in square-shaped spaces of  $4m \times 4m$  [NHS04],  $4.3m \times 4.3m$  [SCFW10],  $5m \times 5m$  [WNR\*07],  $6.5m \times 6.5m$  [PFW11],  $9m \times 9m$  [PFW11],  $11m \times 11m$  [SKFB11] and rectangular tracked spaces ranging between  $9.1m \times 7.6m$  [IRA07],  $10m \times 7m$  [SBJ\*10],  $12.6m \times 6.2m$  [NHK14], and  $45m \times 25m$  [HB13]. Despite the great variety of configurations, there are very few studies comparing RDW performance across multiple tracked spaces of different size. The differences in implementation details also make it hard to generalize the results beyond a specific tracked space used in a given study. In combination with methodological limitations of estimating the "optimal" size of tracked space dis-

cussed in section 1, existing research literature does not provide clear guidance regarding physical space requirements for a RDW setup.

The type of virtual environment where redirected walking is applied determines the layout of users' virtual trajectory and can also significantly affect the algorithm performance. For instance, Hodges and Bachmann [HB13] initially reported S2C to outperform S2O, while a different choice of virtual environment resulted in S2O to be shown superior [HBT14]. For reactive algorithms the essence of the virtual environment is completely captured by the layout of the virtual path taken by the user. It is, therefore, common to guide the user through the environment using a series of waypoints to determine the layout of the virtual path [RKW01, FBV04, Su07, HB13]. Such a path can be defined manually or generated from elements such as straight lines, zig-zags, and figure-eights [HB13].

### 3. Our approach

In this paper we attempt to systematically evaluate physical tracked space requirements for redirected walking by comparing the performance of several RDW algorithms across a variety of tracked space layouts. We focus on the most commonly used reactive algorithms Steer-To-Center (S2C) and Steer-To-Orbit (S2O) since they are generally applicable to any given virtual environment without making assumptions about the structure of the environment and the user behavior. We also investigate the effects of combining rotation, curvature, and translation gains. To that end we introduced a simple algorithm based on translation gains alone and its combinations with S2O and S2C. All of these algorithms used a common reset method, which stopped users at the boundary of the tracked space and reoriented them toward the center.

We used the simulated user approach, which allowed us to exclude between-user variability and enabled experiments with large numbers of lengthy trials. The experiments were conducted on a variety of randomized virtual paths that were generated to represent common types of virtual environments. To facilitate future user studies we implemented all redirected walking algorithms used in this paper in Unity 3D as an interactive framework and then adapted it to run simulated experiments with computer-controlled simulated users and without rendering the 3D scene.

Our first experiment explores the effect of the overall area of square tracked space on RDW performance. The second experiment looks at the differences between rectangular tracked spaces that have the same area but differ in dimensions.

### 4. Experiment 1: What is the optimal tracked space size?

One might intuitively expect redirected walking algorithms to perform better as tracked space becomes larger. It is im-

portant, however, to put this performance improvement into context. In larger tracked spaces users can travel longer distances even when no redirected walking techniques are applied. This subtle point has been often ignored in earlier experiments investigating the effectiveness of redirected walking algorithms. In this experiment a simulated redirected walking environment was used to quantify the effects of tracked space size on the performance of various redirected walking algorithms.

#### 4.1. Experimental setup

##### 4.1.1. Redirected walking algorithms

The five RDW algorithms used in this experiment are defined as follows:

**Steer-To-Center (S2C)** The basic heuristic used by S2C is to inject small visual rotations to steer the user towards the center of the tracked space. Our implementation was based on a modification to Razzaque's original S2C implementation, introduced in [HB13]. The curvature radius was set to 7.5 meters, and rotations were scaled by factors between 0.85 and 1.3.

**Steer-To-Orbit (S2O)** This algorithm functions similar to S2C, but the heuristic is to steer the user to an orbit around the center of the tracked space. S2O was also implemented as described in [HB13]. Gain thresholds were the same as used for S2C.

**Center-based Translation Gain (CTG)** This algorithm used translation gains to slow down the user when she was moving away from the center of the tracked space (i. e. scale visual translation up relative to physical translation). If the dot product between the direction to the center and user's movement vector was negative, the algorithm applied constant translation gain factor. Based on unnoticeable threshold estimate by Steinicke et al. [SBJ\*10] translation gain was set to 14 percent.

**Combined Algorithms (S2C+CTG and S2O+CTG)** Since S2C and S2O both rely exclusively on rotation and curvature gain, and CTG only controls translation gains, it is straight forward to combine both S2C or S2O with CTG. A subtle implementation technicality was to isolate the effects of user's actual movement (caused by walking in the real world) from the user's overall position and orientation change influenced by injected visual translations and rotations.

**Control condition: No Redirection** The performance of these algorithms was compared to a scenario where simulated user travelled within the tracked space without any redirection and was simply reoriented to face the center of the tracked space every time she reached the boundary. During the simulation this condition was essentially treated as a sixth algorithm before the data collected in control condition was folded into the performance measures as described below in section 4.1.5.

#### 4.1.2. Virtual Paths

We defined four categories of virtual paths representing typical virtual environment scenarios used in practice (see Table 1). Note that in all cases the expected length of each path is 1000 meters.

Category	$D_d$ (meters)	$D_a$ (radians)	$N_w$
Office Building	unif(2, 8)	$\{-\frac{\pi}{2}, \frac{\pi}{2}\}$	200
Exploration (small)	unif(2, 6)	unif( $-\pi, \pi$ )	250
Exploration (large)	unif(8, 12)	unif( $-\pi, \pi$ )	100
Long Walk	{1000}	N/A	1

Table 1: Four categories of virtual paths.

Virtual paths were randomly generated based on *distance sampling distribution*  $D_d$ , *angle sampling distribution*  $D_a$ , and a *waypoint count*  $N_w$ . Given these parameters a virtual path can be generated by taking the user's initial configuration, setting a waypoint at distance  $d_1$  (sampled from the distance distribution) along the user's current forward, then placing the next waypoint at distance  $d_2$  (sampled from the distance distribution) such that the user will be required to rotate angle  $a_1$  (sampled from the angular distribution) to face the next waypoint and so on. Therefore in practice, the user will be required to perform a succession of walk and turn actions to clear all waypoints.

#### 4.1.3. Simulated walker

A walking user was simulated by an autonomous agent programmed to traverse the virtual path by walking toward the next waypoint with constant linear velocity of 1 m/s while maintaining its heading toward the waypoint (i.e. attempting to walk on a straight line in the virtual environment). Upon reaching a waypoint, the simulated walking stopped (instantly) and turned in place with angular velocity of 90 deg/s to face the next waypoint (as implemented in [HB13]). Note that for this study no noise was introduced to the simulated user's translation and rotation. This guarantees the simulated user will walk along the virtual path defined by the series of waypoints.

#### 4.1.4. Resets

When the simulated user reached a boundary of the tracked space, a reset procedure was initiated. The simulation paused and the simulated user was reoriented to face the center of the tracked space while maintaining the same orientation in the virtual environment. This result is equivalent to the result of the typical reset procedures performed with real users. Note that a reset was only triggered when the user is facing towards the boundary in contact (a 180 degree range), not merely brushing the boundary while still heading to a direction inside the tracked space.

#### 4.1.5. Performance measures

The most direct way to assess how a successful redirection algorithm is in keeping the user within the boundaries of the allocated physical space is the **reset count**, i.e. the number of times the user contacted the boundary and a reset procedure was triggered. We define our primary measure - **relative effectiveness** - as reduction in reset count relative to no-redirection under the same conditions:

$$RE_{algorithm} = \frac{R_{NoRedirection} - R_{algorithm}}{R_{NoRedirection}}. \quad (1)$$

#### 4.1.6. Procedure

At the start of each trial simulated user was placed in the middle of the tracked space facing "north"(parallel to one of the tracked space sides). For each tracked space (squares ranging from 1 to 60 meters in one dimension) and algorithm (six, including No Redirection) pair we generated 10 randomized virtual paths in each of the first 3 virtual path categories described above and obtained corresponding reset counts. For the Long Walk path category a single reset count was obtained for each pair of algorithm (again, including No Redirection) and area size. We then averaged no redirection reset counts for each path type and tracked space and used these values to compute relative effectiveness metric for each of the 5 algorithms.

#### 4.2. Results

Initial examination of algorithm performance across virtual path types revealed that performance for Office Building, Small Exploration, and Large Exploration virtual path categories was quite similar. Overall, relative effectiveness of all algorithms increased with tracked space size. In contrast, Long Walk path resulted in qualitatively different step-like performance function, where the effectiveness remained relatively flat until tracked space size reached a critical point. For all tracked spaces above this critical value physical paths were completely enclosed into the tracked space, which correspond to 100 percent redirection effectiveness. For further analysis we decided to consider Long Walk Path separately and combine the results for the first 3 randomized path types.

##### 4.2.1. Long Walk Paths

Figure 2 shows experimental results for effectiveness of redirected walking relative to no redirection for Long Walk paths. The four algorithms using rotation and curvature gains to redirect the user (S2C, S2C+CTG, S2O, S2O+CTG) are able to achieve 100 percent effectiveness for tracked spaces sized 31 meters or larger. This happens when redirection algorithms are able to fully redirect the user onto a circle trajectory within the tracked space. In addition, S2O and S2O+CTG algorithms achieve a near-perfect effectiveness of about 98 percent for tracked spaces sized between 22 and

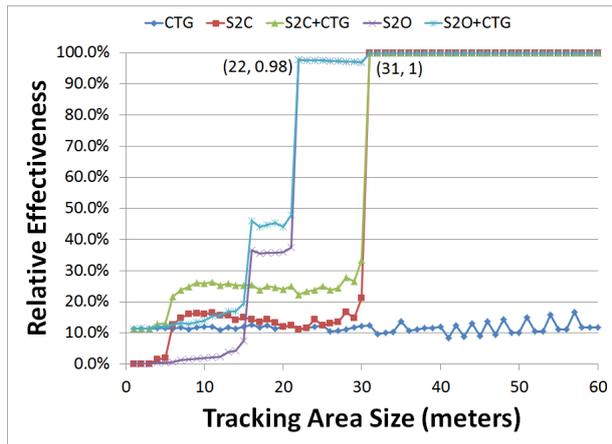


Figure 2: Relative effectiveness of redirected walking for Long Walk path type.

30 meters. In practice this means that a reset is required before the user can be redirected into fully circular trajectory.

The results indicate that both S2C and S2O algorithms are not very effective in very small tracked spaces. S2O algorithm requires tracked space of at least 15 meters to achieve at least 10 percent effectiveness relative to no redirection (and to outperform simple CTG algorithm). S2C algorithm requires at least 6 meters to achieve 10 percent effectiveness and to outperform CTG. For intermediate size tracked spaces between 16 and 31 meters S2O outperforms S2C in Long Walk path scenario.

CTG algorithm has virtually constant effectiveness of about 12 percent over the full range of tracked space sizes. When combined with S2C and S2O algorithms CTG improves effectiveness relatively to original versions by approximately the same amount up to tracked space size of 31 meters, where all four algorithms reach 100 percent effectiveness. For S2O and S2O+CTG algorithms the convergence in performance happens at 22 meters.

#### 4.2.2. Randomized path types

To simplify the comparison between algorithms we grouped tracked space sizes into three categories: up-to 20 meters, 20 to 40 meters, 40 to 60 meters (see Figure 3 (bottom)). We then performed a 2 way ANOVA with grouped area size and algorithm as explanatory factors and relative effectiveness as response variable. The analysis revealed a significant interaction between effects of area size and algorithm type on relative effectiveness ( $F(8, 8985) = 209.326, p < 0.001$ ).

Planned post-hoc comparisons revealed that combination of translation gains (CTG) with other algorithms resulted in significantly higher effectiveness for all three groups of tracked space sizes (all p values below 0.001). In addition, for tracked spaces up-to 40 meters S2C algorithms was more

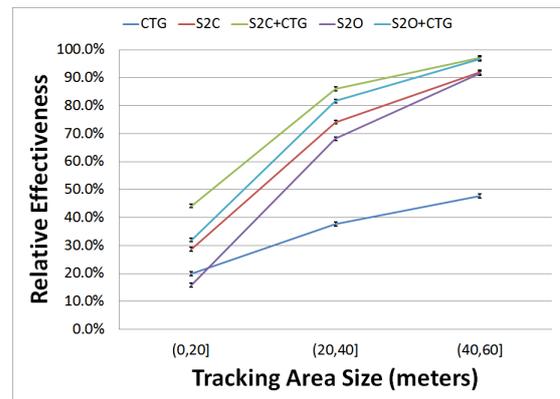
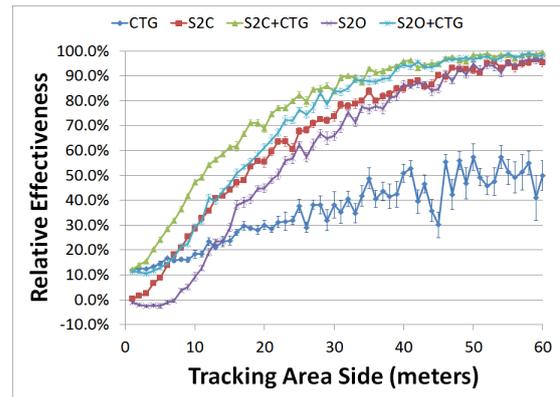


Figure 3: Relative effectiveness of redirected walking algorithms for each tested tracked space size (top) and three aggregate groups of tracked space sizes (bottom). Error bars represent standard error.

effective than S2O (p values less than 0.001). The same was true for the modified versions S2C+CTG and S2O+CTG. However, for tracked space sizes between 40 and 60 meters we found no significant differences between S2C and S2O ( $p = 0.587$ ) and between S2C+CTG and S2O+CTG ( $p = 0.618$ ).

#### 4.3. Discussion

One very intuitive conclusion from these results is that redirected walking algorithms we considered here have a minimal viable tracked space requirement. For a Long Walk scenario S2C required tracked space of at least  $6m \times 6m$  to achieve 10 percent relative effectiveness, which we consider minimally viable. Randomized path scenarios produced similar results: for  $5m \times 5m$  tracked space relative effectiveness of S2C was 8.6% (95% CI [6.5%, 10.6%]); for  $6m \times 6m$  relative effectiveness was 13.7% (95% CI [11.3%, 16%]). For S2O algorithm minimally viable tracked space needs to be larger. Naturally, application of translation gains helps

to somewhat relax this requirement by effectively making tracked space “bigger”.

The Long Walk path data suggest that a 31m x 31m tracked space would be sufficient to achieve infinite straight-line walking in virtual reality for any of the algorithms we tested. This result is in line with earlier estimates of 30m by Razzaque et al. [RKW01] and 35m by Hodgson and Bachmann [HB13].

However, the near perfect performance achieved by S2O in a tracked space as small as 22m×22m indicates that these results are highly sensible to the initial position and orientation of the user. By strategically placing and orienting the user to match the desired trajectory from the beginning we could have achieved “optimal” performance in this smaller tracked space. Conversely, an unfortunate initial configuration (for example, near the tracked space boundary) is bound to require at least one reset regardless of available space. Based on these conclusions, we would argue that the traditional approach of finding an “optimally sized” tracked space, which enables an infinitely long straight-line virtual trajectory is not necessarily a good way to define real-world space requirements for redirected walking.

The randomized paths results suggest that redirected walking effectiveness gradually improves as tracked space area becomes larger (Figure 3, top). Since the rate of performance improvement slows down for larger tracking areas, it seems that choice of tracking area size should be based cost-benefit analysis of potential performance gains vs. expenses for enlarging the tracked space further.

When comparing S2C and S2O to each other we conclude that the choice of algorithm does not seem to matter for very large tracked spaces (40m × 40m and larger). For intermediate and small tracked spaces S2C seems to be more preferable. It is particularly true for tracked spaces smaller than approximately 15m × 15m, where relative effectiveness of S2O algorithm is very small.

In practice most tracked spaces are unlikely to exceed 10m × 10m. It is clear that under such circumstances effectiveness of S2C and S2O algorithms remains limited. As a consequence the users are bound to experience relatively large number of resets. Therefore, it is critical to design reset and reorientation mechanisms to seamlessly integrate into overall virtual reality experience. Our data also demonstrates that for such relatively small tracked spaces combining S2C and S2O algorithms with translation gains provides a significant boost in effectiveness. However, further research is required to fully understand the effects of simultaneous application of curvature and translation gains on the moving user. The primary concern here is that in combination these two types of gains may become more noticeable.

## 5. Experiment 2: What shape of tracked space is optimal?

Physical tracked spaces are not always square. Room layout and hard-to-move physical obstacles may create situations where the use of rectangular shape may be preferable to maximize the use of available physical space. It is, however, unclear how shape of the tracked space may impact performance of redirected walking algorithms. For example, when comparing 8m×9m vs. 6m×12m tracked spaces, it is not apparent which of the two would perform best. Experiment 2 was designed to investigate the impact of tracked space shape on effectiveness of redirected walking algorithms.

### 5.1. Experimental setup

We compared performance of the five redirected walking algorithms (CTG, S2C, S2O, S2C+CTG, S2O+CTG) introduced in experiment 1 (section 4.1.1) in square and rectangular tracked spaces. The shape of the tracked space was determined by the ratio of its longer side length to the shorter side length. The shape ratios tested were 1.0, 1.5, and 2.0. To ensure fair comparison total area was constant at 400 square meters. The algorithms were tested using three types of randomized virtual paths — “Office”, “Large Exploration”, and “Small Exploration” (see section 4.1.2).

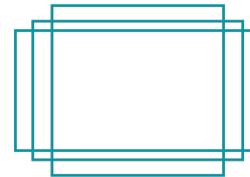


Figure 4: Illustration of tracked space shape changing with fixed area size (ratios 1, 1.5 and 2).

Overall experimental setup was similar to that used in experiment 1. At the start of each trial simulated user was placed in the center of the tracked space. To control for possible effects of initial user orientation, the experiment contained equal number of trials where user started going “North” (along the short side of the rectangle), “East”(along the long side of the rectangle) and “North-East” (towards the corner). For each path type and initial heading direction combination there were 10 trials (90 for each algorithm at each of the tracked spaces for a total of 1350 trials).

Similar to experiment 1 our primary measure was **relative effectiveness** (section 4.1.5) of redirected walking algorithms compared to no-redirection condition. For each tracked space, path type, and initial heading combination we estimated the mean number of resets without redirection (based on 10 trials) and used it to compute relative effectiveness for each of the five redirected walking algorithms.

## 5.2. Results

First, we checked if initial heading of the user affected performance of redirected walking algorithms. We fitted a main effects linear model with relative effectiveness as dependent variable and algorithm, shape ratio, and initial heading as explanatory factors. The test for type III main effects suggests that after adjusting for effects of algorithm type and sides ratio initial heading does not play significant role in explaining variation of relative effectiveness ( $F(2, 1341) = 0.848, p = 0.428$ ). As a result we did not include initial heading in further analysis.

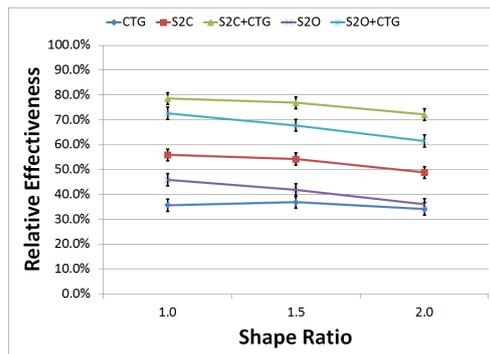


Figure 5: Effects of tracked space shape (sides ratio) on relative effectiveness of redirected walking algorithms. Error bars represent standard errors.

Next we analyzed the effects of algorithm and sides ratio on relative effectiveness using a 2-way ANOVA model. We found a significant two-way interaction between algorithm type and sides ratio ( $F(8, 1335) = 3.284, p = 0.001$ ). This suggests that the effects of the tracked space shape on performance is different for different algorithms as illustrated by Figure 5.

As might be expected, when comparing effectiveness between algorithms we found the same pattern as in experiment 1 for  $20m \times 20m$  tracked space. Therefore, we focused on comparing performance for different area shapes within each algorithm. The pairwise comparisons were performed using Bonferroni adjustments.

The post-hoc analysis revealed that for S2O algorithm relative effectiveness significantly decreased for sides ratio of 1.5 compared to 1.0 ( $p = 0.019$ ) and further decreased for sides ratio of 2.0 vs. 1.5 ( $p < 0.001$ ). Similarly, relative effectiveness of S2O+CTG algorithm decreased for 1.5 vs 1.0 sides ratio ( $p = 0.003$ ) and for 2.0 vs. 1.5 sided ratio ( $p < 0.001$ ). In contrast, we found no evidence that S2C and S2C+CTG algorithms exhibit significant decrease in effectiveness when switching from square to moderately elongated rectangular tracked space with sides ratio of 1.5 ( $p = 0.788$  and  $p = 0.757$  respectively). However, relative effectiveness of these two algorithms decreased for shape

ratio of 2.0 as compared to 1.5 ( $p = 0.001$  and  $p = 0.004$  respectively). We did not find a significant change in relative effectiveness of CTG algorithm ( $p = 1.0$  for shape ratio 1.0 vs. 1.5;  $p = 0.199$  for shape ratio 1.5 vs 2.0; and  $p = 0.887$  for shape ratio 1.0 vs. 2.0).

## 5.3. Discussion

The results of this experiment demonstrate that sensitivity to tracked space shape differs between algorithms. Both S2O and S2O+CTG algorithms perform best in square tracked spaces and have a performance penalty in rectangular spaces. In contrast, CTG algorithm does not appear to be sensitive to tracked area shape. S2C was somewhat more robust compared to S2O. Both S2C and S2C+CTG algorithms did not experienced significant performance penalty in moderately elongated tracked spaces (shape ratio 1.5), but did suffer in more elongated rectangles (shape ratio 2.0). Taken together these observations suggest that square-shaped tracked space was the best choice across all algorithms. These results also suggest that S2C is a more robust algorithm compared to S2O. Given the robustness of S2C-based algorithms in moderately elongated tracked spaces we believe the first priority in planning the layout of the tracked space should be maximization of the total area. However, all other things being equal, more elongated tracking spaces will likely result in higher effectiveness penalty relative to more square spaces.

## 6. Conclusions

The results of our study outline a new way of evaluating the utility of the physical tracked space for redirected walking setups. There is, unfortunately, no single “optimal” size of tracked space that can guarantee complete absence of user contact with the boundary, at least for reactive RDW algorithms such as S2C and S2O. Furthermore, in practice the dimensions of most tracked spaces are much smaller than previously estimated “optimal” size of  $30m \times 30m$  to  $35m \times 35m$  and are unlikely to exceed  $10m \times 10m$ , where reset events are common. It is, therefore, prudent to concentrate on defining and achieving an acceptable level of resets and developing ways for integrating the reset mechanisms as an integral part of the experience.

In relatively small tracked spaces the effectiveness of using traditional reactive RDW algorithms that rely only on rotation and curvature gains is very modest. For tracking spaces smaller than  $6m \times 6m$  effectiveness of original reactive algorithms is unlikely to exceed 10%. Our data demonstrates, however, that a significant performance boost can be achieved by combining these techniques with translation gains. For example, in a  $10m \times 10m$  tracked space S2C reduces the number of resets vs. no redirection condition by 27% (95% CI [22%,33%]); when combined with our naive translation gains implementation into S2C+CTG algorithm it achieves the reduction by 46% (95% CI [41%, 51%]).

Shape of the tracked space also affects RDW performance with square tracked spaces generally being most suitable for redirection. Our data suggests that moderately elongated rectangular shapes can also be used without significant performance penalty.

Our findings are based on a benchmarking framework for assessing redirected walking performance in a systematic controlled manner. This approach relies on analysis of relevant factors to systematically vary the parameters of interest while controlling for all relevant nuisance parameters for complete and fair performance comparisons. We argue that exploration of other properties RDW algorithms in general requires similar controlled approach to yield results that are generalizable and that our framework can be suitable for such studies. In the future we plan to apply this approach to also study performance of predictive algorithms, which may be critical to improve redirected walking performance in tracked spaces of limited size.

Another important avenue for improvement is the development of more realistic representation of a walking user by introducing elements such as noisy movement, gait oscillations, random gaze aversion, and assessing the accuracy of this model with real user experiments. Such changes can substantially improve the validity of simulated user experiments. While live user studies are critically important to gauge realistic performance, simulated user studies enable large scale experiments with multiple trials and long virtual paths that are impractical for human studies.

Finally, in this study our notion of performance centered strictly on reducing wall contacts. A more complete understanding of redirected walking algorithms would require to assess the impact on the user in terms of perceptual and cognitive load and levels of simulator sickness. In the future we plan to include these features in order to be able to predict user's comfort level across various algorithms. Ultimately the goal is to find an optimal compromise between effective wall contact prevention and user strain reduction.

## References

- [AYBS14] AZMANDIAN M., YAHATA R., BOLAS M., SUMA E.: An enhanced steering algorithm for redirected walking in virtual environments. *Proceedings - IEEE VR* (2014), 65–66. 2
- [BLS15] BRUDER G., LUBOS P., STEINICKE F.: Cognitive Resource Demands of Redirected Walking. *IEEE Transactions on Visualization and Computer Graphics* 2626, c (2015), 1–1. 1
- [FBV04] FIELD T., BAY S., VAMPLEW P.: Generalised Algorithms for Redirected Walking in Virtual Environments. *AISAT2004* 65, 11 (2004), 1357–1366. 2, 3
- [HB13] HODGSON E., BACHMANN E.: Comparing four approaches to generalized redirected walking: simulation and live user data. *IEEE TVCG* 19, 4 (2013), 634–43. 2, 3, 4, 6
- [HBT14] HODGSON E., BACHMANN E., THRASH T.: Performance of redirected walking algorithms in a constrained virtual world. *IEEE TVCG* 20, 4 (2014), 579–587. 3
- [HBW11] HODGSON E., BACHMANN E., WALLER D.: Redirected walking to explore virtual environments. *ACM TAP* 8, 4 (2011), 1–22. 1
- [IRA07] INTERRANTE V., RIES B., ANDERSON L.: Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. *IEEE 3DUI 2007* (2007), 167–170. 2
- [NHK14] NESCHER T., HUANG Y.-Y., KUNZ A.: Planning Redirection Techniques for Optimal Free Walking Experience Using Model Predictive Control. *3DUI 2014* (2014), 111–118. 2
- [NHS04] NITZSCHE N., HANEBECK U. D., SCHMIDT G.: Motion Compression for Telepresent Walking in Large Target Environments. *Presence* 13, 1 (2004), 44–60. 2
- [PFW11] PECK T. C., FUCHS H., WHITTON M. C.: An evaluation of navigational ability comparing Redirected Free Exploration with Distractors to Walking-in-Place and joystick locomotion interfaces. *IEEE VR* (2011), 55–62. 2
- [RKW01] RAZZAQUE S., KOHN Z., WHITTON M. C.: Redirected Walking. *EUROGRAPHICS* (2001), 289–294. 1, 2, 3, 6
- [RL09] RUDDLE R., LESSELS S.: The benefits of using a walking interface to navigate virtual environments. 1–18. 1
- [RVB11] RUDDLE R., VOLKOVA E., BULTHOFF H.: Walking improves your cognitive map in environments that are large-scale and large in extent. 1–20. 1
- [SBJ\*10] STEINICKE F., BRUDER G., JERALD J., FRENZ H., LAPPE M.: Estimation of Detection Thresholds for Redirected Walking Techniques. *IEEE TVCG* 16, 1 (2010), 17–27. 2, 3
- [SCFW10] SUMA E. A., CLARK S., FINKELSTEIN S. L., WARTELL Z.: Leveraging Change Blindness for Walking in Virtual Environments. *IEEE VR Workshop on Perceptual Illusions in Virtual Environments* (2010), 10. 1, 2
- [SKFB11] SUMA E. A., KRUM D. M., FINKELSTEIN S., BOLAS M.: Effects of redirection on spatial orientation in real and virtual environments. *3DUI 2011* (2011), 35–38. 1, 2
- [Su07] SU J.: Motion Compression for Telepresence Locomotion. *Presence: Teleoperators and Virtual Environments* 16, 4 (2007), 385–398. 2, 3
- [UAW\*99] USOH M., ARTHUR K., WHITTON M. C., BASTOS R., STEED A., SLATER M., BROOKS F. P.: Walking > walking-in-place > flying, in virtual environments. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99* (1999), 359–364. 1
- [WNR\*07] WILLIAMS B., NARASIMHAM G., RUMP B., MCNAMARA T. P., CARR T. H., RIESER J., BODENHEIMER B.: Exploring large virtual environments with an HMD when physical space is limited. *APGV '07* 1, 212 (2007), 41. 2
- [XLW\*10] XIE X., LIN Q., WU H., NARASIMHAM G., MCNAMARA T. P., CARR T. H., RIESER J., BODENHEIMER B.: A System for Exploring Large Virtual Environments That Combines Scaled Translational Gain and Interventions. *Symposium on Applied Perception in Graphics and Visualization* 1, 212 (2010), 65–72. 2
- [ZBHW13] ZMUDA M., BACHMANN E., HODGSON E., WONSER J.: Improved Resetting in Virtual Environments. *IEEE Virtual Reality* (2013), 91–92. 2
- [ZWBH13] ZMUDA M. A., WONSER J. L., BACHMANN E. R., HODGSON E.: Optimizing constrained-environment redirected walking instructions using search techniques. *IEEE TVCG* 19, 11 (2013), 1872–1884. 2