Redirected Walking in Obstacle-Rich Virtual Environments

Ben J. Congdon

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

November 14, 2022

I, Ben J. Congdon, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Virtual reality (VR) systems allow a user to explore virtual environments (VEs) intuitively by linking display and interaction to physical movements in the real world. Walking is an intuitive method of traversal, but challenging for VR; VEs do not need to match physical spaces, so users may encounter obstacles in the real world. Redirected walking (RDW) is a technique which remaps a user's physical walk onto a subtly different virtual path. The user then unknowingly adjusts their physical path to account for the change. With carefully selected transformations the user can be steered away from physical obstacles, allowing free walking in the VE. However, state of the art RDW techniques still require a large amount of physical space.

The work in this thesis aims to reduce physical space requirements for RDW techniques. Certain RDW tasks such as infinite straight-line walking require large amounts of physical space due to perceptual limits. However, VEs which contain obstacles may not contain long straight paths and can be analysed to provide useful information about future user walk directions. This research therefore focuses specifically on the application of RDW in obstacle-rich VEs to small physical spaces.

We present the following contributions on this theme: (1) MCRDW, a gain selection algorithm for RDW which uses simulated walks to anticipate future user trajectories, (2) a psychophysical study on tolerance to rate of gain change, the results of which indicate that slow gain change is significantly harder to detect than sudden gain change, and (3) Shared Spaces, a multi-user technique to allow users to share spaces virtually while allowing real walking locally.

Impact Statement

The simulation based approach of MCRDW is effective and simple to implement. The results of our experiment on rate of gain change suggest that users are highly sensitive to sudden changes, and therefore that the use of a smoothing term is beneficial for RDW performance and user comfort. These contributions have a potential impact wherever RDW implementations are used. Shared Spaces are more complicated to implement, but may have an impact on shared mixed reality where the problem of sharing environments is felt most keenly.

Primarily, though, this thesis is intended to be a contribution to the world of redirected walking and virtual reality research. The hope is that future scholars will find portions of this document useful in some way to their own work, and that the field continues to develop and grow.

Acknowledgements

My supervisor ANTHONY STEED, a constant source of good advice without whose superhuman patience and support a certain document (this one!) would not exist. I am enormously lucky to have had him as my supervisor.

My colleagues, past and present, in the VECG group at UCL. In particular, DREW MACQUARRIE, DAVID SWAPP, DAVID WALTON and SEBASTIAN FRISTON, who all taught me how to be a better researcher in the lab and out.

My brothers TOM and SAM, both the best of the three. My partner and best friend KLARA, whose enthusiasm at every small step towards this thesis meant more than she knows. My father, RICHARD, who sees clearly through the muddle. My mother, JANE, the strongest person I've ever known.

Research Paper Declaration

Parts of this thesis are based on work already published elsewhere.

Chapter 4 is based on the paper ‘Sensitivity to Rate of Change in Gains Applied by Redirected Walking’ published in the 25th ACM Symposium on Virtual Reality Software and Technology (VRST ’19) [1] by the Association for Computing Machinery (ACM). The work was subject to academic peer review. The authors contributed equally and retained the copyright to the work.

Chapter 5 is based on the paper ‘Merging Environments for Shared Spaces in Mixed Reality’ in the 24th ACM Symposium on Virtual Reality Software and Technology (VRST ’18) [2] by the Association for Computing Machinery (ACM). The work was subject to academic peer review. The authors contributed equally and retained the copyright to the work.

But tonight I say, we must move

forward, not backward;

upward, not forward;

and always twirling, twirling, twirling towards freedom!

KODOS, THE SIMPSONS

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

Virtual environments are three-dimensional spatial representations that respond to user movement and interaction in real time. Virtual reality (VR) systems allow a user to experience a virtual environment in a way that feels closer to reality, where display and interaction are closer to their physical counterparts. This involves combining display and tracking technologies to translate user movement in the physical space (or track space) into movement in the virtual environment.

Typically this involves wearing a head-mounted display (HMD). The tracking system provides information on HMD position and therefore user head direction. This works well for looking around a virtual environment. However, locomotion in VR is inherently difficult. Ideally, we would like virtual movements to bear a close resemblance to a user's tracked physical movements. This reduces the potential for nausea and helps create the illusion that the user is truly present within a virtual environment. However, in the case of many VR systems, only a limited space can be tracked. For other systems, users will instead be limited by the physical space around them and cannot wander freely without meeting obstacles.

This creates the motivation for artificial methods of locomotion; approaches which match a user's movement on some intuitive level without requiring free movement in space. Navigation schemes have been proposed and are in use today which are primarily driven by head direction, or by an external device such as

a joystick. These schemes have proven to be less than ideal, increasing simulator sickness and reducing presence, subjective measures of nausea and immersion respectively. “Real walking”, i.e., physical locomotion within the track space, more closely matches the virtual counterpart, and has been shown to be greatly preferable [3].

However, true locomotion comes with the caveat that users have limited space available. This makes simulating large virtual environments difficult. For devices using this form of tracking to be widely adopted they will need to find new ways of expressing these environments through small track spaces. *Redirected walking* (RDW) is one such way [4]. This technique imperceptibly transforms the virtual environment around the user, disrupting the mapping between track space and virtual environment. Selecting the correct transformations leads to a favourable mapping, permitting exploration of virtual environments larger than the track space.

Early RDW implementations were effective but inflexible. Users were required to follow waypoints in the virtual environment, carefully spaced to allow sufficient time for redirection [4]. Recent research has focused on generalized RDW, extending the principles to allow free exploration of arbitrary environments. However, without a guaranteed user path, selecting good transformations becomes a difficult problem. Simple heuristic techniques require very large track spaces and make no use of virtual environment layout. In simple use cases (such as walking in a straight line) the limiting factor on RDW performance is human perceptual thresholds so heuristic techniques are effective. Increased performance should be possible in obstacle-rich environments which encourage frequent changes of direction. This requires techniques that are able to make use of the virtual environment to generate good transformations.

Deep learning based approaches represent the current state of the art; however, such approaches have so far demonstrated only minor improvements over heuristic techniques [5, 6]. New techniques are required to make effective redirection computationally feasible.

1.2 Contributions and Thesis Structure

The research in this thesis explores applying redirection to small physical spaces using novel redirection algorithms and psychophysical studies. This work demonstrates that there are a variety of plausible approaches to increasing the value of small physical spaces for VR. Contributions are marked in bold.

First, we survey existing approaches to locomotion in VR. We focus on the locomotion methods themselves and their effects on user comfort and environment understanding. The concept is to cover a wide range of the movement schemes in-use in VR, up to and including 'locomotion metaphors', which largely eschew physical movement. This helps to put RDW and mapping-based approaches in context. The survey can be found in Chapter 2.

Second, the **Monte-Carlo Redirected Walking** (MCRDW) algorithm is proposed. MCRDW is designed to guide users on safer physical paths, reducing the likelihood of the user leaving their physical boundary when in a smaller space. The algorithm uses simulated walks to anticipate future user trajectories. In our simulation-based experiment, MCRDW was significantly more effective at keeping users within physical boundaries when compared with existing RDW algorithms, particularly in VEs which are obstacle-rich. The theory, sample implementation, evaluation strategy and results can be found in Chapter 3.

Third, we undertake a **user study on sensitivity to rate-of-change of rotation gain**, a phenomenon which users are exposed to during RDW. The results of the experiment suggest that users are highly sensitive to this type of change, and therefore that the use of a smoothing term is beneficial for RDW performance and user comfort. Details on the user study and all results can be found in Chapter 4.

Fourth, the **Shared Spaces** technique is proposed, designed to allow remote real walking in differently shaped obstacle-rich environments. In this approach a mapping is generated between physical environments. Remote user positions are mapped across, allowing for smooth and continuous movement. As the user is not moved, no conflicts are introduced between the visual and vestibular system. The results of the accompanying study suggest that users believed the remote user was

present in their environment. The theory behind the technique and details on the accompanying user study can be found in Chapter 5.

Finally, we summarize our findings and discuss future work in Chapter 6.

Chapter 2

Survey of Virtual Reality Locomotion

Decades of work in VR in academic and private contexts has led to a broad range of approaches to locomotion in virtual reality. We categorise the work in this area under five headings: locomotion metaphors, chaperones and resetters, augmented locomotion, redirection spaces and redirected walking.

2.1 Locomotion Metaphors

This branch of techniques emulate locomotion while keeping the user static. These are some of the earliest methods proposed, and continue to be used due to space limitations. Typically the techniques in this section are used alongside physical walking as far as available physical space allows.

2.1.1 Flight-Like

The simplest locomotion metaphors are those using joysticks, "wands" or gaze. Respectively, these techniques move move the user in the direction they push a joystick, point and look. Of these, gaze based movement has been shown to be the least preferable from the perspective of task performance [7, 8]. More generally all three have been shown to have similar downsides; there is significant evidence of a negative impact on both presence and simulator sickness when compared with true locomotion [3, 9]. Additionally, these metaphors have been shown to increase the time taken and the number of errors generated when users are tasked with solving simple tasks in a virtual environment due to increased cognitive load [10, 11].

The techniques discussed so far have been rate-controlled; that is, the user

selects a direction of movement and travels in this direction in the virtual environment. The alternative is position-controlled movement, where a point in the virtual environment is mapped directly to a point in the track space. Position-control can be more precise, as the user has fine-grained control over the speed of movements. *Eyeball-in-Hand* and *Scene-in-Hand* [12] are two position-controlled techniques where the camera position is mapped to the user hand position. *Fingers-as-Legs* [13] uses finger movements as a locomotion metaphor in which users control footfalls by "walking" across a touchpad with their fingertips. In practice, these techniques may be overly artificial. In the accompanying user study, joystick-based users accomplished the given tasks faster, and the researchers were unable to establish a significant impact from *Fingers-as-Legs* on ease of use [13].

Intuitively, these techniques can feel more like flight than locomotion, as they provide optical flow information from the virtual environment without accompanying information from the user's proprioceptive system. Reductions in field of view when turning can be used to reduce the conflict [14]. Stepped turning is also notable and currently popular. Here the user is turned immediately rather than smoothly, and in increments of approximately 45 degrees. Both methods are innately artificial, but their impact on presence has yet to be fully investigated. More generally, user studies have shown that reducing field of view is harmful to presence [15].

Bhandari et al. found that in a virtual environment where users were able to easily switch between input methods, those who switched to "flight-like" input were unlikely to return to real walking [16]. This suggests users may find value in lower-effort input techniques despite increased artificiality. However, these same users were more likely to report reduced presence [16].

2.1.2 Teleportation

Teleportation allows a user to move immediately to a targeted location in the virtual environment rather than walking. Teleportation points may be user-selected or limited to pre-defined waypoints. Novel interaction methods may be used to select the next teleportation point. Pausch's *Worlds in Miniature* [17] concept, for example, has the user manipulating and selecting waypoints from a three dimensional scale

model of their environment.

Teleportation deals with the conflict between optical flow and the proprioceptive system by limiting optical flow information; the user merely appears at their destination, or travels there faster than would ever be physically possible through locomotion. This theory appears to work in practice, as teleportation has shown to be effective at dealing with simulator sickness [18], and is a popular technique for virtual reality experiences currently. However, teleportation is not suitable for those experiences where rapid user movement is undesirable.

Most significantly, teleportation has been shown to have a negative impact on environment recollection and long-term navigation when compared with real walking [7, 19, 9], though there is some evidence the effect may be small [20]. Attempts to ease teleport transitions by fading the user's view or by performing large teleportations in small steps were unable to demonstrate an improvement in user comfort or navigation for either approach [21].

2.1.3 Walking-In-Place

Walking-in-Place (WIP) [22, 23] takes a different approach to the conflict between optical flow and proprioception. Here user movements are tracked such that movement in the virtual environment requires some physical action resembling true locomotion. In theory, this stimulates the vestibular system to provide proprioceptive cues, reducing the conflict with optical flow.

Implementations of WIP can vary based on body part tracked, and those types which only require a tracked head ("head bobbing") or hands ("arm-swinging") are widely used in commercial VR. When the term WIP is used in the literature, it often refers to some form of tracked leg or waist movement. Nilsson et al. argue that gestures typically associated with WIP cause greater strain on the user than real walking would, and propose two alternatives which require less movement [24].

WIP has been shown to be a significant improvement over other locomotion metaphors in presence, simulator sickness and spatial awareness [22, 25], though true locomotion has proven to be more desirable on all counts [3, 25]. Recent approaches to WIP have experimented with more accurate walk-state detection

through machine learning, somewhat improving navigation performance [26].

2.2 Chaperones and Resetters

For large virtual environments true locomotion introduces the danger of a user physically leaving the track space. Chaperones are systems that indicate to a user when they are approaching a track space border. Resetters, sometimes known as reorientation techniques, are systems that allow the track space to be repositioned within the virtual environment to allow the user to continue walking. Some techniques combine both.

2.2.1 Barriers

The most straightforward chaperone system is a virtual representation of the track space boundary. *Magic Barrier Tape* (MBT) [27] extends this idea into a hybrid of chaperone and resetter. The user walks freely within the track space. When the user approaches a track space edge, MBT displays a tape graphic across the boundary. The user can then push on the tape to reposition the track space in the direction of the push. Once repositioned the user may then walk freely, regaining fine control over their position. This approach is sometimes known as hybrid position/rate control. Cirio et al. propose the *Virtual Companion* (VC) [28], a similar hybrid chaperone and resetter system. Here the chaperone bounds are represented by an entity in the virtual environment whose position indicates the nearest track space boundary. The user freely walks within the track space. The track space can be moved by controlling the VC with hand gestures mimicking a pair of "virtual reins". By altering their hand gestures, the user may control the rotation and speed of the VC.

Pushing on the barriers in MBT and using the reins with VC is flight-like, and so comes with the same caveats as joystick or wand based metaphors; in particular, that flying movement is associated with increased simulator sickness [3]. Moreover, while moving the track space with a pushing or reining motion is intuitive, MBT and VC deal poorly with experiences where the user must cover a large distance; in this case, the user will always be using rate-control and never return to the track

space for locomotion.

Williams et al. consider three techniques for resetting: *Freeze-Backup*, *Freeze-Turn* and *2:1-Turn* [29]. All techniques are triggered when the user encounters a track space boundary. *Freeze-Backup* freezes user position in the virtual environment while the user takes physical steps away from the boundary. User orientation (head tracking) is not frozen. *Freeze-Turn* freezes user orientation while the user turns physically 180 degrees. *2:1-Turn* does not freeze position or orientation, but magnifies orientation gain. When the user has turned 180 degrees physically, they will have turned 360 degrees in the virtual environment. Williams found *Freeze-Backup* led to fewer errors than the other techniques when assessing spatial awareness. Simulator sickness was not investigated. Based on previous results, we might speculate that as *Freeze-Turn* and *Freeze-Backup* create conflict between optical flow and the proprioceptive system they would cause user discomfort. Interestingly, Williams found that *Freeze-Backup* was the preferred technique by users.

An alternative approach to resetters is to apply significant distortion to the virtual environment as the user approaches the boundary, "bending" the environment such that the user cannot continue on the path that is taking them out of their physical environment [30]. This technique is of limited application. It must be possible to bend the environment to prevent further forward movement, which restricts the design of the virtual environment. Moreover, as the bend must be large, significant visual distortion will be introduced to items in the environment, and a large tracking space is required to prevent the bend being applied constantly. The technique does, however, have one key benefit in that the user is never required to stop. As such no chaperone is required.

2.2.2 Distractors

When the user approaches a track space edge, these techniques create entities inside the track space rather than at boundaries. In doing so, user attention is drawn back within track space bounds. Users therefore need not be made aware of the boundaries of their track space within the virtual environment. This alternative form of chaperoning is described by Peck as distractors [31]. Grechkin et al. expand on

the idea with *Rotate-and-Walk* [32], where tasks are created that require to user to physically walk towards a point in the track space. Both techniques are designed to operate as resetters when combined with redirected walking (see Section 2.5), as increased user movement allows that technique to function more effectively.

The distractions employed cannot be too subtle, or they will fail to draw the attention of the user; Peck found that distractors needed to be combined with more traditional track space boundary visualisations for these cases [33]. Ideally a distraction will be a seamless part of the virtual environment, and will therefore need to be designed by environment creators. This is particularly challenging for *Rotate-and-Walk*, due to increased interactivity. Peck also found that frequent use of distractors was irritating for users [33]. This form of chaperoning is therefore better suited to larger track spaces where track space boundary collisions events are less common.

Significant research has found that the concept of interpersonal space extends to virtual reality [34]. Neth et al. propose using avatars as distractors to exploit this using virtual human figures [35]. This technique will not be appropriate for many types of experience, but may represent a more natural chaperone method. As with distractors frequent interruptions may be irritating so the technique may be best suited for large track spaces. An additional problem is that avatars which are created in view of the user will appear very artificial. Instead, they must be created out of the view frustrum and sent across the user's path. This requires an accurate estimation of the user's path along with a large track space to give the avatar time to meet the player.

2.3 Augmented Locomotion

These techniques allow a user to physically walk while introducing non-immersive design features into a virtual environment to ensure a user will never need to walk further than the track space allows.

2.3.1 World Scaling

Bruder suggests *Arch-Explore* [36], a technique for real locomotion in architectural exploration through scaling the world around the user and teleportation. These transformations are triggered by passing through "portals" into different areas of the environment. Though these interactions do not have a physical counterpart, teleportation and scaling can feel intuitive, and there is little evidence of a negative effect on presence or simulator sickness. However, these techniques are highly prescriptive in terms of the environments they allow us to design, and are therefore inappropriate for many experiences.

LaViola proposes a more general approach in *Step WIM* [37], analogous to *Worlds in Miniature* [17] but here the miniature world is created at the user's feet. It can then be walked around with true locomotion, albeit greatly magnified. When the user is satisfied with their position, they can teleport to where they are standing on the model and return to standard locomotion. This is an elegant solution that is suitable for many types of environment and is not as prescriptive as *Arch-Explore* above. However, the concerns raised over teleportation above also apply here, and it will not be suitable for those experiences where teleportation is undesirable. As the technique allows unrestrained locomotion, it risks the user leaving the track space and will therefore need to be combined with a chaperone and resetter.

2.3.2 Locomotion Scaling

Interrante et al. propose *Seven League Boots* (SLB) [38]. This approach magnifies physical locomotion to cover a greater proportion of the virtual environment through locomotion. The magnification is applied intelligently; only movement in the user's intended direction is magnified so as not to exaggerate head sway. The user's intended direction is identified by a weighted average of previous movements and gaze direction.

Translation gain is a truly environment agnostic approach, but must be applied with care when a user is stationary. Modern consumer virtual reality tracks user movement through head position. Gain may therefore be disorientating when a user is closely inspecting objects or looking around the environment. SLB exacerbates

this problem, as the heuristic used for path estimation makes the gain vary unpredictably when the user is not walking in a straight line. The technique also risks the user leaving the track space so requires a chaperone and resetter.

The problem of perceptual "jitter" with high translation gain factors (10x or more) was identified by Williams, who notes that users found the effect particularly disorientating while stationary and performing tasks [39]. Williams suggests non-linearly scaling translation using a ramp function to reduce the effect at low velocities. Users reported a greater sense of control and reduced nausea when a ramp function was used. However, it should be noted that non-linear scaling disrupts the mapping between the track space and virtual environment. The effect is incremental and does not self-correct, so over time greater areas of the virtual environment may become unreachable. Williams observed this effect during the user study, with some participants unable to reach the target objects due to poorly mapped track space. While SLB also has affects this mapping, the result is less disruptive as movement perpendicular to the walking direction is typically oscillatory. Non-linear gains may therefore be most useful when combined with a technique such as redirected walking (see Section 2.5) which manipulates the mapping between track space and virtual environment and is therefore capable of correcting poor mappings.

If we assume accurate identification of leg movement and head movement, SLB constitutes a practical way of applying translational gain to a user in a virtual environment; concerns arise when considering the effects of that gain. SLB as described in [38] scales user locomotion by a factor of 10. Williams places the upper bound at a factor of 100, observing that any further gain significantly disrupts user spatial awareness [39]. The level of gain used in SLB is described as comfortable by users but is certainly perceptible; Steinicke et al. [40] put the perceptual threshold for translational magnification at 14%. The methodology used to acquire these results applies the work of Interrante et al. [38] to prevent magnification of head sway. Were we to keep gain to an imperceptible level, the size of the walkable area would not be significantly increased.

2.4 Redirection Spaces

The following techniques allow real walking in large virtual environments that seem plausible and fit within the user's track space, but which could not exist physically. These environments are generated or altered dynamically as the user traverses the space based on two key observations. Firstly, that humans are insensitive to surprisingly large environment modifications as long as those modifications are not directly observed [41]. This phenomenon is known as "change blindness". And secondly, that cognitive maps of environments are often hierarchical or categorical rather than spatial [42]. For example, this means we may remember that two rooms are connected without remembering where the rooms are in relation to each other.

Some of the techniques in this section attempt to make use of change blindness by distracting the user when making changes. Remarkably, this may not be required, as eye movement such as blinks and saccades have been shown to mask scene changes [43, 44]. Steinickie et al. found that the effect extends to stereoscopic systems [45]. Exploiting these eye movements requires effective eye tracking to identify saccades and blinks, but may allow more dynamic scene alterations with overt distractions.

Change blindness has proved to be hard to detect in the relatively simple environments designed for user studies. One possible concern is that spatial understanding and navigation ability appear to improve with the frequency of landmarks (i.e., environmental features) [46]. This could lead to reduced effectiveness in the feature-rich virtual environments more commonly found outside the lab.

2.4.1 Impossible Spaces

Suma et al. propose *Impossible Spaces* [47] in the form of self-overlapping architecture. Here users explore a large virtual environment with many rooms off one central corridor. Users are instructed to enter every room and complete a short task. While users are in a room, the door and corridor outside are rotated 90 degrees to run parallel to an adjacent wall. The door to the next room is only a short distance down the corridor and is orientated to significantly overlap the previous room, thus having the user walk back into the center of the track space. When applied in series,

the user is lead around the edge of the track space indefinitely. Suma found that the scene changes went undetected by most participants while they were lead around a 2352 sq. ft. environment with a track space of only 196 sq. ft..

However, *Impossible Spaces* require a carefully designed environment. Each scene change needs to be hand-designed to suit the virtual environment and the user's track space, or the user will not be able to traverse the environment at all. Moreover, users must be co-operative; in the example above, were the user to not enter every room, they would encounter the track space boundary.

2.4.2 Flexible Spaces

Flexible Spaces [48] somewhat generalise the concept of *Impossible Spaces*. Rather than a full virtual environment, designers create individual rooms. *Flexible Spaces* places those rooms within the track space, overlapping if necessary. Rooms are then connected with procedurally generated corridors. These corridors are long and typically have 3 right angle turns to discourage spatial mapping. This approach does not permit contiguous elements (rooms or adjacent collections of rooms) which are larger than the user's track space. Strict control of the layout of a virtual environment is also not permitted. Together these factors rules out many potential applications.

2.4.3 Mapped Spaces

Sun et al. [49] propose a technique for steering users around large virtual spaces through the use of pre-generated static mappings between the physical and virtual spaces. The mappings are used to alter the rendering of the virtual environment, guiding users on the now warped path.

The goal of the techniques remains to allow traversal of large virtual environments from small tracked spaces. Intuitively, the mapping to virtual space must therefore be folded into the tracked space multiple times. The resulting virtual to physical mapping is globally surjective and locally injective. That is - multiple points in the virtual environment can correspond to the same point in the physical environment, but it is possible to disambiguate and identify the meaningful virtual

point by making use of the user's current position in the virtual world.

As with *Flexible Spaces*, this approach can be considered an automated application of *Impossible Spaces* as the large environment becomes folded many times into the smaller, physical environment. The aim is to prevent the user noticing that a space of this size would overlap. Unlike *Flexible Spaces*, this technique places no restrictions on environment design.

Though the technique is applicable to any virtual environment in theory, in practice this can come at the cost of visual distortion after warping and re-rendering. Dong et al. [50] propose *Smooth Assembled Mappings*, a divide-and-conquer approach to mapping designed to reduce this problem. Here the virtual environment is split into patches which are then individually mapped to avoid areas of excessive local distortion. Cao et al. [51] note that while some elements of an environment cope well with distortion (corridors, streets) areas of high visual detail ideally should not be warped at all. Their proposed solution generates visual feature maps and incorporates them into the virtual to physical mapping step. The authors were able to show that for their test environment, average geometric distortion across the entire map was reduced when compared with *Smooth Assembled Mappings*.

As a group these approaches have some significant advantages. Most importantly, through the static mapping process these techniques can guarantee that a user will never encounter a track space boundary and never need resetting. This is in stark contrast to those approaches which dynamically alter the mapping between virtual and physical spaces like locomotion scaling or redirecting walking (see Sections 2.3.2 and 2.5 respectively) where such a guarantee is never possible, even in large track spaces. Static mapping based techniques do, however, come with their own disadvantages as the level of visual distortion can be significant. Limiting distortion to an imperceptible level, simple for dynamic techniques, is a challenge for static maps. Constraints can be applied on the optimisation, but mappings are typically best-effort; a mapping that fully satisfies for a given environment pair may be difficult to find, or may not exist at all. Solutions remain an open area of research.

2.5 Redirected Walking

Redirected Walking (RDW) techniques imperceptibly move the virtual environment around a user as they walk. The two common forms of redirection are translation gain, which manipulates the the user's position, and rotation gain, which manipulates the user's orientation. By combining these factors and applying with varying magnitudes, we can control the mapping between the track space and the virtual environment. The right mapping allows the user to freely walk within the virtual environment while being subconsciously steered within the track space bounds. We can clearly distinguish good mappings from bad, as good mappings will never allow the user to leave the track space. However, choosing redirections to accomplish good mappings at any given time is a difficult task as it relies on the user's past, present and future positions and orientations. In this document we will refer to this as the redirection problem (RDP).

2.5.1 Perceptual Thresholds

In the original RDW pilot study, values for linear and rotational velocity scaling and baseline rotation were set at levels individual users found to be imperceptible [4]. Later work establishes more general thresholds based on psychophysical responses to redirected walking.

Steinicke et al. [40] ran a psychophysical study where users were asked to identify whether redirection (gain) was currently being applied. The results indicated that for users to distinguish gain from no-gain with 75% accuracy, their physical rotation could be scaled up 25% and scaled down 33%. Likewise, translation through linear velocity could be scaled up by 26% and down by 14%. For users to believe they are walking in a straight line while really walking in a circular arc, the radius of that arc must be no less than 22m.

Bruder et al. [52] later reproduced the psychophysical tests performed by Steinicke with similar results; again working at a 75% detection accuracy, thresholds were found to be approximately 26% up, 32% down for rotation and 29% up, 13% down for translation.

Both Steinicke and Bruder also investigated the effect of curvature gain [40,

52]. This is rotation gain applied while walking to guide the user on a circular path. Here there is a significant discrepancy, despite similar methodology: Steinicke found the radius of the arc ought to be no less than 22 meters while Bruder puts the figure at 15 meters [40, 52]. Regardless, these are important results as they provide the size of track space required for infinite walking in one direction. This particular use case is unlikely to be bettered by any improvement in RDW techniques, as it relies only on perceptual thresholds.

Langbehn et al. performed a similar psychophysical study to Steinicke and Bruder but with curved virtual and physical paths of varying degrees where the original study used a straight virtual path. The results suggest that thresholds could be significantly increased when walking on tightly curved virtual paths. With a 75% detection threshold, gain thresholds were 3.25 for the real path with a radius of 1.25m and 4.35 for the real path with a radius of 2.5m, leading to virtual curvatures of 4m and 10.875m respectively [53].

Zhang examined the effect of rate of gain change on perception thresholds and found no significant difference between gradually changing rotational gains compared to instantaneously changing the gain during 360 degree turns [54]. Neth et al. found that sensitivity to rotation gains increases at higher walking speeds; that is, that lower walking speeds allow greater rotation gains [35]. Interestingly, these results run counter to typical RDW implementations, which use a smoothing term to avoid instantaneous gain changes and increase rotational gain with walking speed [55, 56].

Audio information can be spatialised by modifying for user position and orientation, and based on the layout and materials present within the virtual environment. Serafin et al. found that with no visual stimulus, users could be virtually turned 20% more or 12% less than their physical rotation through spatialised audio cues [57]. However, Nilsson et al. found that when both audio and visual information is available, redirection detection thresholds were the same for spatialised, static and disabled audio [58]. This suggests redirection is possible through spatialised audio but visual information appears to take precedence for the proprioceptive system.

Eye movements such as blinks and saccades have promising applications for real locomotion in virtual reality through change blindness (see Section 2.4). Applications to RDW have similarly encouraging results. Bolte and Lappe established perceptual thresholds for redirection during saccades, finding that users could be virtually displaced up to 0.5 meters and rotated up to 5 degrees during saccades with an amplitude of 15 degrees (relatively large eye movements) [59]. Nguyen and Kunz found similar results for redirection during blinks with a threshold around 9 degrees [60]. These results are particularly interesting as they do not rely on magnification of current movement, thereby allowing us to apply significant redirections even while the user is undergoing other RDW transformations or standing still.

2.5.2 Waypointed Redirected Walking

In its original form as proposed by Razzaque et al., RDW guides the user through a series of waypoints in both the virtual environment and the track space [4]. Only rotation gain is used. The magnitude is adjusted dynamically based on user movement. The final rotation applied is the maximum of three factors; scaled linear velocity, scaled rotational velocity and a baseline rotation per second. The direction of this rotation is calculated to send the user towards the next waypoint in the track space based on the user direction in the virtual environment, assumed to be the direction towards the next virtual environment waypoint.

When the user must walk between waypoints, their future position and orientations are known and RDP is greatly simplified. Azmandian et al. propose treating redirection selection as an offline optimization problem [61]. Redirections are discretized in intervals, while the route is broken down into pairs of waypoints. Every combination of redirections is applied in series to the steps of the waypoint route to form a tree. Interleaved pruning operations are used to avoid exponential complexity. Redirection combinations are evaluated through a depth first search of the tree, and the combination that minimises track space boundary collisions is selected. The chosen redirections are then applied to the user as they follow the waypoint path. When supplied with a graph rather than a series of waypoints, the optimization technique can be applied multiple times to evaluate all possible routes around the

environment, finding the route which fits best within the track space.

Waypoint techniques can come close to globally optimal redirection when users do not deviate from the waypointed path in position or orientation. This is relatively unlikely; Azmandian notes that some deviation is inevitable, and suggests a method for partially recovering [62]. Waypointed RDW is an interesting way to explore RDP and provides an insight into the potential of RDW due to its near optimality. However, waypoint walking is ultimately inflexible, and cannot solve the goal of real walking in virtual reality.

2.5.3 Generalised Redirected Walking

In the general case, future positions and orientations are uncertain as the user is free to move as they wish. This is known as generalised [63] or generic [64] RDW. Generalised RDW significantly complicates RDP. Early algorithms make heavy use of heuristics; *Steer-to-Center* (S2C) redirects the user to the center of the track space, *Steer-to-Multiple* (S2M) redirects the user to the closest of multiple targets near the center of the track space, and *Steer-to-Orbit* (S2O) redirects the user to a circle around the edge of the track space [65, 55]. S2C has proven to be the best performing technique when users change direction frequently, while S2O is more effective when users walk long distances in a straight line [56], and in constrained environments [66].

Heuristic techniques are simple to compute, but frequently choose sub-optimal redirections. S2O assumes that the user will never change direction, while S2C assumes that the user may change direction at any time with equal likelihood for each direction. This ignores valuable sources of data; for example, walls and obstacles in the virtual environment provide bounds on walkable space. In constrained environments S2C and S20 waste a great deal of the track space on these unwalkable areas; despite greater available information on user path, they perform no better in constrained environments than open ones [66].

Zhang and Kuhl develop heuristic approaches by increasing the number of heuristics to three [67]. First, that virtual and track space boundaries should match as closely as possible (i.e., the intersection of the virtual environment and the track

space should have the greatest possible magnitude). Second, that small gains are preferable to large gains, as they are less disruptive to the user. Third, that users should have as much open space in front of them as possible. One weakness of this approach is the evaluation of heuristics is performed frame-by-frame. As there is no heuristic valuing consistency in strategy, redirections may change frequently, adversely affecting performance. Likewise the heuristics selected often conflict, particularly heuristics two and three. There is no clear method for weighing them against each other without some understanding of how they will affect the user in the near future.

Further extended heuristic approaches to S2C also exist in the literature, such as using translation gain to extend walkable distance when heading directly away from the center point [68]. APF-RDW uses artificial potential fields (APFs) to generate a steering target rather than using the center in order to create a heuristic approach suitable for more complicated physical environments [69]. Push/pull reactive (P2R) redirection combines both APFs and translation gain [70].

More recent techniques attempt to approximate RDP in real-time. The typical approach used is to view RDP as a optimization problem as with Azmandian's waypointed redirection [61], where the space of all possible combinations of redirections and user paths is covered via depth-first search. Algorithms differ in terms of how possible user paths are enumerated (path predictors, see Section 2.5.4) and how mappings are evaluated. Zmuda et al. propose FORCE [71], decomposing the virtual environment into a graph based on "decision points" where user turns are possible. Redirections are evaluated at the leaf nodes of a depth-first search. FORCE is run again as the user reaches each new graph node to create an updated redirection. It is noted that a stochastic path predictor would be required, though none is proposed. This would help to weigh branches of the tree against each other by assigning probabilities to user paths. Nescher et al. propose MPCRed [72], a similar approach based on model predictive control. MPCRed again decomposes the virtual environment into a graph. Redirections are evaluated in a process analogous to depth-first search, but at all nodes rather than leaf nodes alone. At decision

points, smoothed walking direction is used as a path predictor. Both algorithms use a fixed maximum depth to prevent exponential complexity growth.

The value of decomposing to graph form is highly environment dependent (see Section 2.5.4). However, the innate exponential complexity of both techniques makes more accurate path prediction difficult. Additionally, solving RDP as an optimization problem is difficult as the space is prone to local minima. This makes it a poor fit for optimization techniques from elsewhere in the literature that rely on geometric properties of the surface, such as hill climbing or gradient descent. The ideal cost function may mitigate this problem. Though both techniques note that a cost function is required, neither propose one. The simplest cost function is the number of track space boundary collisions for a set of redirections. This is also the only function that can generate entirely reliable results; other functions are an approximation of the boundary collision function with an infinite search horizon. However, the boundary collision function is less useful when applied with small horizons as it is insufficiently granular; in the short term, reasonable redirections might all lead to no boundary collisions. We might also wish to add a cost to redirections based on the magnitude employed to improve user comfort. Engel et al. discuss a cost function that takes both performance and comfort factors into account [73].

Very recent techniques attempt to solve the redirection problem with machine learning. These approaches treat the task of calculating the correct redirection strategy as a control problem. *Steer-to-optimal-target* (S2OT) uses reinforcement learning to train a model using the Deep Q-Learning approach [5]. The system divides the tracked or physical space evenly into squares and places a target at each intersection point. For each intersection, the predictor is run to determine the outcome were the user to be redirected with the point as the target. The goal function balances the likelihood of a physical collision against the amount of rotational redirection required to redirect towards the target.

Strauss et al. also treat the redirection task as a control problem and propose *Steering via Reinforcement Learning* (SRL) [6]. While S2OT which discretizes the space into a reduced set of targets, SRL instead treats the physical space as

continuous. Like S2OT, SRL uses a Deep Q-Learning approach, though SRL uses a variant known as Proximal Policy Optimisation (PPO). PPO does not directly estimate resultant state but rather an indicator of outcome ("advantage").

S2OT demonstrated improvements in both simulation and user study, reducing collisions by approximately 15% [5]. This performance improvement comes at the cost of higher rotation gain; the goal function weighting in S2OT favours reducing physical collisions over keeping redirection levels low. Overall, S2OT applies significantly more rotational gain than other approaches and 30% or more over S2C.

SRL was able to increase mean distance covered by around 4% on simulated paths when compared with S2C. No significant difference was observed on real paths. The overall level of rotation gain applied was not directly compared. The authors argue that the performance discrepancy when compared with S2OT can be explained by different simulated and real paths, as the paths used for testing SRL were long and straight with only occasional turns.

In practice differing environments will lead to differing performance with machine learning approaches as with optimisation based approaches. A redirection strategy that is effective in one environment may not be the most effective for another. S2O, for example, is optimal for straight-line virtual walking with a large physical space [56], but otherwise ineffective [66]. Comparing redirection techniques directly is difficult as a result. With this caveat, the above machine learning approaches can demonstrate an improvement over heuristic techniques: both S2OT and SRL are shown to reduce collisions when compared with S2C. The effect size is, however, relatively small for both techniques.

2.5.4 Path Predictors

Redirections are contingent on user movement so even the simplest RDW algorithms implicitly rely on some understanding of future positions. For example, S2C and S2O attempt to turn the user towards the center or the orbit of their track space based on the assumption that in the very short term the user will walk forward. Hollands et al. found that this assumption holds around 70% of the time [74]. However, there are user tasks that lead to frequent gaze changes, such as searching for an ob-

ject or exploring an environment. In these cases, gaze-based path prediction causes frequent changes of redirection strategy which harms performance [33]. Effective redirection selection during user orientation changes is particularly important as this is when we can apply the greatest magnitude of redirections without exceeding perceptual limits [40]. Optimization based RDP solvers such as *FORCE* can use a path predictor to weigh alternatives by importance, and for early culling of unlikely paths. More sophisticated models of motion known as path prediction are therefore of use to a wide range of redirection techniques.

The simplest path prediction techniques provide only the user's most likely walk direction, while more sophisticated techniques provide a range of possible movement directions with probabilities. We will call these deterministic and stochastic path predictors respectively. Nescher identifies three classes of path predictor by interval [75]: "short term" covers where the user will next step, with a range in the milliseconds. Next, "medium term" prediction is in the range of seconds, covering user deviation due to gait and head movement but not considering obstacles and path-finding. Finally, "long term" covers the range of many seconds to minutes, and considers strategic movement around obstacles and through an environment based on the user's cognitive map and their goals.

For a number of reasons, RDW is most concerned with medium to long term path prediction. Primarily this is because redirection is most effective when strategies are only changed infrequently, as perceptual limits make it impossible to redirect the user to a point in the track space over the short term. Additionally, RDW with medium and long term path prediction can introduce redirection during orientation deviation caused by gait and head movement. A short term path predictor interprets these movements as a change of path, potentially switching redirection strategy.

Peck describes the three typical forms of path prediction [33]. First, interpreting the user's orientation as their predicted direction, as with S2C. Second, interpreting the user's current walking direction as their predicted direction. Walking direction is calculated through positional changes between readings and smoothed by

averaging results over a short window. Third, gathering predicted direction through a graph of the virtual environment. The user's position is matched to the closest node by distance. The user's walk direction (calculated as above) is then compared against the direction to each neighbouring node. The best match is the predicted direction.

For reasons already discussed, user orientation is an unsatisfactory method of path prediction for RDW, though likely to be accurate in the shortest term. Nescher found the user's smoothed walk direction to be an accurate path predictor in the short term [75]. However, it cannot predict a stationary user's path. Typical implementations fall back on user orientation in this scenario, with the aforementioned poor performance characteristics. Matching user path to a graph of the environment is the longest term predictor discussed. However, relying on the user's walk direction (and therefore orientation when stationary) makes the technique sensitive to short term noise. The form discussed by Peck is deterministic; however, it is simply extended to a stochastic form by weighting alternative nodes by similarity to the user's walking direction.

Using connected graphs to represent walkable space does not permit effective redirection around environments with open areas larger than the track space. However, the innate exponential complexity of both techniques makes the alternatives difficult. A path predictor which provides probabilities for movement in any direction would generate a large number of possible routes quickly, further reducing the search horizon. An additional issue is that these techniques use distance to nodes to infer the user's current position on the graph. In obstacle rich environments it would be possible to incorrectly associate the user with an unreachable node, degrading performance. This problem could be mitigated by using multiple nodes per walkable area; however, this would increase the node count and the degree of connectivity, reducing the search horizon. Ultimately, graph form works well for minimally connected environments with little open space, such as the series of connected corridors used in [71] and [72], but may be a poor fit for general virtual environments.

Zank et al. propose a method for medium term path prediction which makes use of both user position and gaze. The technique assigns probabilities to points of interest dynamically using Bayesian inference [76]. Calculating appropriate Bayesian priors is not trivial; to keep the task simple Zank et al. discretize the space coarsely, with somewhat similar results to graph-based prediction. The method requires points of interest to be identified within the environment and assumes goal-oriented movement only between points of interest. This may be a poor fit for some user tasks such as exploring or searching, where points of interest are unknown and movement is not goal-oriented.

An alternative approach is found in the literature on telepresence. Su et al. describe a medium term curve-fitting path predictor based on second order regression of the user's path history [77]. The future path can then be approximated through extrapolation of the curve. This is an interesting technique, though its effectiveness when applied to RDW is still an open question. The approach is difficult to extend to a stochastic form, and as the curves are not generated with reference to a locomotive model, they may bear little resemblance to real human movement.

Human locomotion follows predictable patterns when it is goal-oriented [78]. Models of locomotion can be used to generate plausible paths under these conditions. Zank et al. propose comparing user path history with these machine-generated paths; the best matching generated path then becomes the path prediction [79]. A number of models exist for generating reasonable paths in this scenario [80, 81, 82, 83, 84]. As noted, many user tasks do not involve goal-oriented movement, so these models do not fully represent human walking behaviour. However, the models have been shown to better resemble human locomotion and to provide more accurate path predictions when compared with simple straight line paths between goal points [79]. This is a more sophisticated technique for inferring future user paths through movement history than simple positional smoothing or curve fitting. It is also stochastic; generated paths can be weighed against each other in order of best fit. In its current form, however, it is insufficiently general; all possible paths must be known prior to generation, and ideally a goal point should

be known to prevent the exponential growth of paths.

Many of the path prediction methods discussed so far have made use of a graph representation of the virtual environment, either to represent points of interest or walkable areas. However, graphs are only infrequently used in modern virtual experiences for representing walkable space; typically this is achieved through use of a navigation mesh. As with meshes for rendering, a navigation mesh is able to represent a surface through a series of faces, implying connectivity through shared edges. Techniques exist for automatically generating navigation meshes for virtual environments [85]. Navigation meshes are most commonly used for agent pathfinding. However, as they better represent space than connected graphs, they should allow more accurate path prediction. Azmandian et al. propose a method for generating fixed depth path prediction graphs from navigation meshes dynamically, adapting the graph around the user position [86].

2.5.5 Cognitive Impact

Many metaphors for locomotion have been shown to place an increased cognitive load on the user, interfering with concurrent tasks and navigation performance [10, 11]. RDW is dissimilar to these techniques in that redirections are applied imperceptibly. Hodgson et al. found no significant impact on verbal and spatial cognitive task performance when redirections were kept below perceptual limits [63]. There is also evidence that user walking speed has a negative correlation with translation gain. That is, at low gains users walked faster, and at high gains users walked slower [87]. These are encouraging results, and suggests that imperceptible redirections are being unconsciously adjusted for naturally through gait. Bruder et al. found supporting results, though also noted the impact of exceeding perceptual limits with redirections: lateral sway in the user's gait (a measure of unsteadiness) was found to significantly increase, and cognitive task performance was found to significantly decrease [88].

Suma et al. report similarly encouraging results for spatial orientation after redirection [89]. In the user study, track space targets were matched 1:1 in a virtual environment. Users were asked to point to the targets in the track space and the

Locomotion Technique	Simulator Sickness	Cognitive Load	Overt	Modifies VE	Restricts User Path
Locomotion Metaphors	X	X	X		
Chaperones & Resetters		X	X		
World Scaling		*[a]	X		
Locomotion Scaling	X	X	X		
Impossible Spaces				X	
Flexible Spaces				X	
Mapped Spaces	*[b]	*[b]			
Waypointed RDW					X
Generalized RDW		*[c]			

[a] Missing data. Could be similar to 'teleportation' metaphor.
[b] Enforcing imperceptibility increases already difficult problem of generating mappings. Current approaches from the literature are not subtle.
[c] Much work has shown that RDW can be subtle and have little cognitive load impact. However, with generalized RDW, boundary collisions are a possibility, and will likely have an impact.

Table 2.1: Overview of virtual reality locomotion techniques in the literature

virtual environment, then the virtual targets were hidden and redirection applied. Users were asked to point at the now hidden virtual targets. Suma et al. found that pointing error after redirection was similar to the error in the control condition where no redirection was applied; i.e., users pointed to the virtual target rather than the track space target. Hodgson et al. report the same findings [63]. This suggests imperceptible redirections are subconsciously integrated into a user's spatial orientation.

2.6 Summary

Previous work in the areas of chaperones, resetters, locomotion metaphors and augmented locomotion has provided methods for traversing space in virtual reality safely and without significant simulator sickness. These techniques have some useful properties. Locomotion metaphors like teleportation, for example, allow a user to explore infinitely without encountering a track space boundary. However, the artificiality of these methods makes them unsuitable for many experiences, and when compared with unconstrained locomotion they are associated with increased cognitive load and a reduced sense of presence and spatial awareness. The positive

properties of these techniques makes them useful for designers and convenient for users, but facilitating true locomotion remains the topic of interest.

Redirection spaces and change blindness illusions are effective but insufficiently general, requiring purpose built environments through which users must follow pre-defined waypoints. Designing these environments is a highly technical task which requires an understanding of user spatial mapping. When combined with the task of designing an interesting and task-appropriate virtual environment, this is impractical. Methods such as [48] aim to simplify the task of designing virtual environments to make use of change blindness, but are limited in the type of environment they create, relying on a structure that is somewhat unrealistic and may be inappropriate for many environments. As with redirected walking, the ultimate goal is a general form which allows free walking with minimal adjustments to the virtual environment. This is a particular challenge for redirection spaces as they rely on manipulating the environment to control user behaviour. Nonetheless, this family of techniques is promising and requires further work.

Redirected walking has been used effectively with waypointed paths. Interest in a general form is significant, and work is ongoing. Techniques based on Optimization and Machine Learning are the current state of the art in VR locomotion, but performance improvements are not yet sufficient for use. The great challenge for these techniques is path prediction; current methods are basic, but more accurate path predictors would permit intelligent weighing and culling of paths, reducing complexity. Path predictors which combine short, medium and long term predictions would be most effective. Predictors which match user movement curves to those from previous data such as [79] are a promising development, as are environment-aware path predictors such as [76] which integrate knowledge of points of interest, though currently these points must be manually identified. The imprecision inherent in user movement makes path prediction a good candidate for machine learning techniques. This would require a large corpus of locomotion tracking data, which is feasible. Ideally, this tracking data would be specific to the environment in question, though this is significantly more difficult to attain. Another

option for dealing with complexity in redirected walking would be offline calculation of path predictions and transformations. This may allow us to benefit from the strengths of the optimization approach without having to reduce complexity. The closest approach to this currently are mapping-based techniques (see 2.4.3).

Employing redirection and change blindness illusions during saccades and blinks is a promising avenue of research, though currently in its very early stages. This technique could have wide-reaching implications for our current understanding of redirection perceptual thresholds, reducing the track space requirements for straight line walking and allowing more effective redirection for all techniques. Eye tracking and specifically saccade and blink detection is a problem that has seen significant work and has effective solutions [90]. These techniques are hardware intensive and may be hard to justify for consumer-level VR, though the advent of techniques such as foveated rendering [91] will, if adopted, make eye-tracking hardware a necessity for VR in coming years.

Chapter 3

MCRDW: Monte-Carlo Redirected Walking Gain Selection Algorithm

This chapter details the design of an algorithm which selects redirections to minimise boundary collisions. The primary contribution is the Monte-Carlo Redirected Walking (MCRDW) approach. This applies the Monte-Carlo method to redirected walking by conducting a large number of simulated virtual walks and using the results to select an appropriate gain. The technique can be used with (or integrated into) other gain selection algorithms. We provide a simple example implementation and a simulation based study for validation. When compared with the next best technique, MCRDW significantly reduced total collisions without increasing total rotation or position gain levels.

3.1 Introduction

Redirected walking (RDW) is capable of steering a user away from physical obstacles and boundaries without impeding virtual navigation and exploration. This process involves manipulating the data gathered by tracking physical user movement. The manipulated data is then applied to the user's virtual body. This disrupts the otherwise 1:1 mapping between virtual and physical movement. The result is dynamic remapping of the virtual space into the physical. In those cases where we choose to use RDW, the virtual space is typically much larger, and there are limits on the level of redirection (gain) that can be applied without disrupting the user. As

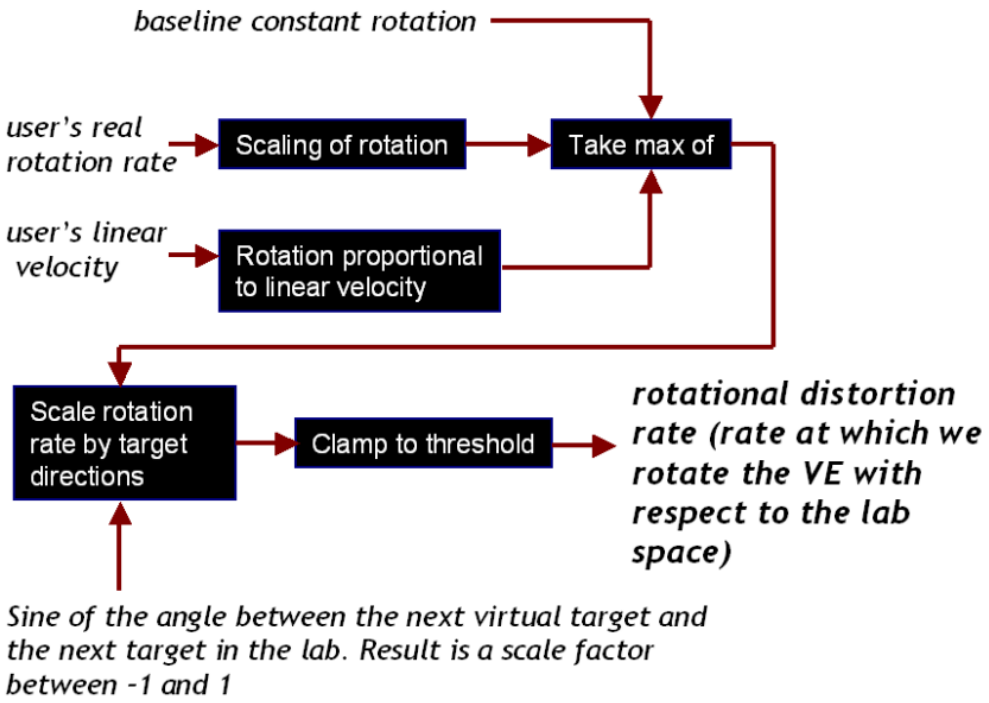


Figure 3.1: Method for calculating redirection direction and magnitude as proposed by Razzaque. Original source is [4]. Rotational distortion rate indicates the rotation due only to redirection, i.e., not including standard user rotation

a result, it is likely the user will occasionally have to stop before encountering a boundary - we call this a 'collision'. Collisions can be minimised by selecting gains levels appropriately.

In the original form of redirected walking as proposed by Razzaque [4] redirections are applied constantly; the magnitude and direction of redirection are decided by a gain selection algorithm (see Figure 3.1). In the accompanying study, users were asked to walk to a series of waypoints as they progress down a corridor. The waypoints are placed in a zig-zag pattern and correspond to a set of waypoints in the track space. The waypoints are key to the functionality of the algorithm; gain is determined by the relative angles between virtual and physical waypoints. With waypoints, a designer or experimenter can be sure of the path their users will take, and therefore design waypoints such that no boundary collision occurs.

Free exploration of a virtual environment is more challenging. In the case of a large open virtual environment, Steer-to-Orbit [65, 55] allows the theoretical limit

on performance to be reached, limited only by the maximum level of gain that can be applied without disrupting users. Redirection performance in obstacle-rich virtual environments is more interesting; the optimal gain varies based on user position in the virtual environment, and the layout of the virtual environment itself. A great many techniques exist:

- Heuristic techniques consider only the user's current position and typically attempt to steer towards a target (e.g., Steer-to-Center, Steer-to-Orbit, Steer-to-Multiple [65, 55, 63])
- Optimization based techniques consider possible future paths and aim to maximize some metric over a window (e.g., FORCE [71], MPCRed [72])
- Deep-learning based techniques use a pre-trained network to estimate good future redirections (e.g., Steer-to-Optimal-Target [5], Steering via Reinforcement Learning [6]).

While developments in optimisation and deep-learning based techniques are encouraging, the authors of these techniques report only a small improvement (at most around 15%) over Steer-to-Center (S2C) [71, 72, 5, 6].

Straightforward heuristic techniques remain competitive. Steer-to-Center with Center-based Translation Gain (S2+CTG) allows S2C to be combined with translation gain. Should the user be walking away from the center point of their physical space, their virtual movements are magnified. Should they be moving towards it, their virtual movements are reduced. The technique is very simple, and does not come with the associated computation overheads of optimisation or deep learning techniques, nor their more challenging implementations. However, in the right conditions it can improve on S2C by around 20% [68].

Comparing the gain selection performance of varying techniques is challenging. The performance of a given approach varies greatly based on the virtual environment in which it is being applied. Even with this in mind, this simple heuristic approach technique remains competitive. This demonstrates how challenging the gain selection problem is.

This chapter contains the design and evaluation of a novel approach to gain selection. The aim of this approach is to improve on existing redirection selection algorithms while remaining widely applicable, simple to implement and computationally lightweight.

The structure of the chapter is as follows. Section 3.2 covers the theory behind gain selection through simulated walks. Section 3.3 is a description of our sample implementation. Section 3.4 covers the design and results of a simulation-based study. Section 3.6 summarizes the chapter and considers future work.

3.2 Method

In this section we describe gain selection through simulated walks at a general level. See Section 3.4 for a sample implementation of these ideas and a simulated evaluation of that implementation.

To start at the very beginning we consider the redirection function f. As the user walks in the physical space, we apply f to their movements (translation and orientation) to calculate the resulting virtual world position. Without redirection, f is the identity function, and the user's movements therefore mapped 1 to 1. With redirection (or 'gain'), the virtual rotation and translation may be smaller or greater. Note that f only applies a redirection strategy, rather than generating one:

$$v_t = f(x, v_{t-1}, w_t, w_{t-1}) \tag{3.1}$$

Where:

$x =$ redirection 2-tuple containing rotation and translation gain

$v_t =$ user position and orientation in the virtual environment at time t

$w_t =$ user position and orientation in the track (physical) space at time t

The aim of redirection selection algorithms is to generate the redirection strategy x. Any value of x is possible, but the ideal value is likely to change frequently as the user moves around the environment, or as the environment changes. At an abstract level the optimal redirection selection algorithm has two goals when selecting

x:

- To maximize **boundary avoidance**; to select redirections that avoid physical boundaries
- To maximize **subtlety**; to select redirections that minimise disruption to the user

The simulated-walk approach to redirected walking is to virtually conduct many possible walks from the user's position under different redirection strategies. The walks are then scored in favour of those that best satisfy the *boundary avoidance* and *subtlety* goals. Various metrics can be applied during scoring; for example, walks could be weighted in favour of those that apply the lowest gain levels, or maximise the time to physical collision. The strategy with the highest score at a given moment is provided to the redirection function in the form of x.

The proposed approach is iterative and can combine the results of previous runs. As each simulation is computationally simple and standalone, this algorithm suits real-time applications as it can be run as long as allowed by the frame timing. A high level description of the algorithm can be found in Algorithm 1, and a 2-walk example diagram in Figure 3.2. The remainder of this section will consider approaches to simulation (see Sections 3.2.1 and 3.2.2) and scoring (Sections 3.2.3 and 3.2.4).

3.2.1 Simulating Virtual Walks

The first step of any simulation is to calculate $\overline{v}$, a path through the virtual environment. The path must start at v_0, the user's current position and orientation in the virtual space, and can end at any point. Ideally we would sample from the space of possible paths based on likelihood.

The simulated-walk approach makes the assumption that a sample of possible walks under different redirection strategies is representative of the set of all possible walks under all possible redirection strategies. To improve the quality of our sampling and bring us closer to a representative sample, walks can be selected for simulations based on a probability distribution which favours more likely paths. The

input : $\overline{y}$, a vector of redirection strategies of length n
$\overline{s}$, a vector of scores of length n
e, virtual and physical environments and user transforms
output: x, the recommended redirection
$\overline{s}$, the updated scores for each redirection

repeat
  $\overline{v} \leftarrow$ `generate-path(`e`)`
  for i **in** $\{0, 1, \ldots, n\}$
    $\overline{w} \leftarrow$ `simulate-walk(`$\overline{v}$`,` y_i`,` e`)`
    $s_i \leftarrow$ `update-score(`$\overline{w}$`,` s_i`)`
  end
until `computation-time-exceeded()`
$x \leftarrow$ `get-max-scoring(`$\overline{y}$`,` $\overline{s}$`)`

Algorithm 1: Monte-Carlo Redirected Walking (MCRDW) overview. The output, x, can be used as input to f, the redirection function (see Equation 3.1). See Figure 3.2 for an example of the approach in action.

task of generating good virtual paths is therefore analogous to the path prediction task found elsewhere in redirected walking.

In theory, any long-term path predictor can be used, and a variety exist in the literature (see Section 2.5.4 from the accompanying survey on path predictors). However, many path predictors only provide a single estimate of the user's path. More useful for our purposes are those that provide a range of possible paths with accompanying probabilities; we will call these stochastic path predictors. Accurate probabilities are to be preferred as less computation time will be wasted on unlikely paths. Additionally, when scoring walks, it becomes possible to weight redirections in favour of those that perform best on more likely paths. The following are examples of elements of the virtual environment that might be considered to hint towards path prediction probabilities:

- Historical user virtual path [33], optionally considering models of human locomotion [79]
- Layout of the virtual environment, in the form of a graph [76] or a navigation mesh [86]
- User task or points of interest in the virtual environment; previous work has

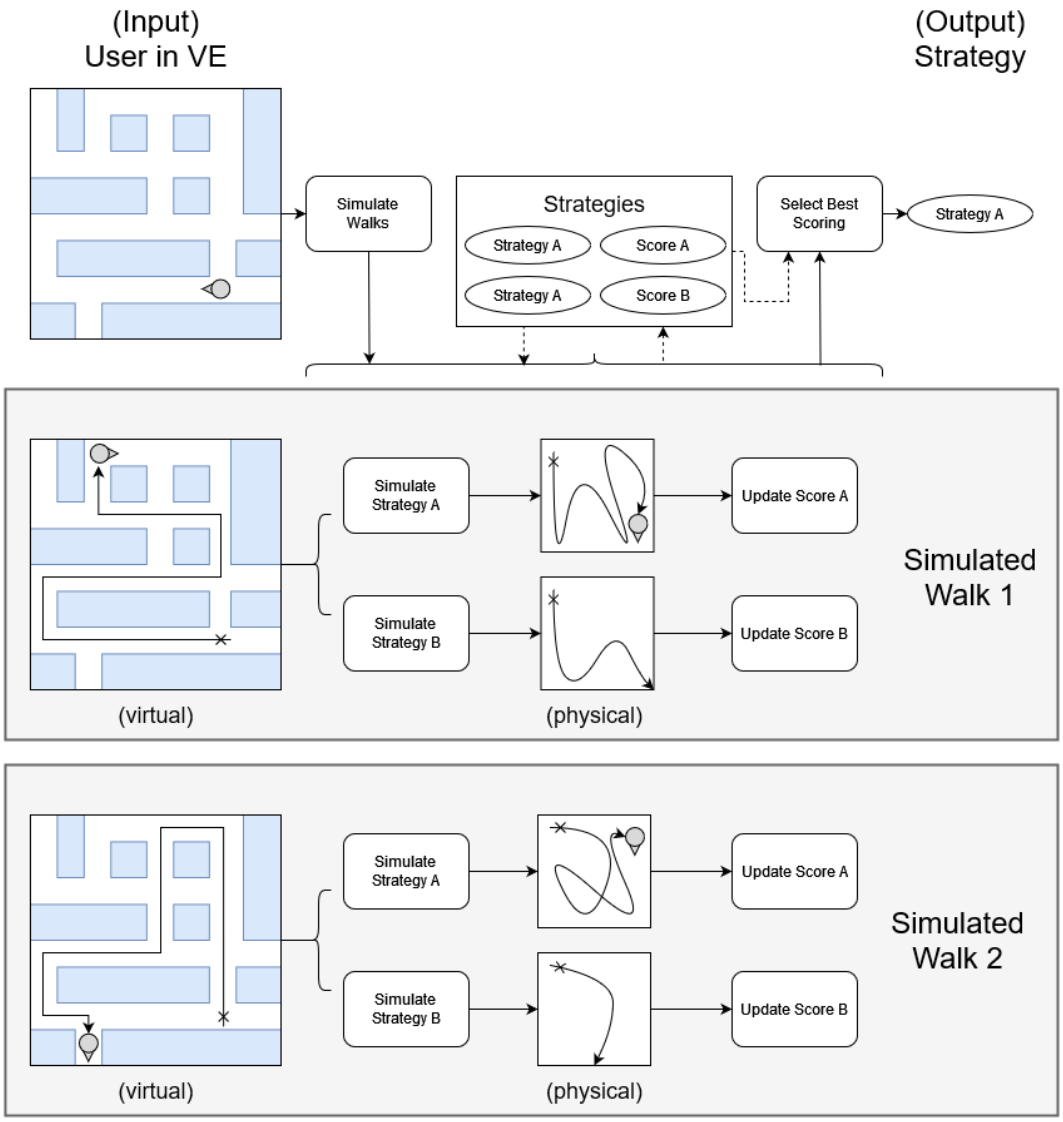


Figure 3.2: Monte-Carlo Redirected Walking (MCRDW) example system. For each simulated walk, a virtual path is generated (to the left of the diagram, labelled virtual). Each strategy is then applied to the virtual path, generating a corresponding simulated physical path (the middle of the diagram, labelled physical). This physical path is then scored and the score for the strategy updated. Finally, after all simulations, the current best scoring strategy is returned. In this example, Strategy B leads to simulated physical collisions in both simulated walks, so Strategy A is preferred. For the sake of clarity in this diagram, only two strategies are compared and two simulations are conducted. In practice, more strategies (in the order of 10s) and many more simulations (in the order of 1000s) will likely be required for good results.

shown paths are predictable when movement is goal-orientated [78]

Finally, as paths must be generated at runtime, a very quick path predictor is also preferred. This provides as much time as possible for simulations.

As the MCRDW approach is compatible with the majority of path predictors we provide no recommendations in this section, other than to note that the ideal path predictor is computationally lightweight and stochastic. The concrete implementation provided in Section 3.4 uses an approach loosely based on that used by Peck [33], extended to provide probabilities.

3.2.2 Calculating Physical Walks

Once the virtual path $\overline{v}$ has been generated, the next task is to work back to the physical path that would have led the user round this virtual path. The physical path is dependent on the redirection strategy in use. Related to f (see Equation 3.1), we now need a new function which calculates where a user would be in the track space if they were to walk a virtual path with a certain set of redirections applied. As the user subconsciously counters the redirections in the virtual environment, we should then be able to calculate where those movements will place them in the physical space. We will call this function g:

$$w_t = g(x, w_{t-1}, v_t, v_{t-1}) \tag{3.2}$$

For our simulations complete virtual paths are generated. All values of v (and consequently w) are therefore known. Different approaches to f are possible; for the sake of calculating physical paths, easily invertible approaches are preferred. Example equations for f and g are provided in Section 3.4.

3.2.3 Subtlety Metrics

One way to satisfy the goal of *subtlety* is to strictly limit possible strategies to those that have been found to be hard to detect (for example, in prior work [52, 40]). Theoretically no metric would then be required; scoring could focus on *boundary avoidance* alone.

However, re-orientation after a boundary collision is overt and constitutes a significant disruption to user experience. It may be that overt redirections are preferable to boundary collisions. The simulated walk approach can allow for a balance to be found between *boundary avoidance* and *subtlety*, as larger redirections are less

subtle but more effective. Redirection strategies can therefore include large redirections with a score penalty, permitting somewhat perceptible redirection if it would prevent a user from encountering a boundary.

If we are to adopt the balanced approach we are required to weigh strategies against one another based on the rate of gain applied.

The mean rate of unsigned redirection (MRUR) during navigation [56] is an existing metric for evaluation purposes which we may be able to apply in an objective function. One issue is that taking the mean redirection rate obscures periods of intense redirection and abrupt changes in redirection magnitude. For example, a technique which oscillated between large redirections and no redirections will score equivalently with a technique which maintains consistently small redirections.

Work exists showing no connection with rate of change and subtlety for redirections [54]. However, most implementations of redirected walking use a smoothing term explicitly to avoid sudden changes in redirection magnitude [55, 56]. Our own work found that rate of redirection has a significant effect on user detection thresholds [1]. More work is required for an authoritative statement, but currently it would be valuable to measure rate of redirection more thoroughly.

The simplest form this could take would be to evaluate the maximum rate of redirection during navigation (MaxRUR). This provides no information about changes in redirection magnitude, which ideally we would also want to consider. A good start would therefore be measuring both MRUR and MaxRUR.

Ultimately the ideal approach depends on use-case. The example implementation described in Section 3.4 uses the first approach described in this section; staying below the prior thresholds found in [52] and [40]. Primarily this is to help us easily compare the MCRDW approach with earlier techniques. However, a very small score penalty is still applied based on the rate of gain (MaxRUR) for the initial strategy. This helps to encourage the algorithm to apply no redirection unless required. Redirection is believed to have very little impact on cognitive load and spatial understanding when kept below the thresholds we use here [63, 89]. However, changing gains frequently does have an impact [1]. Keeping overall redirection applied to a

minimum is likely desirable.

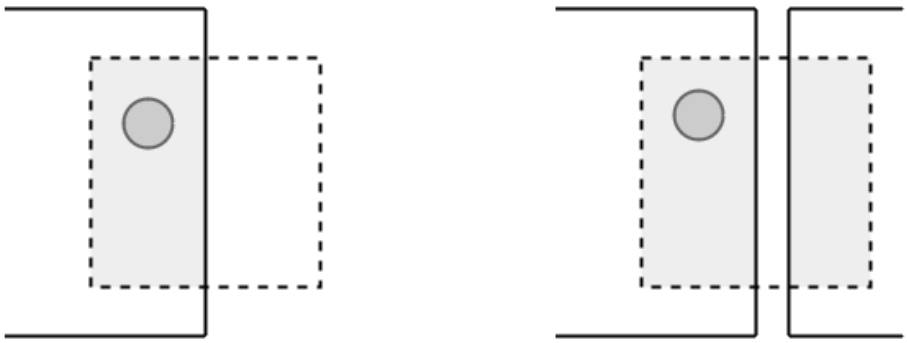
3.2.4 Boundary-Avoidance Metrics

Here we consider how to make best use of a relatively limited simulation window with metrics which estimate future performance based on current state. A good function will provide a precise estimate of the user's future likelihood of a boundary collision. It is also of benefit if they are simply expressed and computationally efficient.

The most straightforward metric is simply counting the time to first boundary collision (TTBC) during the walk. A possible hazard with this technique is its inability to capture near misses. Approaching the boundary and only just avoiding collisions will be scored highly by TTBC. However, as simulations will never be completely precise, in practice the strategy is best avoided if possible as it is likely to lead to collision.

Hodgson et al. suggest a number of metrics for comparing the performance of redirected walking techniques [56] that we may be able to adapt into *boundary avoidance* metrics. The first is measuring the mean distance from the track space center (MDC) over the course of the entire path. This metric is one member of a family of metrics where the resulting score is based entirely on the user's track space position. We consider all these functions together under the name MDC, as the others are in effect extensions to deal with non-square track spaces.

MDC has a significant advantage over TTBC in that it is more granular, making it easier to compare two series of redirections over a short time scale and get sensible results. However, MDC is based on the concept that greater distance from the track space center is an indication of poor redirection performance, which is only sometimes the case. Consider the heuristic technique *Steer-To-Orbit*, which leads the user around the optimal path for straight-line walking [65, 55]. The path used is circular with a high radius. For the same path, Steer-To-Center would lead the user in a figure-of-eight pattern with a much higher maximum radius, increasing the likelihood of boundary collision. However, both techniques would score similarly under MDC.



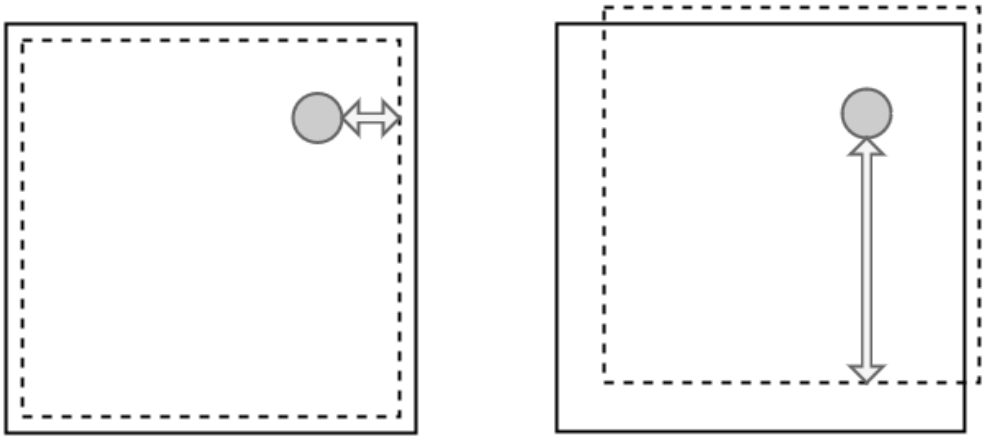
(a) MAI 1 **(b)** MAI 2

Figure 3.3: The MAI (mean area intersection) metric. Dotted lines represent track space boundaries; the area in the virtual environment the user would be able to reach by physical walking with no gain applied. These are invisible to the user. The solid lines are the boundaries in the virtual environment, visible to the user. The circle is the user's position. The light grey rectangle represents the scoring area. Figure a demonstrates the MAI approach in action, while b shows an issue with this approach; the area across the boundary is considered walkable and will increase the score, but does not reduce the likelihood of boundary collision. This is because the area across the virtual boundary (to the right of the image) is not reachable from the user's (virtual) perspective, so they will not walk in that direction.

Hodgson's second metric is measuring the maximum physical distance from the track space center (MaxDC). This solves some of the problems introduced above; now Steer-To-Orbit will score better than Steer-To-Center in straight-line walking. However, MaxDC has the same problem as MDC for our purposes; distance from track space center is not a problem if it does not lead to boundary collisons. Functionally, a very large maximum is equivalent to a small maximum if either of these results is larger than the track space radius.

Allowing functions to take the virtual environment around the user into account significantly increases complexity but may provide a more accurate indication of how likely the user is to leave the track space in the future. This is particularly appropriate to environments where the user's movement is heavily constrained by environment layout or obstacles.

Geometric methods are a simple way of combining track space and virtual environment information. For example, we could use the mean area of the intersec-



(a) DNFB 1 **(b)** DNFB 2

Figure 3.4: The DNFB (distance to nearest false boundary) metric. Dotted lines represent track space boundaries; the area in the virtual environment the user would be able to reach by physical walking with no gain applied. These are invisible to the user. The solid lines are the boundaries in the virtual environment, visible to the user. The circle is the user's position. The arrow points to the nearest FB (false boundary). In a, the boundary is very close and the DNFB score therefore low. With only a small movement of track space as in b, the DNFB score significantly increases. This is a useful property as the user cannot now easily collide with a boundary. However, it does indicate that DNFB values are subject to sudden discontinuities.

tion (MAI) of track and virtual space as an objective function (see Figure 3.3a for an example). MAI is valuable for indicating when track space is being wasted on areas that cannot be traversed. However, the intersection operation does not indicate how likely a walkable area is to be traversed; possibly the area could be behind an obstacle and so unlikely to be a prudent use of track space (see Figure 3.3b).

The greatest problem with MAI is that it cannot identify which parts of the intersected space are useful. For example, consider two cases where the user approaches a wall in a large virtual room with no obstacles; in the first, the track space overlaps the wall slightly, while in the second, the track space falls well short of the wall. The first will score worse under MAI, but only the second will lead to a boundary collision.

Examining these functions in details helps clarify the scenario that leads to boundary collisions. Boundary collisions arise when a track space boundary occurs before a virtual environment boundary in the user's path. Should the user walk in

that direction, they will encounter a track space boundary before they encounter a boundary in the virtual environment, and therefore be stopped out of necessity rather than naturally by the environment. We call this phenomenon a ”false boundary” (FB). FBs can be used as the foundation for a metric by taking the mean distance to the nearest FB (MDNFB). See Figures 3.4a and 3.4b for the technique in practice.

FBs are the underlying cause of boundary collisions, so the metric does not suffer from the same problems as MAI. In our previous hypothetical, the track space overlapping the wall will score significantly better than the one falling short. However, while MDNFB is more sophisticated than distance based metrics such as MDC, it has a similar issue; large values of MDNFB are not necessarily preferable to small values, so long as the value does not reach zero, indicating a collision. MDNFB is also time consuming to calculate; the process requires casting rays against a navmesh representing walkable space in the virtual environment. Accurate values can require many rays to be cast.

The example implementation described in Section 3.4 uses the simplest possible approach, TTBC. As the implementation limits simulations to a fixed length in time, results from TTBC are easily normalised. This simplifies the process of combining *boundary avoidance* and *subtlety* metrics into a single measure. As TTBC is also very simple to calculate, more time is available for simulations.

3.3 Implementation

Our concrete implementation of MCRDW follows the general approach described in Section 3.2: At each timestep, run (generate, then walkthrough) as many walks as possible until the computation time available elapses, then store the results and return the best scoring redirection across all simulations.

3.3.1 Generating Walks

For each simulation we generate a walk using a path predictor. The user is considered to be at the closest current node. The predictor weighs the probability of visiting any neighbouring nodes, then a node is picked randomly from the (weighted) possibilities, and finally the predictor is updated with the new information. This

process repeats until the max path length is exceeded.

The predictor used has two components. Each component outputs a list of probabilities, one for each neighbouring node. The first component, 'history', records visited nodes and decreases the likelihood of visiting those which have been visited very recently. The second component, 'direction', increases the likelihood of visiting nodes which the user is heading towards. Direction is calculated by sampling the user's movements to generate a smoothed direction vector.

To weigh the two components against each other, the consistency of the direction vector and the speed of the user are combined to create a value, 'confidence'. When confidence is low we rely on history. When confidence is high, we rely on direction.

3.3.2 Performing Walkthroughs

When we have the walk generated, we perform a walkthrough. First we divide the walk into sections ('legs'), each treated as a straight line. We use the waypoints of the path as the start and end points of legs. For each leg, we select a redirection strategy.

Early in the walkthrough, all legs use the same strategy: this is the strategy that will have its score updated. Strategies for later legs are selected randomly with no weighting. This is because the algorithm is free to combine strategies. The best approach is almost certain to be one strategy now and different strategies later. Randomly selecting strategies for later legs helps to represent these combinations.

The process of simulating a leg is straightforward. Over the course of the leg, the user must at least turn to face the end point of the leg and must at least move the distance between the start and end points. We call these the virtual turn delta and virtual position delta. The user may turn more or move further should they move on a curved path or back and forth. However, we use the most straightforward path for our simulations as it can be considered the worst case; it gives the algorithm the least opportunity to apply gain.

Typically in RDW physical movement is provided by the physical movement of the user, and therefore can be considered fixed. We apply redirection to this

physical movement to generate redirected virtual movement. Instead, in these walkthroughs, the virtual movement is fixed. We are therefore interested in calculating what physical movement would have generated this virtual movement given a particular redirection strategy. With a sufficiently simple redirection formula, we can work back from virtual turn and position deltas to calculate physical turn and position deltas. We use the following physical-to-virtual equations for delta rotations and positions:

$$\Delta v_{rot} = \Delta w_{rot} \cdot \begin{cases} coro & \text{if } \operatorname{sgn} x_{rot} = \operatorname{sgn} \Delta w_{rot} \\ anti & \text{otherwise} \end{cases} \tag{3.3}$$

$$\left\|\Delta v_{pos}\right\| = \left\|\Delta w_{pos}\right\| \cdot \begin{cases} magn & \text{if } x_{pos} > 0 \\ redu & \text{otherwise} \end{cases} \tag{3.4}$$

Where:

$x =$ 2-tuple containing (rot and pos) redirection instruction

$v =$ 2-tuple containing user's virt rotation and position

$w =$ 2-tuple containing user's phys rotation and position

$coro =$ gain when turning with the redirection instruction, $\in [1, \infty)$

$anti =$ gain when turning opposite to the redirection instruction, $\in (0, 1]$

$magn =$ gain when instruction is to magnify physical movement, $\in [1, \infty)$

$redu =$ gain when instruction is to reduce physical movement, $\in (0, 1]$

We use *coro*, *anti*, *magn* and *redu* because earlier work has found users have varying tolerances depending on the direction of redirection [40, 52]. Note that for positions, only the magnitude is modified as position deltas are applied relative to the current facing vector in the appropriate space (virtual or physical). Additionally, in practice, redirection equations will also apply smoothing. However, we omit this when performing walkthroughs to significantly simplify the process of simulating legs. Finally, note the method is never required to apply these forward transforma-

tions; we only use them to generate the following corresponding virtual-to-physical equations:

$$\Delta w_{rot} = \Delta v_{rot} \cdot \begin{cases} \frac{1}{1+r\cdot(coro-1)} & \text{if sgn } x_{rot} = \text{sgn } \Delta v_{rot} \\ \frac{1}{1+r\cdot(anti-1)} & \text{otherwise} \end{cases} \tag{3.5}$$

$$\left\|\Delta w_{pos}\right\| = \left\|\Delta v_{pos}\right\| \cdot \begin{cases} \frac{1}{1+r\cdot(magn-1)} & \text{if } x_{pos} > 0 \\ \frac{1}{1+r\cdot(redu-1)} & \text{otherwise} \end{cases} \tag{3.6}$$

Where r is a random number $\in [0,1]$, generated per leg. This random term is included to help simulate the possibility that, at runtime, the algorithm may change redirection strategy at any point, including partway through a leg. During early development of the technique and the simulations performed in Section 3.4, the addition of this term led to a small but consistent performance improvement.

The walkthrough continues until encountering the end of the path or a boundary in the (simulated) physical environment. Finally, the score is recorded. This is the total amount of time walked before encountering a boundary.

3.3.3 Finalising Scores

Finally, scores are processed and merged with the scores from previous frames using the following process: First all scores are averaged by the number of walks. A weighting is then applied per-strategy. Those strategies which apply greater levels of redirection are slightly disincentivized based on MaxRUR (see Section 3.2.3). For merging, the resulting scores for each redirection are pushed on to a queue along with a scalar value for the current time. The usable scores at any given time are an average of all the scores in the queue, linearly weighted in favour of the most recent. The highest scoring method using this approach is the current redirection strategy.

3.4 Evaluation

This section contains information on the simulation-based experiment conducted to validate the technique. This includes a concrete implementation of MCRDW and

details on simulation methodology and experimental setup. We conclude with the results of the experiment and a discussion of these results.

3.4.1 Simulator

The simulator used in this section is purely for the purpose of evaluating the MCRDW approach, and does not form part of the algorithm itself. It has three core components:

- A **layout**; a virtual environment and path around that environment
- A **method**; the redirection techniques under evaluation
- A **runner** for conducting simulations

The inputs to the simulation are *methods* and *layouts*. The *runner* applies the *method* to the movements of a simulated user as they follow the path described in a *layout*. During the run, the runner also gathers and logs performance statistics.

We use a large number of procedurally generated layouts to represent the population of possible virtual environments. For the sake of comparison we run through each layout once with each method. A *scheduler* can be used to queue up runs with the correct layout and method. A *visualizer* can optionally provide a graphical output to help monitor the simulations while in-progress.

3.4.2 Layouts

Layouts contain two elements: the virtual environment itself and a path through the environment. The virtual environment is represented by a set of walls which cannot be traversed, and the path by a series of a waypoints. Waypoints can be placed close to one another to represent curved paths. For our purposes, long straight paths are sufficient.

We use a connected graph as an intermediate step to help generate the layout. Note however that the layout itself is not graph based. The inputs to our generation follow:

- The **size** ($n \times m$); the number of nodes in each dimension

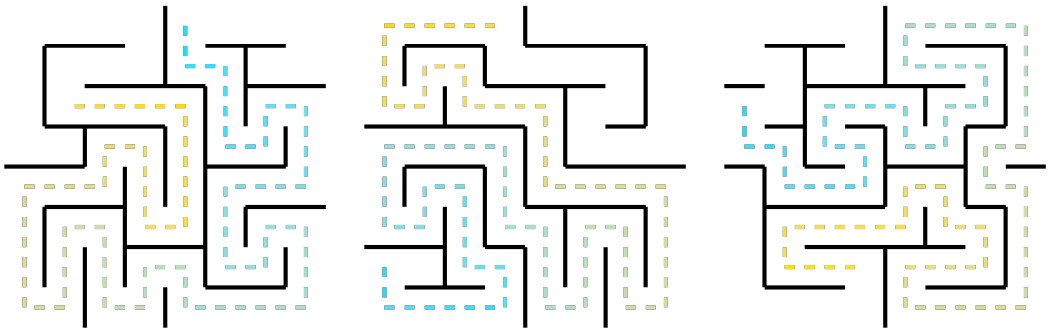


Figure 3.5: Sample Layouts - Edge Factor 0

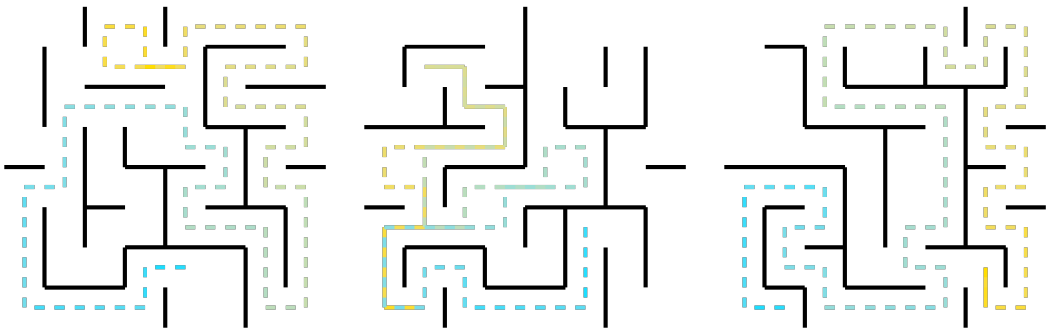


Figure 3.6: Sample Layouts - Edge Factor 0.15

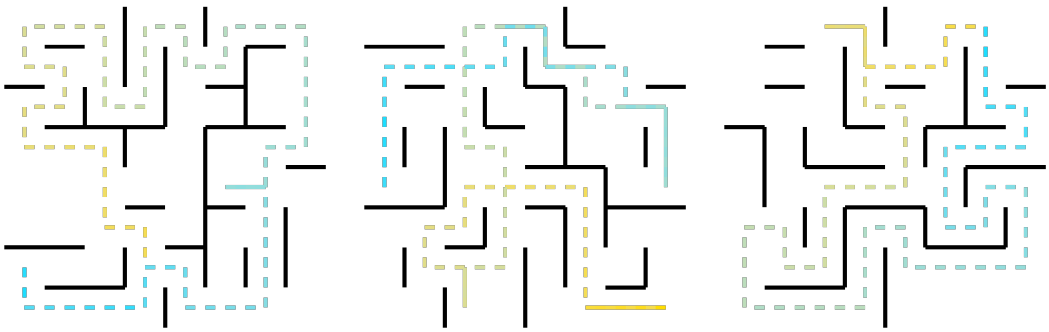
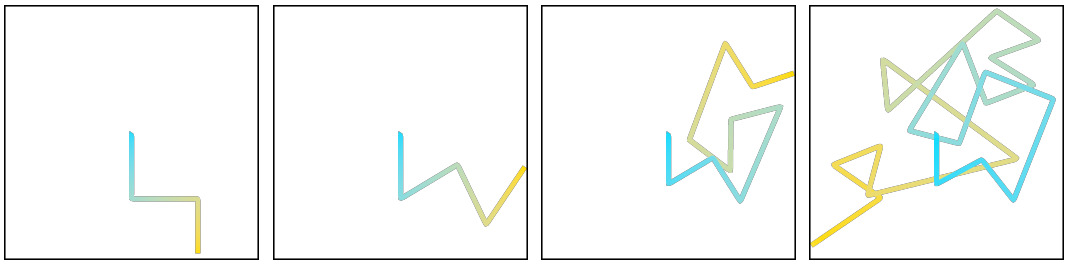


Figure 3.7: Sample Layouts - Edge Factor 0.3

- The **node spacing**; the distance between each node (evenly spaced)
- The **path length**; desired path length
- The **edge factor** $\in [0, 1]$; overall connectedness of the layout
- A **max straight length**; constraint on length of straight sections
- A **max straight path length**; constraint on length of straight sections of path

To generate the environment, we use the following process:



(a) No-RDW **(b)** S2T **(c)** S2T-S **(d)** MCRDW

Figure 3.8: The simulated physical walks for one virtual layout, under different redirection methods. These are the paths for the rightmost layout in Figure 3.5. The blue and yellow ends of the line are the start and end of the path respectively. For clarity in these diagrams, we only include the walk up to the first boundary collision. In our evaluation, we instead log the boundary collision, reposition the user in the center of their physical space and continue the simulation.

1. Start with an $n \times m$ grid of evenly spaced nodes
2. Randomly generate a minimally connected graph of these nodes
3. Randomly add more edges as required by the edge factor
4. Wherever two nodes are not connected by an edge, generate a wall

And for the path:

1. Randomly select a start and destination node
2. Generate the shortest possible path between start and destination
3. If total path length would exceed desired path length (see above), cut it short and return
4. Otherwise, pick another destination node and continue

After generation the path and layout are checked for compliance with the max straight length and max straight path length constraints. Should any constraint fail the check, the path and layout are discarded. These constraints are included to make sure that boundary collisions are theoretically avoidable by good gain selection; see Section 3.4.5 for more on this topic.

The edge factor has the most significant effect on the overall layout. With an edge factor of 0, the layout is minimally connected. With an edge factor of

1, every node is connected to every neighboring node. Edge factor 0 represents dense, obstacle-rich environments, while edge factor 1 represents wide open spaces. We can vary the edge factor to smoothly generate any point in between these two extremes. For sample layouts at different levels of edge factor, see Figures 3.5—3.7.

The virtual path generated as part of the layout is always followed exactly by the simulator in this evaluation, and is repeated once for each method. As the method varies, it is the simulated physical path which varies, and which is used for evaluation purposes.

3.4.3 Runner

The runner is responsible for conducting the simulation using the environment and the path defined by a layout, and the redirection technique defined by a method.

The runner loads the environment described in the layout and advances the simulation at a fixed timestep, first updating the simulated user's position and orientation along the path found in the layout. The runner then passes control to the method. The runner repeats this process until the simulated user reaches the end of the layout's path. Should a user encounter a boundary in their simulated track space, a simulated 'reset'[29] occurs: the user is returned to the center of their track space but their position in the virtual environment remains the same.

The runner records the total number of collisions and the total amount of rotation and translation gain applied during the run for later analysis.

A very simple model of locomotion is used when following the path; the simulated user turns to face their current waypoint and moves towards it. When arriving at the waypoint, the process is repeated with the next waypoint.

The output of the runner is semi-deterministic; on the same hardware, the runner will produce the same outcome. With different hardware, differences may be observed due to differing floating point handling etc.

3.4.4 Methods

At each time step, methods are provided a delta time and the user's virtual and track space position and orientation as input. Methods are then free to manipulate the

virtual position and orientation. To do this, all methods use a common 'redirector' to apply gain. The redirector applies simple smoothing as is typical in RDW applications [55, 56, 1]. Methods are also notified of discontinuities (e.g., resets). This gives the method an opportunity to reset smoothing and prediction variables.

In our simulations, the following methods were evaluated:

S2C *Steer-to-Center,* which guides the user towards the center of their physical space [65, 55]. Rotation gain only.

S2O *Steer-to-Orbit,* which guides the user on a circular path around the edge of their physical space [65, 55]. Rotation gain only.

S2T *Steer-to-temporary,* as S2C but with temporary targets when facing directly away from the center to ensure consistent gain direction [56]. Rotation gain only.

S2T-S *S2T with static magnification,* as S2T but applying a constant translation gain, effectively increasing the size of the user's physical space. Rotation and translation gain.

S2T-D *S2T with dynamic magnification,* as S2T but applying a dynamic translation gain; magnifying user movement when moving away from the center, and reducing it when moving towards the center [68]. Rotation and translation gain.

NoRDW *No redirected walking,* control condition, map user movements 1:1. No rotation or translation gain.

MC *Monte-Carlo redirected walking* (**MCRDW**, abbreviated for space), as described in Section 3.3. Rotation and translation gain.

MC-Fast *Monte-Carlo redirected walking with half compute time,* as MCRDW but reduce available compute time by half, intended to help approximate computation requirements. Rotation and translation gain.

MC-Lo *Monte-Carlo redirected walking, over-threshold, low,* as MCRDW but when a boundary condition is likely the technique is permitted to exceed perceptual thresholds by 10%. Rotation and translation gain.

MC-Med *Monte-Carlo redirected walking, over-threshold, medium,* as MC-Lo but instead exceed thresholds by 20%. Rotation and translation gain.

MC-Hi *Monte-Carlo redirected walking, over-threshold, high,* as MC-Lo but instead exceed thresholds by 30%. Rotation and translation gain.

All MCRDW techniques require an element of chance for path prediction, sampling the wider population of possible walks and deciding on a strategy. These are provided by a pseudo-random number generator with a fixed seed.

3.4.5 Experimental Setup

Using the procedure described in Section 3.4.2, we generate 1000 layouts at 3 different levels of edge factors: .00, .15 and .30. This gives us 3000 layouts total. These levels were selected arbitrarily to represent different kinds of environment. At .30, the virtual environment is quite substantially open, with too many paths to reasonably evaluate exhaustively. The generator had the following configuration: 8x8 nodes, node spacing 1.3m, total path length 60m, max straight length and max straight path length 4m. As the purpose of these simulations is to distinguish between methods, we use a 5m by 5m physical space to increase the frequency of boundary collisions. With such a small physical space, no redirection technique can make long-distance straight line walking possible, so we constrain these two aspects to make each boundary collision meaningful. For selected layouts from the experiment generated with this configuration, see Figures 3.5 - 3.7.

We then simulated the walk generated for each layout with each of the 11 methods described in Section 3.4.4. The recorded outcome measures, summed across each simulation, were:

- Total boundary collisions
- Total absolute position difference (in metres)

Method	$.00 \times .15$	$.00 \times .30$	$.15 \times .30$	η^2
S2C				-
S2O				-
S2T				-
S2T-S	*	*		.009
S2T-D		*	*	.009
NoRDW	*	*		.007
MC	*	*	*	.269
MC-Fast	*	*	*	.285
MC-Lo	*	*	*	.292
MC-Med	*	*	*	.263
MC-Hi	*	*	*	.282

Table 3.1: Pairwise comparisons for **simple main effects of edge factor** on **boundary collisions**, by method. Edge factor had no significant effect on S2C, S2O and S2T.

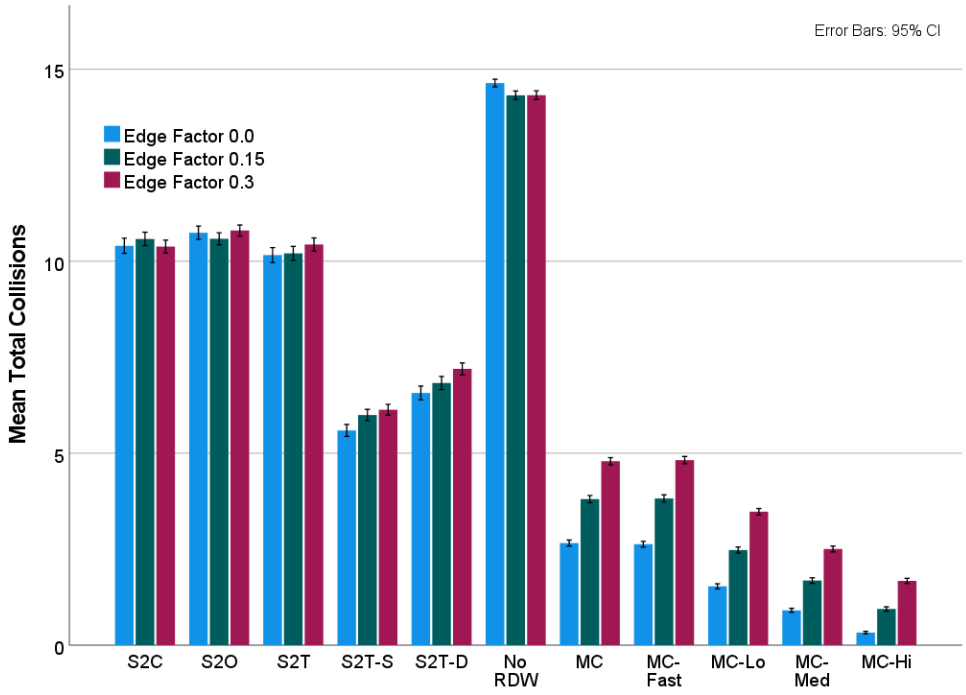


Figure 3.9: Mean total **collisions** by method across all simulations, grouped by edge factor.

- Total absolute rotation difference (in degrees)

For our simulations, the user walking speed is 1 meter per second. The user turns at a rate of 90 degrees per second. As redirection gain is applied multiplicatively, these speeds have minimal impact on the outcome of a simulation. The time step used was 60 updates per second.

For methods with a configurable run-time (MC, MC-Fast, MC-Lo, MC-Med, MC-Hi), the reference calculation time was 10 milliseconds across 6 threads on a Ryzen 7 5800H. However, to help gather results more quickly for this evaluation, simulations were run across machines. To standardise results across machines, a small section of the simulations was performed with the reference setup above and the total number of path sections recorded. The result was a mean of ≈ 22500 and standard deviation of ≈ 2000. This was our calibration value; the methods were therefore limited to calculating no more than 22500 path sections, except for MC-Fast which was instead limited to 11250.

3.4.6 Results

We consider the three independent variables separately. These variables were total collisions, total absolute position difference and total absolute rotation difference. Each was assessed with a 2-way mixed ANOVA. The within-subjects factor was the condition, as each layout had each condition applied. The between-subjects factor was the level of edge factor, as this generated three different sets of layouts.

Given that these results are based on simulations and the sample count is therefore high, we use a more restrictive threshold than is typical threshold of Unless otherwise stated, we look for significance at the 1% level ($p < 0.01$), rather than the typical 5%. This to reflect the impact high sample count has on p.

3.4.6.1 Total Boundary Collisions

The data was normally distributed as assessed by visual inspection of Normal Q-Q Plots. The data included a small number ($n \leq 4$) of outliers (± 4 standard deviations) among all methods. As the sample size is large, we include these values regardless. We omit no values from the dataset.

The data violated Levene's and Box's test for homogeneity of variance and covariance respectively. However, these tests are known to be sensitive with large sample sizes, and with groups of equal size mixed ANOVA is considered robust to heterogenity of variances and covariances [92]. Additionally, visual inspection of scatter plots of residuals showed the expected shape, so we conclude the assump-

Method	$.00 \times .15$	$.00 \times .30$	$.15 \times .30$	η^2
S2T-S	*	*	*	.026
S2T-D				-
MC	*	*	*	.494
MC-Fast	*	*	*	.504
MC-Lo	*	*	*	.471
MC-Med	*	*	*	.422
MC-Hi	*	*	*	.418

Table 3.2: Pairwise comparisons for **simple main effects of edge factor** on total absolute **position difference**, by method. S2C, S2O, S2T and NoRDW methods applied no position gain so are not considered. Edge factor had no significant effect on STS-D.

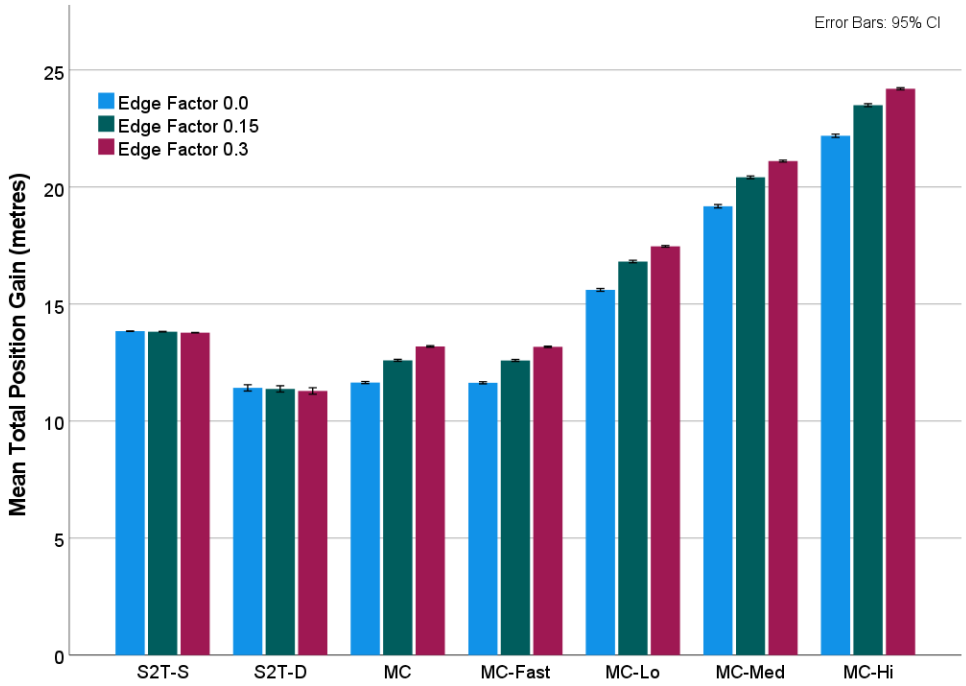


Figure 3.10: Mean total absolute **position difference** by method across all simulations, grouped by edge factor.

tions of the mixed ANOVA are not violated and perform no transformations on the data.

There was a statistically significant interaction between the method and level of edge factor on boundary collisions, $F(13.189, 19763.341) = 62.045$, $p < .0005$, partial $\eta^2 = .4$. A Greenhouse-Geisser correction was used ($\varepsilon = .659$) as Mauchly's test of sphericity indicated that the assumption of sphericity was violated for the two-way interaction, $\chi^2 = 13.34$, $p < .0005$.

Analysis of the **simple main effect of method** is used to determine whether method had an effect on the number of boundary collisions at each level of edge factor:

- **Edge-Factor .00**: $F(5.9, 5905.4) = 5934.8$, $p < .0001, \eta^2 = .856$
- **Edge-Factor .15**: $F(6.7, 6711.9) = 5560.7$, $p < .0001, \eta^2 = .848$
- **Edge-Factor .30**: $F(7.1, 7094.0) = 5166.6$, $p < .0001, \eta^2 = .838$

Pairwise comparisons follow. As is to be expected given the high sample count, almost every pairwise comparison between methods showed significant difference ($p < 0.0001$). So instead we here list the *exceptions*; those comparisons which were *not* significant:

- **MC** $\times$ **MC-Fast**: No significant comparison at any level of edge factor, $p > 0.9999$
- **S2C** $\times$ **S2O**: No significant comparison at edge factor level .00, or .15
- **S2C** $\times$ **S2T**: No significant comparison at edge factor level .00, or .30

Analysis of the **simple main effect of edge factor** is used to determine whether edge factor had an effect on the number of boundary collisions for each method:

- **S2C, S2O, S2T**: No significant effect
- **S2T-S**: $F(2, 2997) = 13.9$, $p < .001, \eta^2 = .009$
- **STS-D**: $F(2, 2997) = 13.3$, $p < .001, \eta^2 = .009$
- **NoRDW**: $F(2, 2997) = 10.8$, $p < .001, \eta^2 = .007$
- **MC**: $F(2, 2997) = 551.6$, $p < .0001, \eta^2 = .269$
- **MC-Fast**: $F(2, 2997) = 597.6$, $p < .0001, \eta^2 = .285$
- **MC-Lo**: $F(2, 2997) = 619.1$, $p < .0001, \eta^2 = .292$
- **MC-Med**: $F(2, 2997) = 638.6$, $p < .0001, \eta^2 = .263$

- **MC-Hi**: $F(2,2997) = 587.3$, $p < .0001, \eta^2 = .282$

Pairwise comparisons summarized in Table 3.1.

3.4.6.2 Total Absolute Position Difference

The methods S2C, S2O, S2T and NoRDW do not apply gain to positions so are excluded from this analysis. The remaining data was visually inspected for normality via Normal Q-Q Plot. S2T-D, MC, MC-Half, MC-Lo, MC-Med and MC-Hi appeared normally distributed. However, S2T-S only fit the model very approximately. With large sample sizes, ANOVA is considered robust to violations of normality so we include S2T-S [92].

The data included a small number ($n \leq 4$) of outliers (± 4 standard deviations) among all included methods. As the sample size is large, we include these values regardless. We omit no values from the dataset.

The data violated Levene's and Box's test for homogeneity of variance and covariance respectively. However, these tests are known to be sensitive with large sample sizes, and with groups of equal size mixed ANOVA is considered robust to heterogenity of variances and covariances [92]. Additionally, visual inspection of scatter plots of residuals showed the expected shape, so we conclude the assumptions of the mixed ANOVA are not violated and perform no transformations on the data.

There was a statistically significant interaction between the method and the level of edge factor on total absolute position difference, $F(3.7, 5542.7) = 222.7$, $p < .0005$, partial $\eta^2 = .129$. A Greenhouse-Geisser correction was used ($\varepsilon = .308$) as Mauchly's test of sphericity indicated that the assumption of sphericity was violated for the two-way interaction, $\chi^2 = 14201.9$, $p < .0005$.

Analysis of the **simple main effect of method** is used to determine whether method had an effect on the total absolute position difference applied at each level of edge factor:

- **Edge-Factor .00**: $F(2.2, 2243.8) = 15354.1$, $p < .0001, \eta^2 = .939$
- **Edge-Factor .15**: $F(1.9, 1865.1) = 21492.3$, $p < .0001, \eta^2 = .956$

- **Edge-Factor .30**: $F(1.4, 1413.9) = 26400.3$, $p < .0001, \eta^2 = .964$

Pairwise comparisons follow. As is to be expected given the high sample count, almost every pairwise comparison between methods showed significant difference ($p < 0.0001$). So instead we here list the *exceptions*; those comparisons which were *not* significant:

- **MC** $\times$ **MC-Fast**: No significant comparison at any level of edge factor $p > 0.9999$
- **S2T-D** $\times$ **MC**, and **S2T-D** $\times$ **MC-Fast**: No significant comparison at edge factor level .00

Analysis of the **simple main effect of edge factor** is used to determine whether edge factor had an effect on the total absolute position difference for each method:

- **S2T-S**: $F(2, 2997) = 40.5$, $p < .001, \eta^2 = .026$
- **STS-D**: No significant effect
- **MC**: $F(2, 2997) = 1461.4$, $p < .0001, \eta^2 = .494$
- **MC-Fast**: $F(2, 2997) = 1521.8$, $p < .0001, \eta^2 = .504$
- **MC-Lo**: $F(2, 2997) = 1331.9$, $p < .0001, \eta^2 = .471$
- **MC-Med**: $F(2, 2997) = 1094.6$, $p < .0001, \eta^2 = .422$
- **MC-Hi**: $F(2, 2997) = 1074.8$, $p < .0001, \eta^2 = .418$

Pairwise comparisons summarized in Table 3.2.

3.4.6.3 Total Absolute Rotation Difference

The NoRDW method does not apply gain to positions so are excluded from this analysis. The remaining data was visually inspected for normality via Normal Q-Q Plot. The MC techniques MC, MC-Half, MC-Lo, MC-Med and MC-Hi appeared to fit a normal distribution well. The heuristic techniques S2C, S2O, S2T, S2T-S and S2T-D, were consistently slightly left-skewed, but still approximately normal. With

Method	$.00 \times .15$	$.00 \times .30$	$.15 \times .30$	η^2
S2C		*	*	.005
S2O	*	*		.008
S2T				-
S2T-S			*	.004
S2T-D	*			.004
MC	*	*	*	.079
MC-Fast	*	*	*	.078
MC-Lo	*	*	*	.171
MC-Med	*	*	*	.212
MC-Hi	*	*	*	.229

Table 3.3: Pairwise comparisons for **simple main effects of edge factor** on total absolute **rotation difference**, by method. The NoRDW method applied no rotation gain so is not considered. Edge factor had no significant effect on S2T.

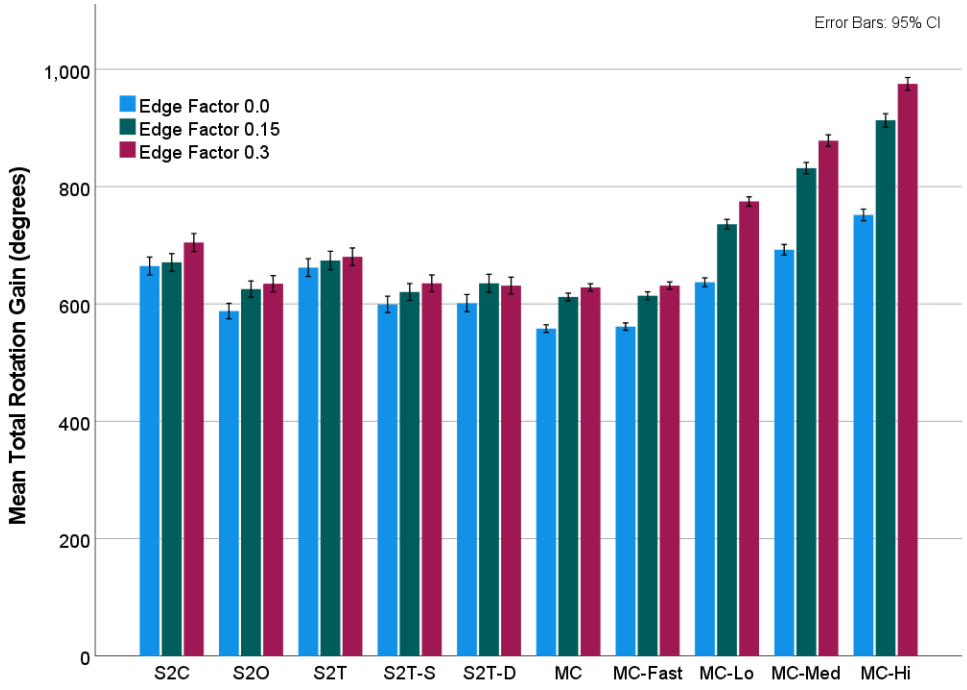


Figure 3.11: Mean total absolute **rotation difference** by method across all simulations, grouped by edge factor.

large sample sizes, ANOVA is considered robust to violations of normality so we include the heuristic techniques [92].

The data included a very small number $(n \leq 1)$ of outliers (± 4 standard deviations) among all included methods. As the sample size is large, we include these values regardless. We omit no values from the dataset.

There was homogeneity of variance for all methods except for MC-Hi ($p < 0.001$). However, the data violated Box's test for homogeneity of covariance. These tests are known to be sensitive with large sample sizes, and with groups of equal size mixed ANOVA is considered robust to heterogenity of variances and covariances [92]. Additionally, visual inspection of scatter plots of residuals showed the expected shape, so we conclude the assumptions of the mixed ANOVA are not violated and perform no transformations on the data.

There was a statistically significant interaction with a small effect size between the method and the level of edge factor on total absolute rotation difference, $F(12.5, 18706.4) = 46.6$, $p < .001$, partial $\eta^2 = .03$. A Greenhouse-Geisser correction was used ($\varepsilon = .694$) as Mauchly's test of sphericity indicated that the assumption of sphericity was violated for the two-way interaction, $\chi^2 = 9838.7$, $p < .0001$.

Analysis of the **simple main effect of method** is used to determine whether method had an effect on the total absolute rotation difference at each level of edge factor:

- **Edge-Factor .00**: $F(5.9, 5878.4) = 129.6$, $p < .001, \eta^2 = .115$
- **Edge-Factor .15**: $F(6.3, 6253.2) = 347.1$, $p < .0001, \eta^2 = .258$
- **Edge-Factor .30**: $F(6.5, 6461.7) = 494.4$, $p < .0001, \eta^2 = .331$

Pairwise comparisons follow. Unlike boundary collisions and position gain, many of these comparisons showed no significant difference. For reasons of space we summarise notable patterns below:

- **MC and MC-Fast**: No significant comparison at any level of edge factor $p > 0.9999$.
- **MC-Lo**, **MC-Med** and **MC-Hi**: Significant comparisons with all other methods at all levels of edge factor

Analysis of the **simple main effect of edge factor** is used to determine whether edge factor had an effect on the total absolute rotation difference for each method:

- **S2C**: $F(2,2997) = 7.8$, $p < .001, \eta^2 = .005$
- **S2O**: $F(2,2997) = 12.7$, $p < .001, \eta^2 = .008$
- **S2T**: No significant effect
- **S2T-S**: $F(2,2997) = 6.2$, $p \approx .002, \eta^2 = .004$
- **STS-D**: $F(2,2997) = 5.9$, $p \approx .003, \eta^2 = .004$
- **MC**: $F(2,2997) = 127.9$, $p < .001, \eta^2 = .079$
- **MC-Fast**: $F(2,2997) = 126.4$, $p < .001, \eta^2 = .078$
- **MC-Med**: $F(2,2997) = 304.0$, $p < .001, \eta^2 = .171$
- **MC-Med**: $F(2,2997) = 402.6$, $p < .001, \eta^2 = .212$
- **MC-Hi**: $F(2,2997) = 445.8$, $p < .001, \eta^2 = .229$

Pairwise comparisons summarized in Table 3.3.

3.5 Discussion

We loosely structure this discussion around the three independent variables. Where helpful we consider the variables together.

3.5.1 Total Boundary Collisions

Overall the family of MCRDW techniques significantly reduced collisions when compared with the heuristic approaches (see Figure 3.9). At edge factor 0 in our simulations, MCRDW reduced collisions by over 75% when compared with S2C and over 50% when compared with the previous best technique, S2T-S. While performance gains were still good at higher levels of edge factor, there was a notable performance reduction. As an environment becomes more open, path prediction becomes more challenging, so we make less accurate guesses about which path a user might take. However, MCRDW methods did continue to outperform heuristic methods even with very open environments at edge factor 0.3.

The over-threshold MCRDW methods MC-Lo, MC-Med and MC-Hi demonstrated a linear performance gain correlated with the level of over-thresholding permitted. The effect was significant. However, as covered later in this Section, these performance improvements came at the cost of noticeably higher total gain overall.

No significant difference was found between MC and MC-Fast ($p > 0.9999$, see Figure 3.9). This is a good result and likely indicates the method had more than enough time for computation for the environments in our simulations. The available computation time could likely have been reduced further to find the point at which it begins affecting performance. However, the aim of this experiment is to discover whether simulations can be a robust and performant approach to finding good redirection strategies, and this appears to be well demonstrated. Any of the MCRDW techniques can operate as a lightweight background process in the kind of machines capable of driving VR. This consitutes a substantial improvement over a brute force approach which would not be possible in real time.

Based on [56], amongst the heuristic techniques S2C, S2O and S2T, we expected S2C to outperform S2O as S2O is unable to lead a user around a large circle in a confined space. We also expected S2T to slightly outperform S2C due to more consistent handling of rotations. In practice, these expectations were accurate and are present in the data. However, only very small (but statistically significant) differences were observed. These methods performed similarly across edge factor levels.

One surprising result was the performance difference observed between S2T-S and S2T-D. Theoretically S2T-D directs the user towards the center of their space more efficiently than S2T-S, but overall S2T-S performed significantly better. One possible explanation would be that during long straight walks S2T-S maximized the overall physical space while S2T-D had a more neutral effect.

Another surprising result is that these methods and NoRDW performed differently across edge factor levels, despite using no form of prediction. This likely indicates that edge factor is not a perfect method for adding branching to environments. We can see why this occurs from the path generation algorithm described in Sec-

tion 3.4.2. We only generated paths after already adding the branching edges from edge factor. As a result, edge factor also has an effect on our path generation, and the kind of paths generated. This is a methodological flaw which could be avoided by generating paths before adding edge factor. However, it is unlikely to have effected the results materially, as the effect size is notably very small when compared with those techniques which do use path prediction: $\eta^2 < .009$ vs $\eta^2 > .263$.

3.5.2 Total Absolute Position Difference

MC reduced total absolute position difference when compared with S2T-S. However, the S2T-S method applies the maximum level of gain for the entire walk and MC only slightly reduces the level of gain applied (see Figure 3.10). As a result, MC does appear to favour position gain and used it a great deal in our simulations. As with the other metrics, MC and MC-Fast had similar results.

The superior boundary collision performance of the over-threshold methods MC-Lo, MC-Med and MC-Hi came with the cost of higher position difference overall. There was a possibility that these over-threshold methods would not dramatically increase total gain due to their scoring mechanism favouring low gain strategies. However, this is not borne out by the data, especially as in our simulations the MCRDW family of methods appeared to select redirection strategies with position gain frequently. This could perhaps be achieved through better weights, or a more intelligent scoring mechanism.

Edge factor had a notable impact on the level of position gain applied (see Table 3.2 for effect sizes). However, this is likely a result of low standard deviations among position difference results across techniques. Discrepancies in mean total absolute position difference are small in practice. The likely explanation is that those layouts with higher edge factor more frequently placed the user in situations likely to lead to boundary collision. As a result, the small bias in favour of not applying translation gain is more frequently ignored.

3.5.3 Total Absolute Rotation Difference

MC had the lowest rotation difference overall by a small but significant factor, though this gap closed at higher edge factors. Interestingly at low edge factors, the over-threshold methods MC-Lo, MC-Med and MC-Hi saw slightly increased rotation differences overall, but significant performance gains. This may be a good argument for allowing at least rotation gain to slightly exceed established thresholds in challenging situations, as it is possible that MCRDW is able to only apply it in situations where it is necessary. However, confirming this would require additional outcome measures that were not included in this experiment. Again as with position difference and boundary collisions, no significant difference was found between MC and MC-Fast $p > 0.9999$.

Edge factor had a less notable impact on the level of rotation difference than it did with position difference (see Table 3.3 for effect sizes). However, as with position difference, the effect was particularly notable for the over-threshold methods. Likely this is for the same reason; layouts with higher edge factor more frequently put the simulated user on path towards boundary collision, overwhelming the small bias towards low gain strategies.

3.6 Summary

This chapter introduces the MCRDW algorithm, a method for selecting redirection strategies which minimises boundary collisions through simulated walks. We include an evaluation of MCRDW under a variety of different conditions, and a comparison with existing methods. In our evaluation, the MCRDW family of methods significantly reduce boundary collisions when compared with the heuristic methods by over 50% while reducing total rotation or position gain. No significant difference was found between MC and MC-Fast ($p > 0.9999$) on all metrics. MCRDW can therefore make good decisions without needing as much computation time as allotted within our simulations. The technique can be considered lightweight.

The over-threshold MCRDW methods MC-Lo, MC-Med and MC-Hi significantly outperformed all other techniques. However, for these over-threshold meth-

ods specifically, this comes at the cost of significantly increasing total rotation and position gain. This indicates the methods resorted to over-threshold gains too often. This behaviour could potentially be avoided by tweaking the associated 'score' penalties, or by introducing a more intelligent *subtlety* metric.

While MCRDW methods significantly outperformed heuristic methods at all levels of edge factor, MCRDW performance was more closely linked to the virtual environment than with heuristic methods. This is likely due to simple environments having fewer potential routes. This makes the simple path prediction used in our implementation more reliable. Quality of available path prediction should be considered a significant factor in MCRDW performance. Advanced path prediction techniques would be a valuable avenue of future research.

Chapter 4

Sensitivity to Rate-of-Change of Rotation Gain During Redirected Walking

This chapter details the design and results of a user study on the effects of rate of gain change. The primary contribution of this chapter is a set of results that indicate that slow gain change is significantly harder to detect than sudden gain change. The work in this chapter was published in a modified form under the name 'Sensitivity to Rate of Change in Gains Applied by Redirected Walking' in the 25th ACM Symposium on Virtual Reality Software and Technology (VRST '19) [1].

4.1 Introduction

In Chapter 3 we described an approach for improving redirected walking performance through an environment-aware gain selection algorithm. The goal of this algorithm and of other gain selection techniques is to identify the level of movement gain which is most likely to keep a user within their physical space. These gain levels can then be applied to the user as they walk and turn with the ultimate aim of minimising collisions.

Existing studies have established guidelines on perceptual limits for gain [40, 52]. These limits are the level of gain at which a user cannot reliably identify whether gain is being applied; the thresholds are discussed in detail in Section 2.5.1.

The form of gain these studies used for their analysis was static rotation and translation gain applied multiplicatively to user movement. However, the techniques described above require gain levels to change frequently as the user moves around the environment in order to perform well. While perceptual thresholds for static gain have been well addressed, user perception of gain changes are comparatively little understood.

It has been demonstrated that the above thresholds can change as a result of external factors. Neth et al. found that sensitivity to rotation gains increases at higher walking speeds; i.e., that lower walking speeds allow greater rotation gains [35]. Serafin et al. found that with no visual stimulus, users could be virtually turned 20% more or 12% less than their physical rotation through spatialised audio cues [57]. However, Nilsson et al. found that when both audio and visual information is available, redirection detection thresholds were the same for spatialised, static and disabled audio [58].

Rotation gain rate-of-change is a possible factor which has received only a small amount of research attention. Zhang examined the effect of rate of gain change on perception thresholds and found no significant difference between gradually changing rotational gains compared to instantaneously changing the gain during 360 degree turns [54]. These results run counter to typical redirected walking implementations which use a smoothing term to avoid instantaneous gain changes and increase rotation gain with walking speed [55, 63, 93]. The purpose of this chapter is to examine this discrepancy and to further investigate the effects of rate of gain change.

The method of constant stimuli is the typical basis for experimental design on this topic [40, 52, 35]. The aim of the experiment is to identify the level at which a stimulus (such as rotation gain) is perceptible to a user. With this approach a range of gain levels is identified which is believed to contain the upper and lower gain thresholds. Participants are exposed to each gain level multiple times in a random order and are asked whether their virtual turn was larger or smaller than their physical turn. The responses at each gain level are then fit to a curve. The

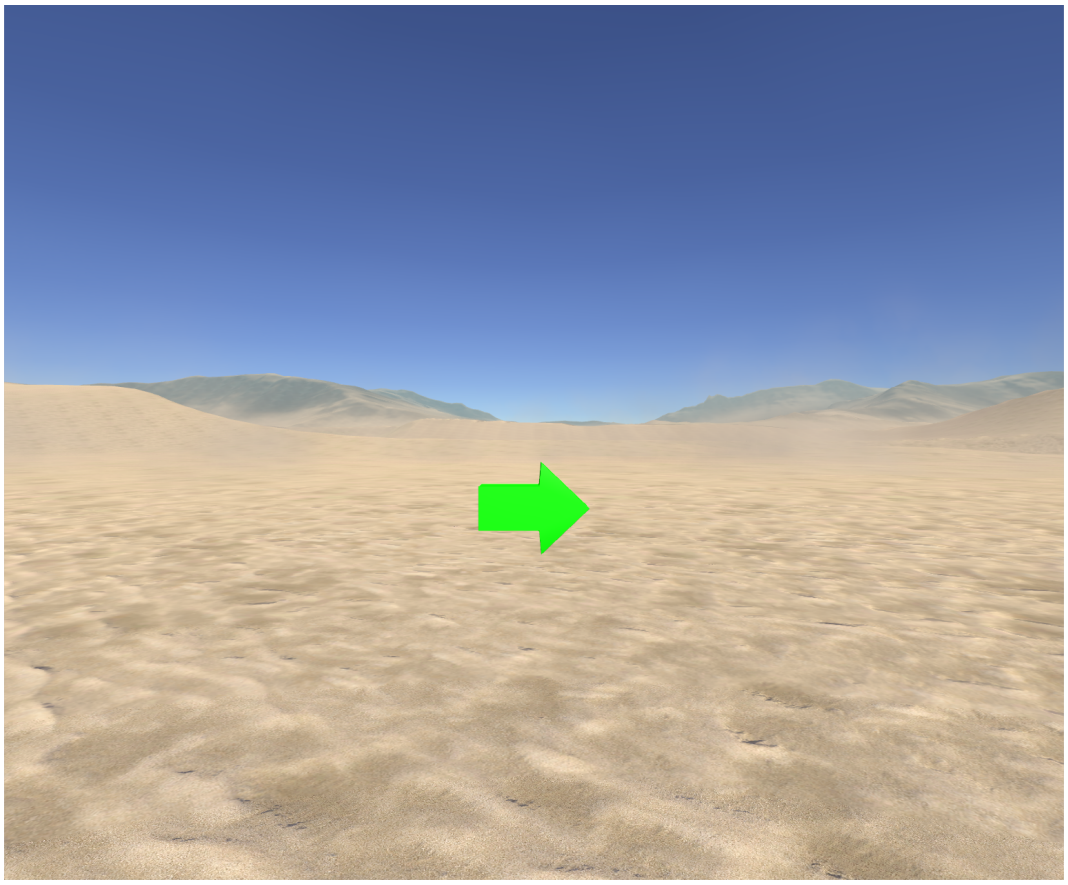


Figure 4.1: Example view of the virtual environment used for the study. Participant position and orientation were randomized between trials.

upper and lower detection thresholds are those points on the curve where the 75% of the participant's responses accurately identified the direction of virtual gain.

Staircase procedures are an alternative experimental approach with the same objective [94]. In this case the participant is once again subjected to a large number of trials, each with its own level of stimulus, and asked to identify whether the stimulus is present. However, the participant is not shown all the stimulus levels randomly interspersed. Instead, the stimulus level is dynamically adjusted by participant responses. In a redirection context, this means that should the participant correctly identify the gain direction the gain level is too high and can be decreased, and should the participant incorrectly identify the gain direction the gain level is too low and can be increased.

Staircases can more precisely estimate thresholds than non-adaptive procedures as less of the participant's time is spent on stimulus levels which are not

close to the threshold [95]. Additionally, staircases typically require fewer trials to find their estimate [96]. This is significant for the work in this chapter as we intend to generate and compare multiple thresholds which would otherwise require participants to remain in virtual reality for long periods. Reducing study time per participant is particularly useful in this context as simulator sickness may become a factor when a participant is exposed to high levels of gain [4].

This chapter addresses the question of user sensitivity to sudden gain changes. Specifically, we are interested in whether sudden gain changes are more noticeable than slowly increasing gain. The motivation for this work is to provide a reliable indication of whether quickly changing rates of gain as used in redirected walking has an effect on users, and whether this should be mitigated by a smoothing term in redirected walking implementations. Our approach is to determine whether different rates of gain change affect the level of gain at which users perceive inconsistencies between their virtual and physical motion.

To this end, this chapter presents the design and results of a user study on the effects of rate of gain change. The study takes the form of a psychophysical experiment with 21 participants. Each participant completed a series of two-alternative forced choice tasks in which they determined whether their virtual motion differed from their physical motion while experiencing one of three different methods of gain change: sudden gain change, slow gain change and constant gain. Gain thresholds were determined by 3 interleaved 2-up 1-down staircases, one per condition. The primary contribution of this chapter is a set of results that indicate that slow gain change is significantly harder to detect than sudden gain change. This finding can be integrated into existing RDW implementations to improve user comfort and potentially increase thresholds.

The structure of the chapter is as follows. Section 4.2 covers the experiment in detail. Section 4.3 and 4.4 present and discuss the results respectively, and Section 4.5 summarizes the chapter.

4.2 User Study

Redirected walking allows the mapping between real and virtual motion to be modified by applying scaling ("gain") to user motion. In this section we present an experiment designed to identify whether rate of gain change affects user experience of motion gain. Each participant was exposed to 3 conditions representing alternative methods of gain change: sudden gain change, slow gain change and constant gain. Participants were divided into two groups, experiencing either larger or smaller virtual turns. For each trial participants physically turned on the spot and a corresponding virtual turn was calculated from the physical motion based on condition and group. Gain levels were selected by following a staircase procedure. One staircase was used per condition and all were interleaved. The result is three thresholds (one per condition) for each participant indicating the level at which a discrepancy between virtual and physical motion can be reliably detected. We expect to find significant differences between the thresholds for each condition.

4.2.1 Study Design

Participants were asked to complete a series of physical turns while in virtual reality. Each participant wore a HMD and held one controller. Participants were asked to turn only with their feet while looking directly forward and without moving their neck, shoulders, waist or hips. This was stipulated so that motion would be consistent across trials and participants. As participants turned their physical motion was mapped to virtual position as determined by their condition, further described in Section 4.2.4.

Trials followed a precise series of stages within the virtual experience. Participants started each trial with their view faded to grey. Participants were asked to stand up straight and look forward, then invited to press a button on the controller when they were ready to progress. When participants pressed a button, the virtual environment faded in. A small grey sphere was placed in front of the participants. The sphere was replaced by a green arrow when participants' view had fully cleared. Participants then turned in the direction indicated by the arrow. Left and right turns were interspersed to prevent the participant becoming tangled in the HMD cable.

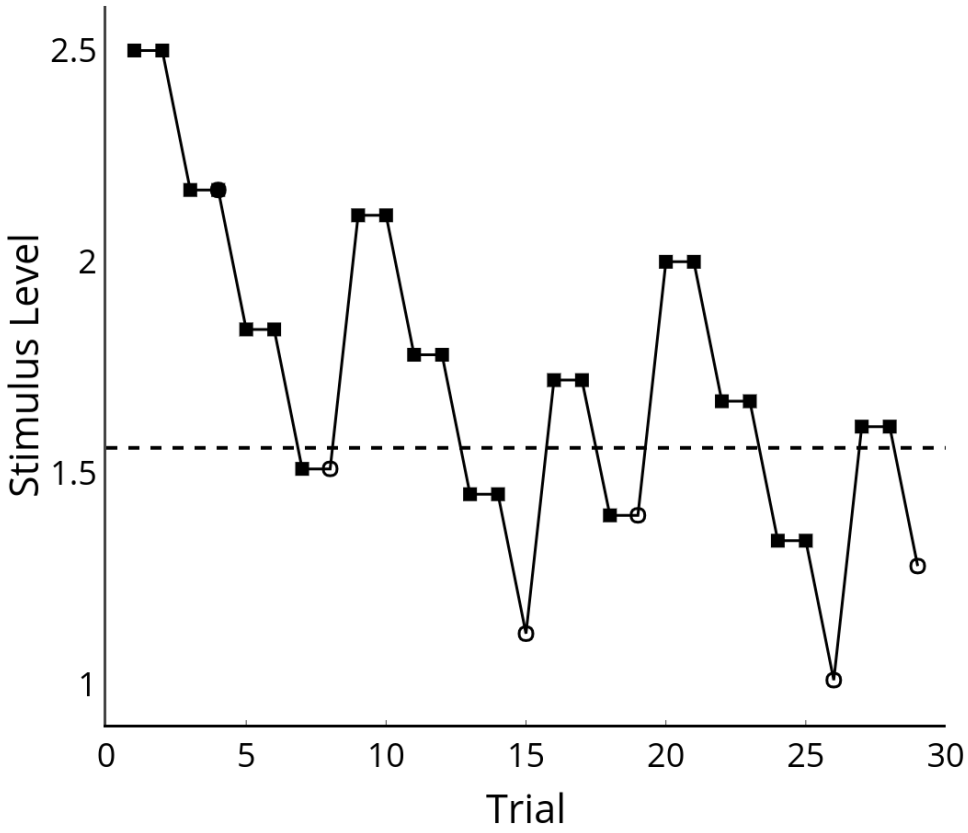


Figure 4.2: One participant's staircase for the *delay-max* condition. Markers indicate responses to the 2AFC question. Empty markers indicate the stimulus was not detected. The dashed line indicates the threshold estimate for this participant and condition.

Once the participant turned a pre-determined physical angle, the environment faded out. The participant was asked about the turn with a two-alternative forced choice (2AFC) question (further discussed in Section 4.2.2). Finally, the response was recorded and the participant given the opportunity to rest before the next turn.

Trials where the participant turned too quickly, slowly or inconsistently were rejected. For speed, participants were required to turn physically at between 20 and 80 degrees per second averaged across the entire turn. The test for inconsistency required that participants not turn more than 35 degrees in any 0.25 second window. Trials were also tested to ensure participants did not turn more than a total of 10 (physical) degrees against the desired direction of motion. Should a turn fail, one of the tests the 2AFC question would be skipped and the participant informed which test had not been passed. As above, these tests were calculated based on physical

turn, ignoring gain.

Participants were divided into two groups. Physical turns for the *virtual-larger* group and the *virtual-smaller* group were mapped to larger and smaller virtual turns respectively. The physical angle for each turn was selected randomly from a range of 100 to 130 degrees. The point at which a turn was deemed complete was based entirely on physical turn, ignoring gain magnitude and gain condition.

The gain for each turn was selected following a staircase procedure where levels of gain were adjusted up and down based on responses to the 2AFC question [94]. Additional "neutral" trials with no gain applied were also interleaved with the trials required for the staircases. The intention with these trials was to prevent participants acclimatizing to high levels of gain. Details of the staircases used including all parameters and the interleaving mechanism for neutral trials can be found in Section 4.2.2.

Each participant completed a short training step before the experiment began. A total of 8 training trials were used, 2 with no gain applied and 2 interspersed trials for each condition where the maximum gain was set to 5. Participants were told when a training turn was the same as their physical movement and when gain was applied. The intention behind the training step was to allow participants to familiarize themselves with the sense of movement in virtual reality and to practice the particular turning movement and speeds we required. Additionally, by using a very high level of gain during the training step the aim was for participants to be able to recognize gain being applied and not discount it as a problem with the equipment or software. Finally, before starting the experiment each participant completed 2 further trials with no gain applied to re-familiarize with normal turning motion.

The virtual environment was designed to allow participants to have a sense of their own motion while avoiding discrete markers that could be used to orientate across trials. We specifically aimed to avoid 90 degree angles (e.g., on buildings or crossroads). The result was the desert environment seen in Figure 4.1. The environment was textured and included gradual slopes and a skybox to provide enough information for both optical flow and landmark recognition [97]. A selection of 9

spawn points were chosen and the participant moved between these randomly while their view was faded to grey between trials. The participant's starting (virtual) orientation in the transverse plane was also randomized. No spawn point was visible from any other, with the intention of preventing participants from "learning" the environment as much as possible.

4.2.2 Staircase Procedure

Staircase procedures are adaptive psychophysical techniques that aim to estimate the level at which a stimulus (in this case, the maximum extreme of gain, high or low depending on group) is perceptible to a user. Following this method, gain levels were dynamically adjusted up and down between each turn based on participant responses. An example of a participant's staircase for one condition can be seen in Figure 4.2. Responses were in the form of a 2AFC where participants received the prompt "Compared with my physical movement..." and were asked to choose from two options. The first option was always "My movement in the virtual world was the same as my physical movement", indicating the participant did not detect the stimulus. The second option was either "My movement in the virtual world was smaller than my physical movement" or "My movement in the virtual world was larger than my physical movement" depending on whether the participant was in the *virtual-smaller* or *virtual-larger* group respectively. This option indicated the participant detected the stimulus.

Often multiple "positive" responses are required to reduce the stimulus level when using staircase procedures. Different staircase designs provide thresholds at different levels on the psychometric curve, i.e., at different levels of user certainty. We used a 1-up 2-down staircase to approximate the level at which participants can identify whether the stimulus is present with a success rate of 70.7% [96]. With this method the stimulus would be decreased if the participant indicated they detect the stimulus twice in a row, and would be increased if the participant indicates they do not detect the stimulus once. Thresholds were calculated by averaging the stimulus values at "reversal" points - the levels at which the staircase changes direction.

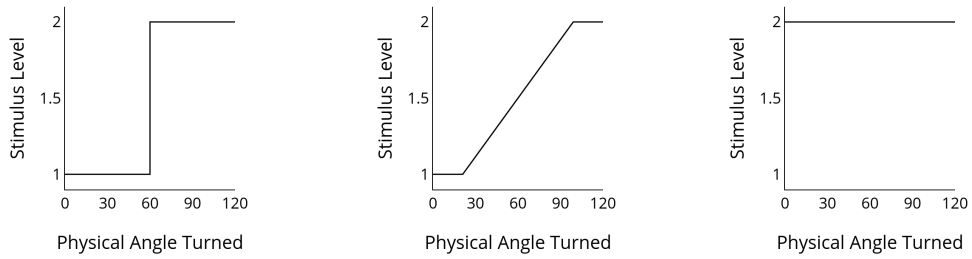
Each staircase was run until it had reversed 8 times, ignoring the first response

where initial direction was determined, and the experiment continued until all three staircases were complete. The final 7 reversal values were averaged to generate the threshold for each condition. The spread is the range of stimulus values over which the underlying psychometric function is non-asymptotic. Based on existing studies [40, 52] we anticipated a spread of 1 based on stimulus levels between 1 and 2. For each downward step the stimulus was decreased by 0.32928 and each upward step the stimulus was increased by 0.6, for a ratio of 0.5488 [98]. The initial stimulus was set at 2.5, significantly above the presumed threshold, to give participants a sense of extremes of gain [99]. The lower limit on stimulus was 1. Should the staircase reach this point positive detection responses were still accepted but the stimulus would not be decreased. No upper limit on stimulus was stipulated.

Three staircases were used to accommodate the three conditions. These staircases were interleaved, meaning that subsequent trials would be drawn from different staircases. Each staircase was entirely self-contained and the response to each trial only affected the staircase from which that trial was selected. The purpose of interleaving is to obfuscate the staircase mechanism from the perspective of the participant, encouraging them to rely on their observations rather than guessing at the staircase progression [96]. The interleaving procedure was random except for the following constraints. Firstly, that every four trials included at least one turn for each incomplete staircase. Secondly, that every four trials included exactly one neutral trial. Finally, that no trial from already completed staircases was included.

4.2.3 Stimulus and Gain

Section 4.2.1 describes how the stimulus level provided by staircases increases and decreases with participant responses. The stimulus level is bounded between 1 and infinity and always decreases when participants are aware of gain, increasing otherwise. Stimulus should therefore be considered a measure of the intensity of gain to be applied. We distinguish between stimulus and gain so staircases can increase or decrease in constant units, and so the two groups (*virtual-larger* and *virtual-smaller*) can be compared. Here we describe how gain is calculated from the stimulus level. Note that as the gain applied at any given moment during a turn depends



(a) Delay-Max **(b)** Slow-Increase **(c)** Start-Max

Figure 4.3: Stimulus levels at points during the participant's turn for each of the the three conditions assuming an s of 2. The gain applied is calculated from the stimulus level based on group (see Section 4.2.4). Due to randomized parameters the actual stimulus values for each trial may differ slightly.

upon the condition, stimulus s instead controls the target gain (x_{target}) reached during a turn. This mapping varies by participant group:

$$x_{target} = \begin{cases} s & \text{if } \textit{virtual-larger} \\ s^{-1} & \text{if } \textit{virtual-smaller} \end{cases} \tag{4.1}$$

4.2.4 Conditions

The study uses a within-subjects design with 3 conditions representing different approaches to rate of gain change. The condition determines moment to moment gain within a trial. Visualizations of the three conditions can be found in Figure 4.3 and Figure 4.4.

As participants follow the procedure outlined in Section 4.2.1 their physical turn is mapped to a larger or smaller virtual motion. The specifics of the mapping are dependent upon the condition selected for the trial. Each trial uses one of the three conditions described in this section. These conditions are not intended to be methods for achieving redirected walking but rather to be representative of different contexts in which users might experience gain change: either sudden gain change, slow gain change or no gain change at all (that is, constant gain). Physical and virtual head orientations consist of a 3-dimensional vector such that:

$$\mathbf{r} = (\theta_s, \theta_t, \theta_f) \tag{4.2}$$

Where θ_s, θ_t and θ_f are the angles of orientation in the sagittal, transverse and frontal planes respectively (equivalently pitch, yaw and roll). Gain is applied to participant motion regardless of whether direction is with or against the direction specified by the trial. However, as is common with redirected walking, we only apply gain in the transverse plane [55]. We therefore eschew a 3-dimensional gain vector and focus on the transverse gain x. Virtual turns are calculated from the physical with this gain factor:

$$\theta_{t,virt} = x \cdot \theta_{t,phys} \tag{4.3}$$

Should $x = 1$ a virtual turn will match the physical turn one-to-one. If $x > 1$ the virtual turn be larger, possibly giving a user the impression the world is turning against them. By contrast, when $x < 1$ the virtual turn will be smaller, giving users the sense that the virtual world is turning with them. For example, should a user physically rotate 180 degrees in the transverse plane while looking forward, gains of $x = 0.5$, $x = 1$ and $x = 2$ will cause the virtual camera to rotate 90, 180 and 360 degrees respectively, also in the transverse plane.

4.2.4.1 Start-Max

In trials with the *start-max* condition gain is immediately set to the maximum gain determined by stimulus and remains at max until the trial is complete. This condition is intended to provide gain without abrupt gain change.

$$x = x_{target} \tag{4.4}$$

The gain is set before the participant's view has faded in and continues until the

view has entirely faded out. The screen which participants observe between trials has no visual features to prevent the sensation of sudden gain change at the start or end of the trial.

4.2.4.2 Slow-Increase

Initially no gain is applied (i.e., $x = 1$). As the participant turns the gain is slowly interpolated towards the target. The target gain will always be reached before the trial is complete. This condition was intended to provide gain through a long period of very low gain change.

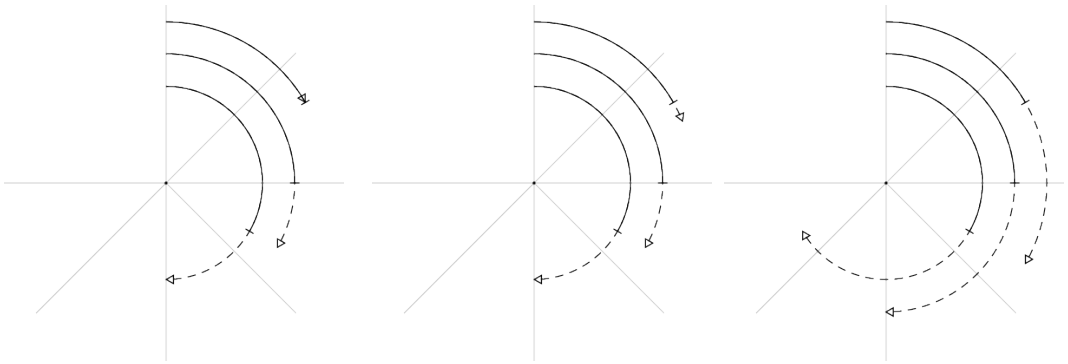
$$x = 1 + (x_{target} - 1) \cdot \text{clamp}\left(\frac{\theta - \theta_{begin}}{\theta_{complete} - \theta_{begin}}\right) \tag{4.5}$$

Where clamp is a function that clamps between 0 and 1, θ is the angle currently turned, θ_{begin} is the angle at which interpolation is started and $\theta_{complete}$ is the angle at which interpolation ends. The angles θ, θ_{begin} and $\theta_{complete}$ all refer to the participant's physical angle before gain is applied. The function clamp limits the input to a value between 0 and 1 inclusive. Turning against the target direction did not increase θ. The value of θ_{begin} and $\theta_{complete}$ depended upon the θ_{total}, the total physical angle selected for the trial:

$$\theta_{begin} = s_0 \cdot \theta_{total} \tag{4.6}$$

$$\theta_{complete} = \theta_{begin} + s_1 \cdot (\theta_{total} - \theta_{begin}) \tag{4.7}$$

Where s_0 and s_1 are configuration values. In our experiment s_0 was a randomized variable between 0.125 and 0.175 and s_1 was set to 0.65. The intention was that the user experience the target level of gain for some time before the turn was complete. Turns against the target direction did not increase the gain.



(a) Delay-Max **(b)** Slow-Increase **(c)** Start-Max

Figure 4.4: A participant's virtual (dotted) and physical (solid) turn angle for each of the three conditions, under *virtual-larger* with a stimulus (and therefore x_{target}) of 2. Starting from the outside, the concentric circles represent the angle of the participant after they have physically turned 60, 90 and 120 degrees respectively. Each condition increases gain at varying rates throughout the turn before reaching the same maximum level.

4.2.4.3 Delay-Max

Initially no gain is applied. Once the participant reaches a pre-determined point in their turn, gain is immediately set to the target. This condition was intended to provide gain through a short period of very high gain change.

$$x = \begin{cases} 1 & \text{if } \theta < \theta_{delay} \\ x_{target} & \text{if } \theta \geq \theta_{delay} \end{cases} \tag{4.8}$$

Where θ is the physical angle currently turned and θ_{delay} is the physical angle at which gain is set to the target. Turning against the target direction did not increase θ. The value of θ_{delay} depended upon θ_{total}, the total physical angle selected for the trial:

$$\theta_{delay} = d \cdot \theta_{total} \tag{4.9}$$

Where d is a configuration value. In our experiment d was a randomized variable between 0.35 and 0.65.

4.2.5 Outcome Measures

The primary outcome measures were the thresholds generated by the staircases for each condition (see Section 4.2.2). Participants were also asked to complete a short questionnaire on their age, gender and level of virtual reality experience. This information was collected for analysis alongside the thresholds to see if there was an effect on general threshold levels or response to individual conditions. Information on participant's level of virtual reality experience was gathered via multiple choice question with the options: "Never", "1-2 times", "3-5 times", "6-10 times", "10-50 times" and "50+ times". For the purpose of analysis, participants who chose one of the first three responses were considered "inexperienced" while those who selected one of the final three responses were considered "experienced". Additionally, participant responses to the 2AFC were recorded during neutral trials to get a sense of each participant's general level of accuracy. Telemetry data (head position and orientation) was gathered throughout each trial.

4.2.6 Study Setup

Participants wore a HTC Vive Pro headset and held one Vive hand controller. The display has a resolution of 1440 x 1600 per eye, a refresh rate of 90Hz and a vertical field of view of 110 degrees. The Vive native tracking system was used for position and orientation information. The HMD was set to the mean interpupillary distance of 63mm [100]. The virtual environment was developed in Unity3D with the SteamVR plugin. The system running the experiment was using the Windows 10 operating system and was powered by a 6 core Intel CPU, 16GB of memory and an NVIDIA GTX 2080. The maximum refresh rate of 90 frames per second was maintained throughout the study.

The HMD was connected to the system via wire. Occasionally participants would drift from the center of the room. When this occurred the experimenter would pause the experiment and guide the participant back. With the exception of this and the training step, all information relevant to the experiment was contained within the virtual environment such that no communication or intervention was required between participant and experimenter and participants were able to proceed at their

own pace.

4.2.7 Study Protocol

The study took approximately 30 minutes per participant. On arrival the experimenter gave the participant an information sheet describing the study and invited the participant to read the sheet and ask any questions they may have. The experimenter then asked the participant to complete the pre-questionnaire (see Section 4.2.5).

The experimenter briefly described visual/vestibular conflict and the participant was told the goal of the study was to better understand this conflict. The mechanics of virtual gain were explained to the participant and they were told whether to expect a *virtual-larger* or *virtual-smaller* stimulus. The individual conditions (described in Section 4.2.4) were excluded from the discussion.

The participant was shown to the track space and provided with the HMD and their controller which were adjusted for fit. The experimenter loaded the virtual environment and proceeded with the training step. The experimenter then started the core study step and allowed participants to proceed at their own pace. Both training and study steps are as described in Section 4.2.1.

The study step continued until the virtual environment reported all staircases had concluded. The experimenter asked participants to remove the HMD. The participant was debriefed and their travel expenses compensated. Finally, they were given the opportunity to ask questions and give feedback.

4.2.8 Participants

Participants were recruited from graduate study mailing lists and external advertisements and paid £5 to cover travel expenses. Participants were required to be between the ages 18 and 65 and to be able to walk unassisted.

A total of 23 participants were recruited. 2 participants became nauseous and were not able to complete the study. These results were excluded from analysis. Of the remaining 21 participants, 8 were female and 13 male and the mean age was 28.57 with a standard deviation of 7.72. This study was approved by the University College London Research Ethics Committee, approval number 4547_12.

Table 4.1: Threshold stimulus levels for each combination of group and condition.

Group	VR-Experience	N	Condition	Mean	SD
virtual-smaller	inexperienced	5	delay-max	2.191	0.869
			slow-increase	4.368	2.331
			start-max	2.895	1.394
virtual-smaller	experienced	6	delay-max	1.551	0.405
			slow-increase	2.115	0.591
			start-max	2.198	1.405
virtual-larger	inexperienced	5	delay-max	5.574	2.273
			slow-increase	6.274	2.528
			start-max	4.325	1.367
virtual-larger	experienced	5	delay-max	1.579	0.241
			slow-increase	1.800	0.270
			start-max	1.774	0.212

4.3 Results

Table 4.1 shows the mean and standard deviation for each combination of group and condition. Note that these refer to stimulus values rather than gain. Higher stimulus threshold values indicate that a further extreme of gain is required to make the condition perceptible (i.e., that the condition is harder to detect). Two participants in the *virtual-larger* group were found to have very high threshold values (> 8 stimulus). As the results showed clear differences between conditions and accuracy when responding to neutral trials was good ($> 90\%$) it appeared that the participants had understood the task so the results were included in the analysis.

A mixed ANOVA was conducted to investigate the effect of rate of gain on detection thresholds for the *virtual-smaller* group. The within-subjects factor was the threshold stimulus level under the three gain conditions: *slow-increase*, *delay-max* and *start-max*. The between-subjects factors were gain direction and virtual reality experience. Mauchly's test of sphericity indicated that the assumption of sphericity had not been violated, $\chi^2 = 1.029$, $p = .598$. There was homogeneity of variances for the *start-max* ($p = .837$) and *delay-max* ($p = .196$) conditions, but not the *slow-increase* condition ($p = .007$). ANOVA is robust to violations of this type when group sizes are approximately equal, so we continue with the analysis.

The main effect of condition showed a statistically significant difference in thresholds between gain conditions, F(2,18) = 7.724, p = .004, partial η^2 = .462. Post hoc pairwise comparisons with Bonferroni adjustment indicated a significant difference between *slow-increase* and *delay-max* conditions, p = 0.017. No significant pairwise differences were found between the *start-max* condition and the other conditions. The two-way interaction between condition and virtual reality experience was not significant, F(2,18) = 3.442, p = 0.054, partial η^2 = .277. We include pairwise comparisons due to the narrow margin. For the inexperienced group, we found a significant difference between *slow-increase* and *delay-max* p = 0.011. For the experienced group, no significant difference between conditions was observed.

Another mixed ANOVA was conducted for the *virtual-larger* group. The within and between subjects factors were as above. The main effect of condition showed a statistically significant difference in thresholds between gain conditions, F(2,16) = 6.682, p = .008, partial η^2 = .455. Post hoc pairwise comparisons with Bonferroni adjustment indicated a significant difference between *slow-increase* and *delay-max* conditions, p = 0.016. No significant pairwise differences were found between the *start-max* condition and the other conditions. The two-way interaction between condition and virtual reality experience was significant, F(2,16) = 6.856, p = 0.007, partial η^2 = .461. For the inexperienced group, we found a significant difference between *slow-increase* and *delay-max*, p = 0.011, and *delay-max* and *start-max*, p = .008. For the experienced group, no significant difference between conditions was observed.

4.4 Discussion

Our results indicate there is a significant difference between thresholds for *delay-max* and *slow-increase*. This supports the hypothesis that users are sensitive to sudden changes in gain. While there was a large variance between participants in absolute thresholds, all 21 participants had a higher threshold for *slow-increase* than *delay-max*. The *slow-increase* and *delay-max* conditions can be effectively compared as they fundamentally operate similarly. Participants started trials for

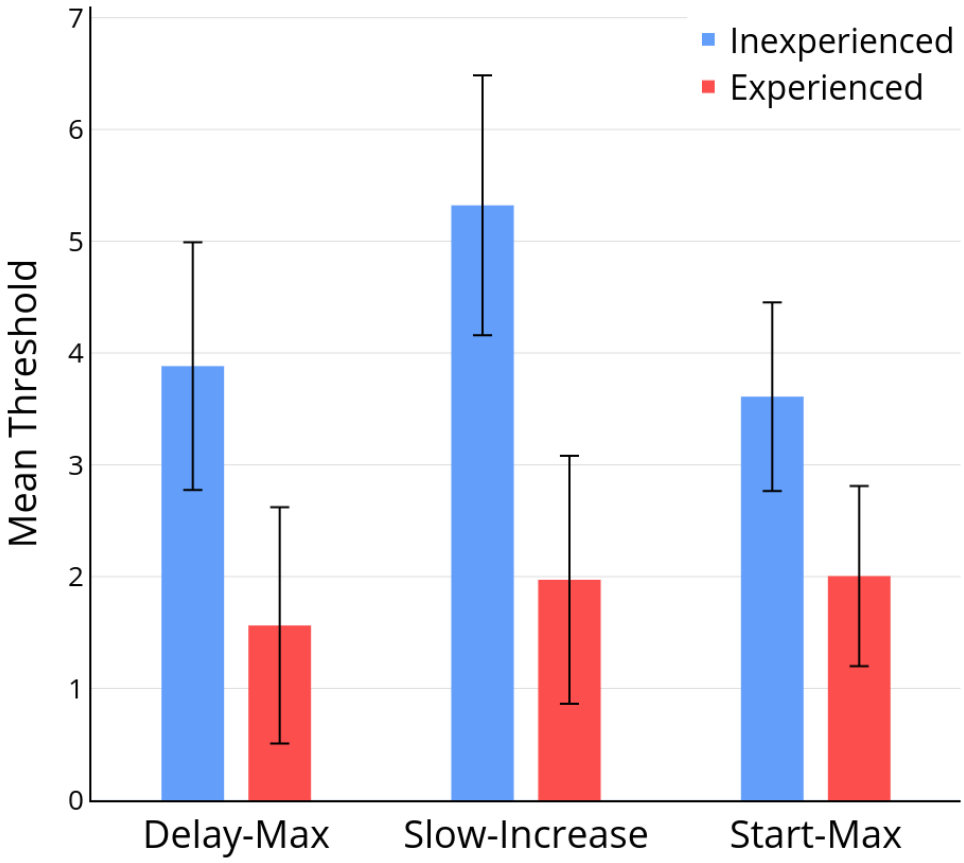


Figure 4.5: Mean stimulus thresholds across conditions for the *experienced* and *inexperienced* groups. Error bars are 95% CI.

these conditions at zero stimulus and ended trials at the gain target, with the stimulus level increasing as they turned. Likewise trials with these conditions covered approximately similar total virtual angles, with slight variation due to the randomized parameters. We can be confident that the observed difference between these two conditions is caused by users detecting the sudden gain change.

In pairwise analysis, the difference between thresholds was significant only for inexperienced users. Though the same trend can be observed for experienced users, we are not able to demonstrate significance with these results. This is likely due to smaller stimulus values overall among experienced users making the trend harder to observe in this group. A larger study would be useful to confirm the trend.

We found no significant difference between *start-max* and any other condition in our main effect analysis. We did, however, find a significant difference ($p < 0.05$)

between *start-max* and *slow-increase* in the inexperienced group only. Comparison between *start-max* and the other conditions is difficult as while the maximum gain level reached during a turn is the same, the total gain applied is greater. Gain is introduced immediately in this condition, so it could be possible that users are detecting the difference between the gain used in previous trials or during the 2AFC question rather than a mismatch between visual and vestibular cues, particularly as it was the inexperienced group who experienced higher levels of gain. The findings for *start-max* should be considered inconclusive. It would be an extremely interesting finding if, as is suggested by results for the *experienced* group, at low levels of gain participants are primarily sensitive to rate of gain change and not absolute gain level. This suggests a promising avenue for further work.

Our results also indicate that level of virtual reality experience has a significant effect on a user's ability to identify gain. The more experienced participants in our study were able to determine whether gain was present with much greater accuracy leading to considerably lower thresholds in all gain conditions as seen in Figure 4.5. We also found a change in the experienced users' perception of the gain conditions. Unlike the inexperienced group a statistically significant difference between gain conditions was not found ($p = 0.064$) and effect size was smaller (partial $\eta^2 = 0.241$ as opposed to partial $\eta^2 = 0.470$). This could be a factor of the lower levels of gain the experienced group encountered.

Thresholds were higher than those found in similar psychophysical studies. This is particularly true for the *virtual-larger* group. There is not a large body of data from other studies suitable for comparison with the *slow-increase* or *delay-max* conditions, but the *start-max* condition is the same approach used by Steinicke [40] and Bruder [52], so we would expect similar thresholds. As we were looking for differences between conditions the absolute values of the thresholds are less important but worth considering. In our experiment participants only experienced either higher or lower gain, depending on their group, and the gain experienced was entirely within the control of the user through the staircase. It seems possible that participants were adapting to the higher levels of gain they reached; by contrast,

participants in [40, 52] were exposed to a range of gains centered around 1 causing each participant to be exposed to virtual rotations both smaller and larger than their physical turns. The virtual environment used could also be a contributing factor. As vision is partially dependent on landmark recognition [97] and our desert environment is relatively simple (see Figure 4.1), it may have been harder for participants to understand how far they had turned than in the city environments used in [40, 52].

4.5 Summary

In this chapter we described a study that investigated the effect of rate of gain change as measured by a within-subjects psychophysical experiment with adaptive 2-up 1-down staircases. We compared sudden gain change, slow gain change and constant gain. Our results indicate that rate of gain change has a significant effect on user experience of motion gain. In particular, slow gain change appears significantly more subtle than sudden gain change. We also found that more experienced users of virtual reality were able to identify levels of gain significantly more accurately than less experienced users. These findings should be helpful for future redirected walking implementations by confirming that gain smoothing is an important factor in keeping redirected walking imperceptible.

Chapter 5

Shared Spaces: Multi-User Environments in Mixed Reality

This chapter details the design of a remote redirection technique intended to allow real walking for remote users sharing virtual spaces in both virtual and augmented reality. The technique was a collaboration with Tuanfeng Wang, also a UCL PhD student, who was responsible for generating the mappings. The chapter also includes the design and results of a user study with 38 participants conducted to prove the concept and validate the technique's performance. The work in this chapter was published in a modified form under the name 'Merging Environments for Shared Spaces in Mixed Reality' in the 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18) [2].

5.1 Introduction

The aim of the approach taken to redirected walking in Chapters 3 and 4 is to allow a user with a small physical tracked space to explore a larger virtual environment while physically walking. The work in this chapter aims to make use of similar real-time modification of movement in a multi-user context where user's physical spaces differ.

Remote mixed reality collaboration is a topic that has received significant research interest. The typical solution for collaboration with a user in augmented reality is that only the local user is immersed. The remote user can observe the

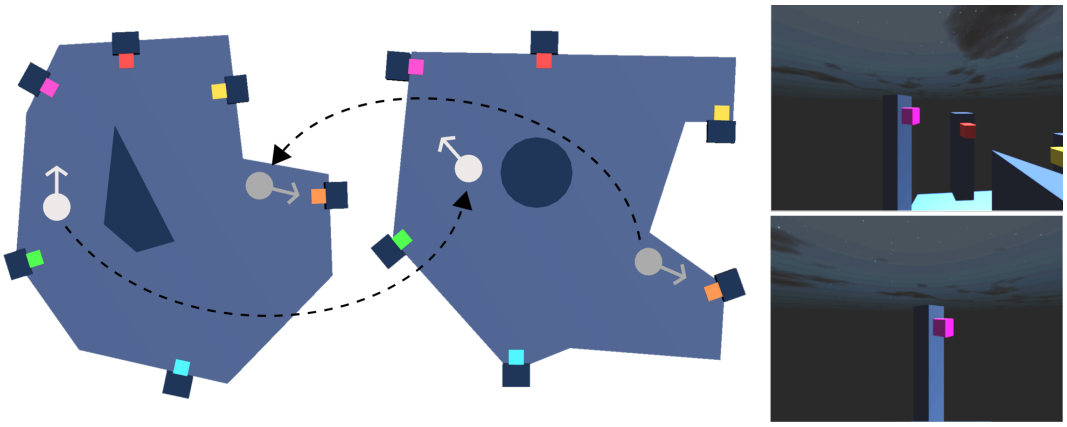


Figure 5.1: Overview of the technique. In order to maximize use of local physical space, the virtual world is constructed and presented differently to two users. Their movements are dynamically mapped into their collaborator's environment so as to create the impression that they are each sharing their own environment with the other. The left two images show floorplans for two rooms, each containing one local user and an avatar of a remote user. The remote user's position and orientation in their own room is used to place the avatar via a pre-generated forward mapping. The right two images show the viewpoint of the user in white in their own room (top) and the viewpoint of their mapped avatar (bottom).

local user's view on a screen or head-mounted display and label elements of the environment [101, 102, 103]. This is useful when the collaborative task involves detailed work on physical objects, such as remote surgery [103]. Embodied remote collaboration has received less attention but has great potential for design and social applications.

In virtual reality a virtual environment may be difficult to navigate with physical walking due to a mismatch with a user's tracked space. One approach in this scenario is to allow locomotion through alternative means. However, real walking has frequently been shown to be preferable for navigation and simulator sickness when compared with more abstract alternatives such as via joystick, wand or gaze-based movement [3, 9, 10, 11] and teleportation [7, 9, 22, 3]. This is covered in significantly more detail in Chapter 2.

As real walking has so many positive characteristics in virtual reality, there is the motivation for techniques which use the limited physical space more intelligently. This allows a real walking interface to cover a much greater distance. Interrante et al. propose *Seven League Boots* [38], an approach which magnifies physical movements. Magnification is applied only in the user's movement direction so as

not to exaggerate head sway. In practice, however, movement direction is hard to estimate, so the effect is disorientating while stationary or performing tasks [104].

Redirected walking takes this approach further, applying gain to translation but also to a user's rotation as they walk [4]. Thresholds have been found beneath which users do not perceive that gain is being applied [105]. Redirected walking has been proven to work well, but unless the user's path is known in advance a great deal of space is required. Steinicke et al. found that for users to believe they are walking in a straight line while really walking in a circular arc, the radius of that arc must be no less than 22 meters [40]. This makes it a poor fit for smaller tracked spaces or awkwardly shaped virtual environments.

Both redirected walking and *Seven League Boots* work by dynamically disrupting the mapping between the virtual and physical space. The technique described in this chapter is inspired by Sun et al., who propose computing a planar map between the virtual and physical floor plans [49]. By contrast with redirected walking and *Seven League Boots*, this mapping is pre-calculated and static. The mapping goes through an optimisation step to map the virtual space to the available physical space and is surjective but not injective, potentially overlapping the virtual space on the physical many times. At run-time the map is consulted to render the virtual environment in the space of the physical environment.

The static mapping proposed by Sun et al. [49] has the great advantage over redirected walking that the user will never leave the track space. However, unlike redirected walking, mapping can not be limited to perceptual thresholds. We can attempt to minimize distortion during the map generation stage, but as we do not know how fast the user will be moving we can not know how much redirection is being applied. For redirected walking, high gains are noticeable to users and lead to simulator sickness [105]. As we see in Chapter 4, sudden changes of gain are also noticeable, and mappings give us no control over this rate of change. To avoid this problem, we generate two static forward mappings between each pair of spaces (A to B and B to A), and apply the map to user's avatars rather than the users themselves. As a result, users need not deal with any perceptual distortion or

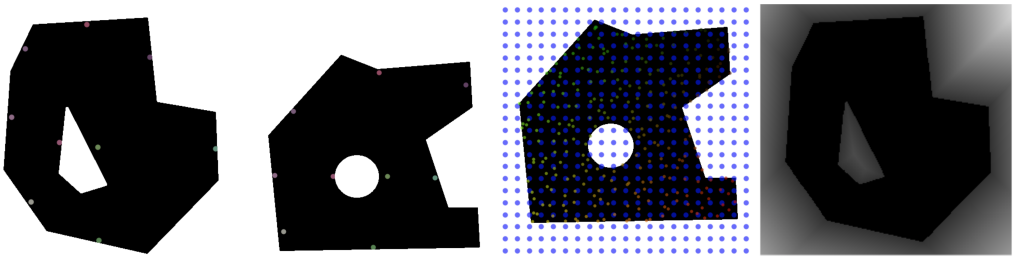
redirection, but still appear to be within the other user's space.

The result is a novel technique for virtual co-location and collaboration which allows each user to operate in their own virtual environment while seamlessly interacting with embodied remote users. The technique operates in real-time with little overhead. Remote users are placed in the local user's environment through pre-generated mappings, allowing movements to be meaningfully transposed between spaces. From each user's perspective their own movements are not manipulated, and as such the technique does not introduce conflicts between the visual and vestibular systems.

This approach is motivated by observing the small but consistent modifications applied during redirected walking. These modifications allow for resulting movement which has a natural verisimilitude even from the perspective of a third party.

The technique has applications in multi-user virtual reality for small virtual spaces but is also readily applicable to augmented reality (AR). In AR the virtual environment is the user's physical environment with virtual additions. A significant part of the appeal of AR is that a user is not separated from the physical world, and can continue to interact with that environment. However, this poses a problem for collaboration, as again we have a different virtual environment per user that we must somehow combine. The typical solution is that a person in AR can only be joined by others who observe the AR user's view on a screen or head-mounted display [101, 102, 103]. By contrast, the technique described in this chapter is designed to allow two or more remote users to collaborate while both are in AR or VR. Remote users are mapped into the each user's local space, preventing clipping through walls or environmental obstacles and creating the illusion of a shared environment. The work in this chapter uses virtual reality for demonstration purposes.

The results of a user study for validation are included in this chapter, along with a discussion of the study design. In the study, 38 participants were divided into pairs and invited to collaborate on a virtual reality puzzle-solving task while in two different virtual rooms. For each participant an avatar representing the remote participant was mapped into the local participant's space. The results of the study



(a) Floorplan of S_A **(b)** Floorplan of S_B **(c)** Sample points **(d)** Distance field

Figure 5.2: Static mapping optimization. Figures (a) and (b) contain the binary floor plans for room S_A and S_B. Correspondence points are coloured for visualization. Figure (c) shows the sampled control points (in blue) and sampled positions used in Equation 5.2 (coloured by 2D position). Figure (d) is the distance field described in Equation 5.4.

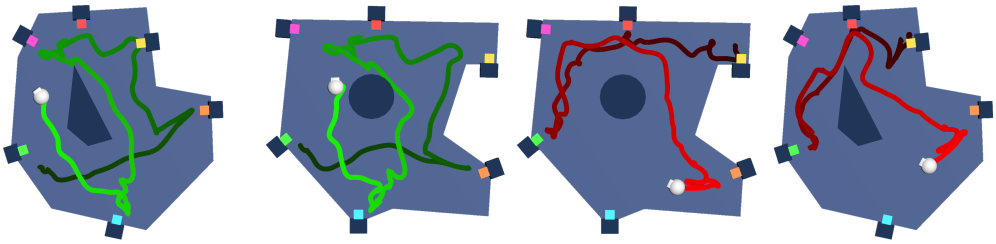
suggest that the system is effective and that good results can be had from immersive collaborative systems even where local environmental representations are actually quite different.

The structure of the chapter is as follows. Section 5.2 covers the theory and method behind generating and applying the spatial mappings used in the technique. Section 5.3 covers the design and results of the user study. Section 5.4 summarizes the chapter and considers future work specific to spatial mapping.

5.2 Generating and Applying Mappings

Our goal is to allow two users (A and B) to interact with each other between two working spaces (e.g. room S_A and room S_B) with different layouts. To do this, we map user A's position in S_A to S_B, reconstruct their rotation and render their avatar in S_B. This process is repeated for user B. We break this into two steps, separating position and rotation into the following sections. We use the following notation for the equations in this section: ${}^{B}x^{A}$ is the position of user B in room S_A. P^A is a function that maps positions to room S_A (from room S_B).

We represent a 2D floorplan of a room in a binary image as shown in Figure 5.2a. Black pixels refer to free space where user can walk through while white pixels refer to obstacles, i.e. wall or furniture in the center of the room. In order to enable user interactions, users must add pre-defined corresponding locations between S_A and S_B, as shown in Figure 5.2a and 5.2b.



(a) A in S_A **(b)** A mapped in S_B **(c)** B in S_B **(d)** B mapped in S_A

Figure 5.3: Demonstration of the effect of mapping on movement paths. Figures (a) and (c) show A and B's movement paths in their own environments. Figures (b) and (d) show the mapped path of their avatars in the remote environment. Brighter lines mean more recent movement. We separate A and B's paths into different images for clarity. In practice S_A would contain B's avatar and A, while S_B would contain A's avatar and B as seen in Figure 5.1.

5.2.1 Positional Mapping

In this section we describe how to establish a forward mapping P^A from a 2D floorplan S_B to another 2D floorplan S_A. We assume the walkable areas in S_A and S_B are flat so ignore height for position mapping. Therefore, for position:

$$ {}^B x^A = P^A({}^B x^B) \tag{5.1} $$

We expect the mapping P^A to (1) map any interior locations of S_B to a interior location of S_A; (2) minimize the distance between corresponding positions; (3) minimize the angle and distance distortion; and (4) ensure local bijective for local smoothness. Inspired by a recent work [49], we represent our mapping P^A in a basis-function form and format our objectives and constrains into an optimization.

Forward mapping We first uniformly sample N center points from S_B in 2D for our basis functions, $x_i (i = 1, ..., N)$, as shown in Figure 5.2c. Consider a 2D position x in S_B, we construct our mapping P^A as:

$$ P^A(x) = \sum_{i=1}^{N} w_i K_i(x) + T(x) \tag{5.2} $$

Where w_i are the directions of basis functions K_i and $T(\cdot)$ is a 2D affine transformation. We chose a Gaussian function for our basis function where $K_i = \exp(\frac{-\|\mathbf{x}-\mathbf{x_i}\|}{2\sigma^2})$.

Objective function First of all, we densely sample a subset of positions $u_i (i = 1,...,M)$ from the free space of S_B (as shown in Figure 5.2c) to reduce our computational cost. Since we want our mapping P^A to be feasible as it will only map a walkable position (black pixel) from S_B to a walkable position in S_A. To efficiently achieve this, we convert this hard constraint to a soft one by casting it into an energy term. To do so, we first build a binary floorplan of S_A, as shown in Figure 5.2d. This is indexed by the function W. Based on this, we then create a distance field D. In detail, the distance field is constructed as follows:

$$W(x) = \begin{cases} 0, & S_A \text{ at } x \text{ is not walkable} \\ 1, & \text{otherwise} \end{cases} \tag{5.3}$$

$$D(x) = \begin{cases} \min_{W(t)=0}(\|x-t\|), & W(x) = 0 \\ 0, & W(x) = 1 \end{cases} \tag{5.4}$$

Therefore, our term for obstacle free mapping is

$$E_{obs} = \sum_{i=1}^{M} D(P^A(u_i)) \tag{5.5}$$

To minimize the distance between two set of corresponding points, $C_i^A \in S_A$ and $C_i^B \in S_B$, where $i = 1,...,P$, we simply formulate this as a correspondence term

$$E_{corr} = \sum_{i=1}^{P} ||C_i^A - P^A(C_i^B)|| \tag{5.6}$$

A locally isometric mapping requires its Jacobians J to satisfy $J^T J = \mathbf{1}$. To minimize the distance and angle distortion after applying $f_{B\to A}$, we convert this constraint into a energy term

$$E_{iso} = ||J^T J - \mathbf{1}|| \tag{5.7}$$

For local bijective, according to [106], the determinant of Jacobian of the inverse of P^A should be positive everywhere. We keep this as a constraint and our

final optimization is

$$E = E_{obs} + \lambda_{corr} E_{corr} + \lambda_{iso} E_{iso} \tag{5.8}$$

subject to the constraint

$$J_{-1}^{T} J_{-1} > 0 \tag{5.9}$$

When applied in real-time, the effect of the map on user position can be seen in Figure 5.3.

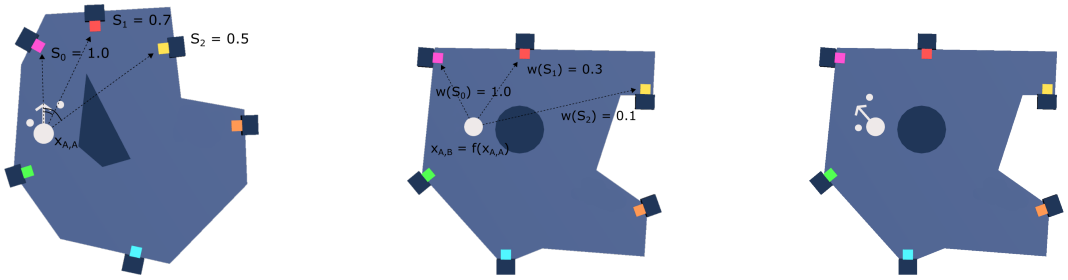
5.2.2 Reconstructing Rotation

After mapping B's position into S_A, B's absolute rotation in S_B is no longer usable. It is likely that the relative direction to objects will not be preserved by the mapping process, so it may not be clear where B is looking. Formally, this section describes how to implement function R^A, where ${}^B\theta^A$ represents the rotation of user B in room S_A and ${}^B\theta^B$ is the rotation of user B in room S_B:

$${}^B\theta^A = R^A({}^B\theta^B) \tag{5.10}$$

As described in Figure 5.4, we can use B's relative direction to objects in S_B to approximate B's rotation in S_A. As we only map in 2D, we can greatly simplify this process by only mapping rotation in 1D, in the transverse plane for the head 'yaw'. We can take 'roll' and 'pitch' directly from the user's local orientation in their own room. For simplicity of notation, rather than an angle, ${}^B\theta^B$ will refer to the 2-component forward-look-vector of user B in room S_B.

We first select M corresponding points across rooms ${}^{u_i}x(i = 1,...,M)$. These points exist in both rooms; we use ${}^{u_i}x^A$ and ${}^{u_i}x^B$ for the correspondence point in A and B respectively. Correspondence points can be the position of any object which is in both rooms. We chose to include the mapping correspondence points due to their significance in the task selected for the study (see Section 5.3.1). We also included the position of the remote user's head to allow for mutual gaze as a



(a) Similarities in S_A **(b)** Weightings in S_B **(c)** Final rotation

Figure 5.4: Calculating rotation and associated object positions. Similarities are calculated as a function of the angle between the user's gaze direction and the direction to each correspondence point and to the other user's face in S_A. After mapping the user's position to S_B, a weighting function is applied to each similarity and the weighted direction vectors are summed to find the new gaze direction. The relative positions of the hands (the small white circles in the Figure) to the user are maintained.

communication tool [107]. We calculate similarities $\mathbf{s}_i (i = 1,...,M) \in [-1,1]$ by taking the dot product of the user's local forward-vector and the direction vector towards each of the corresponding points:

$$\mathbf{s}_i = {}^{B}\theta^{B} \cdot \frac{{}^{u_i}x^{B} - {}^{B}x^{B}}{\left\| {}^{u_i}x^{B} - {}^{B}x^{B} \right\|} \tag{5.11}$$

Where ${}^{u_i}x^{B}$ is the position of point u_i in S_B, s_i is B's similarity to that point. We calculate the reconstructed forward vector for user B in room S_A as an average of the correspondence points in S_A weighted by similarities s:

$${}^{B}\theta^{A} = \sum_{i=1}^{M} w(s_i) \frac{{}^{u_i}x^{A} - {}^{B}x^{A}}{\left\| {}^{u_i}x^{A} - {}^{B}x^{A} \right\|} \tag{5.12}$$

Where w is the weighting function in $[0,1]$. We found it helpful to use the non-linear weighting function $w(s_i) = (s_i \cdot 0.5 + 0.5)^7$. Note that this equation normalises to within $[0,1]$ before raising to the power, so the odd power of 7 is not a concern. The weighting function helped to give the impression that user B was looking directly at objects in room S_A. The particular degree chosen was based on informal testing. We found that lower degrees led to users appearing to look between objects, while higher degrees caused unrealistically fast head movement.

5.2.3 Associated Object Positions and Rotations

In the previous subsection we have discussed placing a user's head through mapping. However, some applications may require associated objects to move with a user such as their hands, feet, torso or held objects. In our tests for the accompanying study, users were provided with virtual hands and bodies to help with communication [108]. Hand position and orientation was provided by tracking for a hand-held controller, while body position and orientation was generated from head and hand positions and orientations.

We preserve the relative position and orientation of associated objects after mapping using the following approach: Let an associated object linked to user B be B_i, and its position in room S_B be ${}^{B_i}x^B$. Let T be the affine transformation that positions B_i relative to user B, such that:

$$ {}^{B_i}x^B = T({}^Bx^B) \tag{5.13} $$

As we know where B_i is relative to B in S_B, we can then place B_i there relative to B in S_A:

$$ {}^{B_i}x^A = T({}^Bx^A) = T(P^A({}^Bx^B)) \tag{5.14} $$

Rotations were directly copied across from the origin room using the same process:

$$ {}^{B_i}\theta^B = T({}^B\theta^B) \tag{5.15} $$

$$ {}^{B_i}\theta^A = T({}^B\theta^A) = T(R^A({}^B\theta^B)) \tag{5.16} $$

We found that mapping objects individually did not produce good results as the mapping interfered with physical movements (e.g., gestures and locomotion) and made them appear unnatural. Preserving relative positions and rotations, as with the above approach, helps to maintain those gestures across rooms. It is, however, a

source of small visual artefacts. In our study, these were noticeable when users were directly interacting with objects or other users. As mapping is not applied directly, the remote user's hands appeared to float some distance away from the objects they were interacting with. These errors are mapping and environment dependent. In our case, they were typically in the region of 10cm, and were noticed by 2 participants out of 38 in our user study (see Section 5.3.7).

5.3 Implementation and User Study

A user study was conducted to prove the concept and to determine if users would notice discrepancies caused by the technique. There are two potential issues which we expected to arise. First the actions of one user may not appear plausible in the other user's space. In particular, users may appear to move too quickly or too slowly and move in curves rather than straight lines. Secondly, features of interest will be positioned differently across the spaces. Users might appear to not be looking or standing where they think they are looking or standing, and appear to be not quite touching or holding something they should be. In this section we aim to discover whether these issues arise in practice.

5.3.1 Study Design

Participants were invited to work together in pairs to complete a short puzzle in virtual reality. The time limit for the task was 5 minutes. Each participant was placed in their own virtual environment, which was designed to fit within the track space used for the study. The virtual environments for the two participants had a different floor plan and different points of interest (see Figure 5.2). Mappings were generated before the study as described in Section 5.2.1.

An avatar representing each participant's head, hands and torso was created. Participants were able to see the full avatar of the remote user, but not their own head or body. The position of the remote participant was provided over the network and the remote participant's avatar placed in the local user's space as described in Section 5.2.1. Orientation of the remote participant was then reconstructed as described in Section 5.2.2. Participants were not informed mapping was taking

place, or that the virtual environments were different.

Each room was surrounded with a set of buttons on small pillars. Each button was a different colour. Either participant could push a button by moving their hand towards it. Once pushed, a button would be depressed for approximately 2 seconds before returning to its default position. Buttons were paired secretly before the experiment begun. If two paired buttons were pushed within 0.2 seconds of each other, a sound would play and the button pair would lock, not returning to their default position. If all buttons were locked the sequence in which they were pressed would be recorded, a chime would play and all buttons would return to neutral. For each new sequence, a new chime would play, up to the maximum of 6.

The design of the task was intended to (1) create a cognitive load, (2) encourage participants to move around and (3) draw attention to the movement of the remote participant. The objective was that the task represent a challenging but realistic scenario for the technique, where users are collaborating on a task with a significant spatial component.

5.3.2 Outcome Measures

Participants were asked to complete questionnaires before and after the study. The pre-questionnaire contained questions were on gender, age, occupation and how many times the participant had used virtual reality. We were interested to examine whether a greater familiarity with virtual reality might make the technique more noticeable to users so chose to investigate this as a between-subjects variable.

The first section of the post-questionnaire combined questions on co-presence [109] and embodiment [110] with the SUS presence questionnaire [111]. As distortion was only applied to user's remote avatars and not to the users themselves, we did not require participants to answer questions on simulator sickness. The second section contained a set of questions specifically designed to assess whether the participant had noticed disruptive effects of the mapping. Participants were asked about a set of phenomena that did not occur during the study, with a statement describing "real" phenomena embedded amongst them. This is similar to the approach taken by Peck et al. [112] and Suma et al. [113].

The embedded question section of the questionnaire was preceded with the statement "This section is on unnatural phenomena you may have noticed in the virtual environment. Please rate the following statements. Note that these phenomena may or may not have happened.". Participants were asked to rate each of the statements on a scale of 1 to 7, where 1 meant "did not notice or did not happen" and 7 meant "very obvious". One of the statements (below in italics) refers to a side effect of the technique that participants may have noticed where mapping was poor. The rest are "decoy" statements. The phenomena described in decoy statements were not present in the virtual environment. In the order presented to the participant, the statements were:

- The floor seemed to change colour
- The floor seemed to move beneath me
- The other person seemed to change shape
- *The other person seemed not to know where things were*
- I felt as if the world was changing shape
- I felt as if the world was rotated around me
- Objects in the world seemed to move without being touched
- Objects in the world seemed to change colour

Finally, participants were asked a series of open questions about the virtual experience. We were primarily interested in whether participants had found that the technique negatively affected their experience. To this end, participants were asked "what aspects of the virtual experience did you find disruptive or unpleasant?" and "what aspects of the virtual experience detracted from it as a whole?". We also included an open-ended question on unusual phenomena to see if participants had observed any artefacts of the technique in a way that had not been detected by the embedded statement.

5.3.3 Study Setup

Participants were equipped with the HTC Vive head-mounted display and hand controllers. Both provide tracking for position and orientation. The display has a resolution of 1080 x 1200 per eye, a refresh rate of 90Hz, and a vertical field of view of 110 degrees. The HMDs were set to the approximate population mean interpupillary distance of 63mm [100]. Participants wore noise-canceling headphones. A voice-over-IP application was used with the built-in microphones in the Vive and set up so participants were able to converse naturally without pressing any buttons.

The room was divided into two 3.5m x 3.5m track spaces, with a 1m gap between track spaces to prevent collisions. The Vive uses external base stations for tracking, so a curtain was placed between the two track spaces to prevent interference. The virtual environments were developed in Unity3D and used the HLAPI for networking and the SteamVR plugin for VR. The systems were connected via ethernet over a local area network. Each Vive was powered by an NVIDIA GTX 1080 and maintained the maximum refresh rate of 90 frames per second throughout. Mappings were generated between the environments on an Intel Core i7 CPU with 8GB memory and took approximately an hour. The maps and environments used for the study are available online [114].

5.3.4 Study Protocol

The study took approximately 30 minutes per pair of participants. Participants were invited in teams of 2. On arrival, participants were given an information sheet describing the study. After an opportunity to ask questions, the experimenter gave each participant a consent form to sign. After consent was obtained the experimenter asked participants to individually complete a short pre-questionnaire (see Section 5.3.2).

The experimenter showed participants to their track space and provided each with a head-mounted display, a pair of controllers and a pair of noise-canceling headphones. The experimenter loaded a demo environment to help participants familiarize themselves with the devices and with a real-walking movement interface. Participants were shown how to identify when they were approaching the edge of

the track space. The experimenter then connected the participants over voice chat and tested audio levels.

When both participants were comfortable the experimenter loaded the study environment. The experimenter connected the two sessions, spawning the rooms and avatars for both participants. The experimenter started a timer and instructed participants that their goal was to "hear 6 chimes". After participants completed the task or 5 minutes elapsed, the experimenter ended the task by bringing participants back to the demo environment and instructing them to remove their headsets.

After participants removed their headsets they were asked to fill out the post-questionnaire (see Section 5.3.1). Finally, participants were debriefed and given the opportunity to ask questions and give feedback.

5.3.5 Participants

38 participants (23 female, 15 male) were recruited. The mean age was 26.02 with a standard deviation of 6.13. Participants were recruited from graduate study mailing lists and external advertisements and paid a small amount to cover travel expenses. Participants were required to be between the ages 18 and 65 and to be able to walk unassisted. This study was approved by the University College London Research Ethics Committee.

5.3.6 Results

Figure 5.5 shows the mean responses to the decoy questions and the embedded question across all participants. Means across all questions were contained within a relatively small range, from lowest (M = 1.66) to highest (M = 2.26), indicating no statement particularly stood out. The highest rated statement was "I felt as if the world was rotated around me" (M = 2.26, SD = 1.46). The lowest was the embedded statement, "The other person seemed not to know where things were" (M = 1.66, SD = 1.15).

As per the methodology used for the original SUS, we invite answers to questions on a 7 point scale. The score for each participant is the number of responses in the 5-7 range, which we then divided by the number of questions. We asked

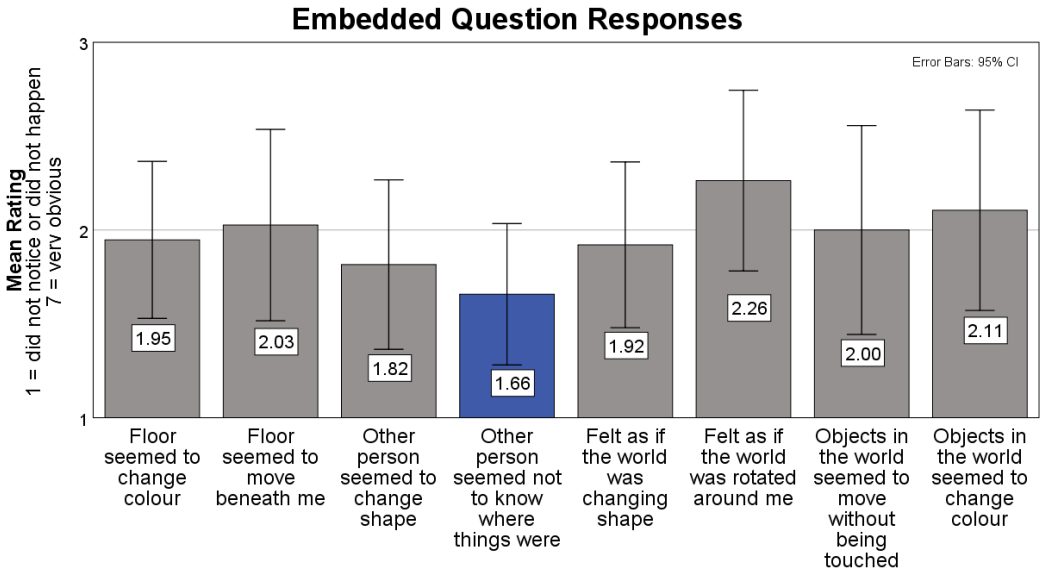


Figure 5.5: Mean participant responses to decoy questions (in grey) and the embedded question (in blue). Note that participants were asked to rate phenomena on a scale of 1 to 7, which has been capped to 3 here for clarity. The rating for the embedded question is particularly low, indicating that users believed the remote user knew where objects were in the local room.

questions on presence (M = .61, SD = .27), co-presence (M = .76, SD = .38) and embodiment (M = .65, SD = .30). Of the 38 participants, 7 scored the maximum on presence, 26 scored the maximum on co-presence and 8 scored the maximum on embodiment.

Participants were broken into 3 groups based on the self-reported number of times each had used virtual reality ("never" with 14 participants, "1-2 times" with 16 participants, or "3+ times" with 8 participants). A one way univariate ANOVA was conducted to test for the effect of previous virtual reality experience on ratings for the decoy questions and embedded question. No significant effect was found for the embedded question on experienced users (M = 1.75, SD = 1.04) over occasional users (M = 1.5, SD = .89) or those who had never used VR (M = 1.79, SD = 1.48), $F(2,36) = .254$, $p = .78$. No significant effect was found for VR experience on any of the decoy questions. No significant effect was found for VR experience on presence ($p = .07$) or co-presence ($p = .505$).

5.3.7 Discussion

The results appear promising as very few participants appeared to be aware of artefacts caused by the mapping. In response to the open ended questions 1 participant described an incident where "my partner was once floating above the obstacle in the middle". This may have been caused by an area in the map which was optimized poorly. Possibly, though, it was caused by the remote participant walking through an obstacle in their own area without realizing, as there was no collision to prevent this in the study. Of the 38 participants 2 identified issues that were likely caused by the mapping. The first noted that "when the other player pressed a button, it looked like they were missing on my screen", while the second participant noticed their partner appeared to miss "when we tried to high-five".

It makes sense that participants would notice the latter two incidents as hand placement is a particularly difficult task for the technique. As described in Section 5.2.3, we preserve the position and orientation of hands across rooms to allow for gesture. However, this means the static map is not directly applied, so even at points where the two mappings agree (the pre-selected correspondence points, e.g., the buttons in the study) hand positions will appear to not quite match. If we were to solve this problem, virtual high-fiving still poses a challenge. The maps are static so hands cannot be included as correspondence points. This means the maps do not have the information necessary to make hands coincide. These distortions are likely to remain small even for very different rooms as they are limited by the distance between the user and their hands.

One possible reason the majority of participants were unaware of hand placement artefacts was the nature of the study task. The design did call for participants to be aware of remote user position, but did not require any attention to be paid to the remote user's hands. The only situation in which this was required was social interaction, beyond the bounds of the task.

Co-presence scores were particularly high when compared with other metrics. Though we did not specifically ask for thoughts on co-presence in the open ended questions, responses to our final question "What are your general thoughts on the

virtual experience?" received a number of positive comments such as "I enjoy the feeling that someone is together with me" and "it was great that I could do it with someone else".

The embedded statement was rated lower than all the decoy statements, which suggests participants were confident the remote user knew where objects were in their local rooms. We were surprised to find that in the results we collected VR experience had no significant effect on responses to the embedded statement. Our assumption was that new VR users would dismiss the artefacts as a limitation of VR technology. Though the task required co-operation and discussion, participants were not strictly required to look at one another, and this may have had a more significant impact on whether or not distortions were recognized.

5.4 Summary

In this chapter we described using mapping for space sharing in mixed reality. This is a novel technique for virtual co-location which allows each user to experience their own environment while appearing to interact with other users in their own spaces. The technique has applications in augmented and virtual reality. Our user study indicates the concept works in practice. The participant responses to the embedded question indicate they believed the remote user was present in their local environment, and only 2 of 38 participants noticing mapping artefacts.

The main source of artefacts appeared to be associated objects (see Section 5.2.3). This is a challenging problem to solve that should be considered in future implementations. One possible fix would be apply a small magnetism to points of interest (including other user's hands) in the remote room based on their distance in the local room. Remote associated objects could then be moved closer to local points of interest, making mapping discrepancies less noticeable. With the correct weighting, this could occur only when the user's local hand is close, and left unmodified most of the time, hopefully not interfering with mapping or gestures.

Chapter 6

Conclusion

The research in this thesis explored applying redirection to small physical spaces using novel redirection algorithms and psychophysical studies. Here we summarize the primary contributions from the thesis and consider potential future work.

Monte-Carlo Redirected Walking (MCRDW) is a method for selecting redirection strategies through simulated walks (see Chapter 3). By selecting good gain values, MCRDW can guide users around physical paths which significantly reduce the likelihood of the user encountering their physical boundary. We include an evaluation of MCRDW under different conditions and metrics, and provide a comparison with existing gain selection methods. In our simulations, MCRDW significantly reduced boundary collisions when compared with the next best gain selection technique. Additionally, MCRDW did not increase total rotation or position gain over existing techniques, and in many cases scored better than existing methods on these metrics. MCRDW performance suffered in environments which had fewer obstacles, where our simple path predictor was unable to provide good guesses on future directions. As a result the greatest performance improvement was observed in the most obstacle-rich environments. Regardless, MCRDW did still outperform existing techniques in all environments, even those which were more open.

There are several promising avenues for further work. A follow up study with users rather than simulations would further validate the technique. Likewise, testing the technique in different kinds of virtual environments would help to better understand when it is a good fit for a specific use case. Finally, the path prediction used

in the sample MCRDW implementation is very simple. There have been promising developments in path prediction based on user movement and environment understanding [76, 79]. However, likely the next step in path prediction will be machine learning based and is yet to arrive. A more advanced path predictor could be simply integrated with the MCRDW approach, leading to a significant performance improvement overall.

As with most gain selection algorithms, the MCRDW approach changes gain direction and magnitude frequently. The intention of our included **user study on sensitivity to rate-of-change of rotation gain** (see Chapter 4) is to determine the effect these changes have on users. The results suggest that gain changes can be disruptive when sudden, but that slow changes appear significantly more subtle. This result is integrated into our sample MCRDW implementation, and is hopefully helpful for future RDW implementations by confirming that a smoothing term is beneficial for RDW performance and user comfort.

A further interesting finding from the study was that inexperienced users were less able to identify when gain was applied. Part of this may be due to an unexpected effect of the adaptive staircase procedure. This approach slowly increases or decreases gain based on participant response. This is unlike previous studies on user response to redirection [40, 52], which by design test fixed levels centered around a gain of 1. This gradual increase may have given participants a chance to become accustomed to very high gain. In psychophysics this phenomenon, known as habituation error [115], is a common occurrence for many kinds of stimulus. However, the finding that users can become habituated to high levels of gain is itself interesting, and an interesting direction for future work.

We also introduce a technique for remote real walking in differently shaped obstacle-rich environments. An advantage of this work is that the above concerns do not apply, as no gain is applied to the local user's movements. With this approach, **Shared Spaces**, mappings are generated between each user's local space (see Chapter 5). Each user experiences their own environment and can observe remote users interacting with it as if they were present. However, in reality, remote

users are being positioned in each user's local space using the mappings. Shared Spaces have applications in both virtual and augmented reality. In the accompanying user study, participants appeared to believe remote users were present in the same environment. This is promising and suggest Shared Spaces could be a useful method for virtual co-location.

However, this is a novel technique and there are many potential directions for further work. In our Shared Spaces user study we only observed one pair of environments, and while distortions between these environments went largely unnoticed in the study it remains to be seen how well the technique will work on very different environments. The technique also currently requires manual labelling of correspondence points, which is likely impractical for end-users. Finally, the optimization task for generating static maps is time-consuming. This latter cost could be reduced by using pre-calculated partial maps between track spaces, reducing the final optimization time per room. Positional mapping and rotation reconstruction appeared to work well, but associated objects (and hands in particular) were the cause of many visible mapping artefacts. A more sophisticated approach to conformal mapping, potentially with a dynamic map to increase walkable area, would be a promising avenue for future study.

Bibliography

[1] Ben J Congdon and Anthony Steed. Sensitivity to rate of change in gains applied by redirected walking. In *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19. Association for Computing Machinery, 2019.

[2] Ben J Congdon, Tuanfeng Wang, and Anthony Steed. Merging Environments for Shared Spaces in Mixed Reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, pages 11:1—-11:8, New York, NY, USA, 2018. ACM.

[3] Martin Usoh, Kevin Arthur, Mary Whitton C., Rui Bastos, Anthony Steed, Mel Slater, and Frederick Brooks P. Walking > Walking-in-Place > Flying, in Virtual Environments. *SIGGRAPH '99 Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 359–364, 1999.

[4] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. Redirected Walking. *Proceedings of EUROGRAPHICS*, pages 289–294, 2001.

[5] Dang Yang Lee, Yang Hun Cho, and In Kwan Lee. Real-time optimal planning for redirected walking using deep q-learning. In *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, 2019.

[6] Ryan R. Strauss, Raghuram Ramanujan, Andrew Becker, and Tabitha C. Peck. A Steering Algorithm for Redirected Walking Using Reinforcement

Learning. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[7] Doug a Bowman, David Koller, and Larry F Hodges. Travel in Immersive Virtual Environments: An Evaluation Motion Control Techniques of Viewpoint. *IEEE Virtual Reality Annual Symposium*, pages 45–52, 1997.

[8] Daniel Medeiros, Mauricio Sousa, Alberto Raposo, and Joaquim Jorge. Magic Carpet: Interaction Fidelity for Flying in VR. *IEEE transactions on visualization and computer graphics*, 26(9):2793–2804, sep 2020.

[9] Roy. A. Ruddle and S. Lessels. The benefits of using a walking interface to navigate virtual environments. *ACM Transactions on Computer-Human Interaction*, 16(1):1–18, 2009.

[10] Evan A. Suma, Samantha L. Finkelstein, Seth Clark, Paula Goolkasian, and Larry F. Hodges. Effects of travel technique and gender on a divided attention task in a virtual environment. In *3DUI 2010 - IEEE Symposium on 3D User Interfaces 2010, Proceedings*, pages 27–34, 2010.

[11] Catherine A. Zanbaka, Benjamin C. Lok, Sabarish V. Babu, Amy C. Ulinski, and Larry F. Hodges. Comparison of path visualizations and cognitive measures relative to travel technique in a virtual environment. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):694–705, 2005.

[12] Doug A Bowman, Ernst Kruijff, Joseph J LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[13] Zhixin Yan, Robert W. Lindeman, and Arindam Dey. Let your fingers do the walking: A unified approach for efficient short-, medium-, and long-distance travel in VR. In *2016 IEEE Symposium on 3D User Interfaces, 3DUI 2016 - Proceedings*, pages 27–30, 2016.

[14] Ajoy S. Fernandes and Steven K. Feiner. Combating VR sickness through subtle dynamic field-of-view modification. In *2016 IEEE Symposium on 3D User Interfaces, 3DUI 2016 - Proceedings*, pages 201–210, 2016.

[15] J.J.-W. Lin, H.B.L. Duh, D.E. Parker, H. Abi-Rached, and T.A. Furness. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. *Proceedings IEEE Virtual Reality 2002*, 2002(February 2002):164–171, 2002.

[16] Jiwan Bhandari, Sam Tregillus, and Eelke Folmer. Legomotion: Scalable Walking-Based Virtual Locomotion. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, VRST '17, New York, NY, USA, 2017. Association for Computing Machinery.

[17] Randy Pausch, Tommy Burnette, Dan Brockway, and Michael E. Weiblen. Navigation and locomotion in virtual worlds via flight into hand-held miniatures. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95*, pages 399–400, 1995.

[18] Sebastian Freitag, Dominik Rausch, and Torsten Kuhlen. Reorientation in virtual environments using interactive portals. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 119–122, 2014.

[19] Daniel Cliburn, Stacy Rilea, David Parsons, Prakash Surya, and Jessica Semler. The Effects of Teleportation on Recollection of the Structure of a Virtual World. In *Proceedings of the 15th Joint Virtual Reality Eurographics Conference on Virtual Environments*, JVRC'09, pages 117–120, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[20] Lindsay K. Vass, Milagros S. Copara, Masud Seyal, Kiarash Shahlaie, Sarah Tomaszewski Farias, Peter Y. Shen, and Arne D. Ekstrom. Oscillations Go the Distance: Low-Frequency Human Hippocampal Oscillations Code Spatial Distance in the Absence of Sensory Cues during Teleportation. *Neuron*, 89(6):1180–1186, 2016.

[21] Kasra Moghadam, Colin Banigan, and Eric D Ragan. Scene Transitions and Teleportation in Virtual Reality and the Implications for Spatial Awareness and Sickness. *IEEE transactions on visualization and computer graphics*, 26(6):2273–2287, jun 2020.

[22] Mel Slater, Martin Usoh, and Anthony Steed. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction*, 2(3):201–219, 1995.

[23] James N. Templeman, Patricia S. Denbrook, and Linda E. Sibert. Virtual Locomotion: Walking in Place through Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 8(6):598–617, 1999.

[24] Niels C. Nilsson, Stefania Serafin, Morten H. Laursen, Kasper S. Pedersen, Erik Sikstrom, and Rolf Nordahl. Tapping-in-place: Increasing the naturalness of immersive walking-in-place locomotion through novel gestural input. In *IEEE Symposium on 3D User Interface 2013, 3DUI 2013 - Proceedings*, 2013.

[25] Preston Tunnell Wilson, William Kalescky, Ansel MacLaughlin, and Betsy Williams. VR Locomotion: Walking > Walking in Place > Arm Swinging. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1*, VRCAI '16, pages 243–249, New York, NY, USA, 2016. ACM.

[26] Sara Hanson, Richard A Paris, Haley A Adams, and Bobby Bodenheimer. Improving Walking in Place Methods with Individualization and Deep Networks. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 367–376, 2019.

[27] Gabriel Cirio, Maud Marchal, Tony Regia-Corte, and Anatole Lécuyer. The Magic Barrier Tape: A Novel Metaphor for Infinite Navigation in Virtual Worlds with a Restricted Walking Workspace. In *Proceedings of the 16th*

ACM Symposium on Virtual Reality Software and Technology, VRST '09, pages 155–162, New York, NY, USA, 2009. ACM.

[28] Gabriel Cirio, Peter Vangorp, Emmanuelle Chapoulie, Maud Marchal, Anatole Lecuyer, and George Drettakis. Walking in a Cube: Novel Metaphors for Safely Navigating Large Virtual Environments in Restricted Real Workspaces. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):546–554, apr 2012.

[29] Betsy Williams, Gayathri Narasimham, Bjoern Rump, Timothy P. McNamara, Thomas H. Carr, John Rieser, and Bobby Bodenheimer. Exploring large virtual environments with an HMD when physical space is limited. *Proceedings of the 4th symposium on Applied perception in graphics and visualization - APGV '07*, 1(212):41, 2007.

[30] Adalberto Simeone, Niels Nilsson, André Zenner, Marco Speicher, and Florian Daiber. The Space Bender: Supporting Natural Walking via Overt Manipulation of the Virtual Environment. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 598–606, 2020.

[31] Tabitha C. Peck, Henry Fuchs, and Mary C. Whitton. Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. In *Proceedings - IEEE Virtual Reality*, pages 35–38, 2010.

[32] Timofey Grechkin, Mahdi Azmandian, Mark Bolas, and Evan Suma. Towards context-sensitive reorientation for real walking in virtual reality. *2015 IEEE Virtual Reality (VR)*, pages 185–186, 2015.

[33] Tabitha C. Peck. Redirected free exploration with distractors: a large-scale real-walking locomotion interface. 2010.

[34] Jeremy N. Bailenson, Jim Blascovich, Andrew C. Beall, and Jack M. Loomis. Interpersonal distance in immersive virtual environments. *Personality and Social Psychology Bulletin*, 29(7):819–833, 2003.

[35] Christian T. Neth, Jan L. Souman, David Engel, Uwe Kloos, Heinrich H. Bülthoff, and Betty J. Mohler. Velocity-dependent dynamic curvature gain for redirected walking. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1041–1052, 2012.

[36] Gerd Bruder, Frank Steinicke, and Klaus H. Hinrichs. Arch-explore: A natural user interface for immersive architectural walkthroughs. In *3DUI - IEEE Symposium on 3D User Interfaces 2009 - Proceedings*, pages 75–82, 2009.

[37] Joseph J. LaViola Jr., Daniel Acevedo Feliz, Daniel F. Keefe, and Robert C. Zeleznik. Hands-Free Multi-Scale Navigation in Virtual Environments. *Proceedings of the 2001 symposium on Interactive 3D graphics SI3D 01*, pages 9–15, 2001.

[38] Victoria Interrante, Brian Ries, and Lee Anderson. Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In *IEEE Symposium on 3D User Interfaces 2007 - Proceedings, 3DUI 2007*, pages 167–170, 2007.

[39] Betsy Williams. *Design and evaluation of methods for motor exploration in large virtual environments with head-mounted display technology*. PhD thesis, Vanderbilt University, 2008.

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

[41] D. J. Simons. Current Approaches to Change Blindness. *Visual Cognition*, 7(1-3):1–15, 2000.

[42] S C Hirtle and J Jonides. Evidence of hierarchies in cognitive maps. *Memory & Cognition*, 13(3):208–217, 1985.

[43] J. Kevin O'Regan, Heiner Deubel, James J. Clark, and Ronald a. Rensink. Picture Changes During Blinks: Looking Without Seeing and Seeing Without Looking. *Visual Cognition*, 7(1-3):191–211, 2000.

[44] J. Triesch, B. T. Sullivan, Mary M. Hayhoe, and Dana H. Ballard. Saccade contingent updating in virtual reality. *In Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 95–102, 2002.

[45] Frank Steinicke, Gerd Bruder, Klaus Hinrichs, and Pete Willemsen. Change blindness phenomena for stereoscopic projection systems. In *Proceedings - IEEE Virtual Reality*, pages 187–194, 2010.

[46] Patrick Foo, William H. Warren, Andrew Duchon, and Michael J. Tarr. Do humans integrate routes into a cognitive map? Map- versus landmark-based navigation of novel shortcuts. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(2):195–215, 2005.

[47] Evan A. Suma, Zachary Lipps, Samantha Finkelstein, David M. Krum, and Mark Bolas. Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):555–564, 2012.

[48] Khrystyna Vasylevska, Hannes Kaufmann, Mark Bolas, and Evan A. Suma. Flexible spaces: A virtual step outside of reality. *Proceedings - IEEE Virtual Reality*, pages 109–110, 2013.

[49] Qi Sun, Li-Yi Wei, and Arie Kaufman. Mapping virtual and physical reality. *ACM Transactions on Graphics*, 2016.

[50] Zhi-Chao Dong, Xiao-Ming Fu, Chi Zhang, Kang Wu, and Ligang Liu. Smooth assembled mappings for large-scale real walking. *ACM Transactions on Graphics*, 36(6), 2017.

[51] Antong Cao, Lili Wang, Y Liu, and V Popescu. Feature Guided Path Redirection for VR Navigation. *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 137–145, 2020.

[52] Gerd Bruder, Victoria Interrante, Lane Phillips, and Frank Steinicke. Redirecting walking and driving for natural navigation in immersive virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):538–545, 2012.

[53] Eike Langbehn, Gerd Bruder, and Frank Steinicke. Subliminal Reorientation and Repositioning in Virtual Reality During Eye Blinks. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, SUI '16, page 213, New York, NY, USA, 2016. ACM.

[54] Ruimin Zhang and Scott A Kuhl. Human Sensitivity to Dynamic Rotation Gains in Head-mounted Displays. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '13, pages 71–74, New York, NY, USA, 2013. ACM.

[55] Sharif Razzaque. *Redirected Walking*. PhD thesis, Chapel Hill, NC, USA, 2005.

[56] Eric Hodgson and Eric Bachmann. Comparing four approaches to generalized redirected walking: simulation and live user data. *IEEE transactions on visualization and computer graphics*, 19(4):634–43, 2013.

[57] Stefania Serafin, Niels C. Nilsson, Erik Sikstrom, Amalia De Goetzen, and Rolf Nordahl. Estimation of detection thresholds for acoustic based redirected walking techniques. *Proceedings - IEEE Virtual Reality*, (April 2016):161–162, 2013.

[58] Niels Christian Nilsson, Evan Suma, Rolf Nordahl, Stefania Serafin, and Niels Christian Nilsson. Estimation of Detection Thresholds for Audiovisual Rotation Gains. (January):1–3, 2016.

[59] Benjamin Bolte and Markus Lappe. Subliminal Reorientation and Repositioning in Immersive Virtual Environments using Saccadic Suppression. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):545–552, 2015.

[60] Anh Nguyen and Andreas Kunz. Discrete Scene Rotation during Blinks and Its Effect on Redirected Walking Algorithms. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, New York, NY, USA, 2018. Association for Computing Machinery.

[61] Mahdi Azmandian, Rhys Yahata, Mark Bolas, and Evan Suma. An enhanced steering algorithm for redirected walking in virtual environments. *Proceedings - IEEE Virtual Reality*, pages 65–66, 2014.

[62] Mahdi Azmandian, Mark Bolas, and Evan Suma. Countering User Deviation During Redirected Walking. *Sap 2014*, page 129, 2014.

[63] Eric Hodgson, Eric Bachmann, and David Waller. Redirected Walking to Explore Virtual Environments: Assessing the Potential for Spatial Interference. *ACM Trans. Appl. Percept.*, 8(4):22:1—-22:22, dec 2008.

[64] Frank Steinicke, Gerd Bruder, Timo Ropinski, and Klaus Hinrichs. Moving Towards Generally Applicable Redirected Walking. *Proceedings of the 10th Virtual Reality International Conference ({VRIC} 2008)*, pages 15–24, 2008.

[65] Tom Field, Sandy Bay, and Peter Vamplew. Generalised Algorithms for Redirected Walking in Virtual Environments. *AISAT2004 International Conference on Artificial Intelligence in Science and Technology*, 65(11):1357–1366, 2004.

[66] Eric Hodgson, Eric Bachmann, and Tyler Thrash. Performance of redirected walking algorithms in a constrained virtual world. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):579–587, 2014.

[67] Ruimin Zhang and Scott A. Kuhl. Flexible and general redirected walking for head-mounted displays. In *Proceedings - IEEE Virtual Reality*, 2013.

[68] Mahdi Azmandian, Timofey Grechkin, Mark Bolas, and Evan Suma. Physical space requirements for redirected walking: How size and shape affect performance. The Eurographics Association, 2015.

[69] Justin Messinger, Eric Hodgson, and Eric R Bachmann. Effects of Tracking Area Shape and Size on Artificial Potential Field Redirected Walking. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 72–80, 2019.

[70] Jerald Thomas and Evan Suma Rosenberg. A General Reactive Algorithm for Redirected Walking Using Artificial Potential Functions. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 56–62, 2019.

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

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

[73] D Engel, C Curio, L Tcheang, B. J. Mohler, H Bülthoff, and H. A psychophysically calibrated controller for navigating through large environments in a limited free-walking space. *ACM Symposium on Virtual Reality Software and Technology*, pages 157–164, 2008.

[74] M. Hollands, A. Patla, and J. Vickers. "Look where you're going!": Gaze behaviour associated with maintaining and changing the direction of locomotion. *Experimental Brain Research*, 143(2):221–230, 2002.

[75] Thomas Nescher and Andreas Kunz. Using head tracking data for robust short term path prediction of human locomotion. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7848, 2013.

[76] Markus Zank and Andreas Kunz. Eye tracking for locomotion prediction in redirected walking. In *2016 IEEE Symposium on 3D User Interfaces, 3DUI 2016 - Proceedings*, pages 49–58, 2016.

[77] Jianbo Su. Motion Compression for Telepresence Locomotion. *Presence: Teleoper. Virtual Environ.*, 16(4):385–398, aug 2007.

[78] Halim Hicheur, Quang-Cuong Pham, Gustavo Arechavaleta, Jean-Paul Laumond, and Alain Berthoz. The formation of trajectories during goal-oriented locomotion in humans. I. A stereotyped behaviour. *European Journal of Neuroscience*, 26(8):2376–2390, 2007.

[79] Markus Zank and Andreas Kunz. Using Locomotion Models for Estimating Walking Targets in Immersive Virtual Environments. In *Proceedings - 2015 International Conference on Cyberworlds, CW 2015*, pages 229–236, 2016.

[80] Gabriel Cirio, Anne Hélène Olivier, Maud Marchal, and Julien Pettré. Kinematic evaluation of virtual walking trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 19(4), 2013.

[81] Gustavo Arechavaleta, Jean Paul Laumond, Halim Hicheur, and Alain Berthoz. Optimizing principles underlying the shape of trajectories in goal oriented locomotion for humans. In *Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS*, pages 131–136, 2006.

[82] Brett R Fajen and William H Warren. Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of experimental psychology. Human perception and performance*, 29(2):343–362, 2003.

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

[84] Courtney Hutton and Evan Suma. A realistic walking model for enhancing redirection in virtual reality. In *Proceedings - IEEE Virtual Reality*, volume 2016-July, pages 183–184, 2016.

[85] R. Oliva and N. Pelechano. NEOGEN: Near optimal generator of navigation meshes for 3D multi-layered environments. *Computers and Graphics (Pergamon)*, 37(5):403–412, 2013.

[86] Mahdi Azmandian, Timofey Grechkin, Mark Bolas, and Evan Suma. Automated path prediction for redirected walking using navigation meshes. In *2016 IEEE Symposium on 3D User Interfaces, 3DUI 2016 - Proceedings*, pages 63–66, 2016.

[87] Anh Nguyen and Andreas Kunz. Discrete Scene Rotation during Blinks and Its Effect on Redirected Walking Algorithms. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, New York, NY, USA, 2018. Association for Computing Machinery.

[88] Gerd Bruder, Paul Lubas, and Frank Steinicke. Cognitive Resource Demands of Redirected Walking. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):539–544, 2015.

[89] Evan A. Suma, David M. Krum, Samantha Finkelstein, and Mark Bolas. Effects of redirection on spatial orientation in real and virtual environments. In *3DUI 2011 - IEEE Symposium on 3D User Interfaces 2011, Proceedings*, pages 35–38, 2011.

[90] Marcus Nyström and Kenneth Holmqvist. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42(1):188–204, 2010.

[91] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3D graphics. *ACM Transactions on Graphics*, 31(6):1, 2012.

[92] Geoffrey Keppel, William H Saufley Jr., and Howard Tokunaga. *Introduction to design and analysis: A student's handbook, 2nd ed.* W H Freeman/Times Books/ Henry Holt & Co, 1992.

[93] Tabitha C. Peck, Henry Fuchs, and Mary C. Whitton. An evaluation of navigational ability comparing Redirected Free Exploration with Distractors to Walking-in-Place and joystick locomotio interfaces. In *Proceedings - IEEE Virtual Reality*, pages 55–62, 2011.

[94] Tom N. Cornsweet. The staircase-method in psychophysics. *The American journal of psychology*, 1962.

[95] Walter H Ehrenstein and Addie Ehrenstein. *Psychophysical Methods*, pages 1211–1241. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[96] Marjorie R. Leek. Adaptive procedures in psychophysical research. *Perception & Psychophysics*, 63(8):1279–1292, nov 2001.

[97] William H. Warren, Bruce Kay, Wendy D. Zosh, Andrew P. Duchon, and Stephanie Sahuc. Optic flow is used to control human walking. *Nature Neuroscience*, 4:213–216, 2001.

[98] Miguel A. García-Pérez. Forced-choice staircases with fixed step sizes: Asymptotic and small-sample properties. *Vision Research*, 1998.

[99] David M. Green, Virginia M. Richards, and Timothy Forrest. Stimulus step size and heterogeneous stimulus conditions in adaptive psychophysics. *Acoustical Society of America Journal*, 86:629–636, 1989.

[100] Neil A. Dodgson. Variation and extrema of human interpupillary distance. In *Proc. SPIE 5291*, pages 36–46, 2004.

[101] Thammathip Piumsomboon, Arindam Day, Barrett Ens, Youngho Lee, Gun Lee, and Mark Billinghurst. Exploring enhancements for remote mixed reality collaboration. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, SA '17, pages 16:1–16:5, New York, NY, USA, 2017. ACM.

[102] M. L. Chenechal, T. Duval, V. Gouranton, J. Royan, and B. Arnaldi. Vishnu: virtual immersive support for helping users an interaction paradigm for collaborative remote guiding in mixed reality. In *2016 IEEE Third VR International Workshop on Collaborative Virtual Environments (3DCVE)*, pages 9–12, March 2016.

[103] Matthew Christopher Davis, Dang D. Can, Jonathan Pindrik, Brandon G. Rocque, and James M. Johnston. Virtual interactive presence in global surgical education: International collaboration through augmented reality. *World Neurosurgery*, 86:103 – 111, 2016.

[104] Betsy Williams. *Design and evaluation of methods for motor exploration in large virtual environments with head-mounted display technology*. PhD thesis, Vanderbilt University, 2008.

[105] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, Markus Lappe, and Harald Frenz. Analyses of Human Sensitivity to Redirected Walking. *Vrst*, 146(3):149–156, 2008.

[106] Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. Locally injective mappings. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, pages 125–135. Eurographics Association, 2013.

[107] Michael Argyle and Mark Cook. *Gaze and mutual gaze.* Cambridge U Press, Oxford, England, 1976.

[108] Trevor J. Dodds, Betty J. Mohler, and Heinrich H. Bülthoff. Talk to the virtual hands: Self-animated avatars improve communication in head-mounted display virtual environments. *PLoS ONE*, 2011.

[109] Cagatay Basdogan, Chih-Hao Ho, Mandayam A. Srinivasan, and Mel Slater. An experimental study on the role of touch in shared virtual environments. *ACM Trans. Comput.-Hum. Interact.*, 7(4):443–460, December 2000.

[110] Elena Kokkinara, Konstantina Kilteni, Kristopher J. Blom, and Mel Slater. First person perspective of seated participants over a walking virtual body leads to illusory agency over the walking. 6:28879, 07 2016.

[111] Mel Slater, Martin Usoh, and Anthony Steed. Depth of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 1994.

[112] Tabitha C. Peck, Henry Fuchs, and Mary C. Whitton. Evaluation of reorientation techniques and distractors for walking in large virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 2009.

[113] Evan A. Suma, Seth Clark, David Krum, Samantha Finkelstein, Mark Bolas, and Zachary Warte. Leveraging change blindness for redirection in virtual environments. In *Proceedings - IEEE Virtual Reality*, pages 159–166, 2011.

[114] Ben Congdon. Accompanying Data for "Merging Environments For Shared Spaces in Virtual Reality", DOI:[10.6084/m9.figshare.7154702.v1]. 2018.

[115] Mark E Bouton. *Learning and behavior: A contemporary synthesis.* Sinauer Associates, 2007.