Published in final edited form as:
IEEE Trans Vis Comput Graph. 2012 July ; 18(7): 1053–1067. doi:10.1109/TVCG.2011.289.

The Design and Evaluation of a Large-Scale Real-Walking Locomotion Interface

Tabitha C. Peck [Member, IEEE],
EVENT Lab at the University of Barcelona, Barcelona, Spain, 08032. tpeck@cs.unc.edu

Henry Fuchs [Member, IEEE], and
Department of computer science at The University of North Carolina at Chapel Hill

Mary C. Whitton [Member, IEEE]
Department of computer science at The University of North Carolina at Chapel Hill

Abstract

Redirected Free Exploration with Distractors (RFED) is a large-scale real-walking locomotion interface developed to enable people to walk freely in virtual environments that are larger than the tracked space in their facility. This paper describes the RFED system in detail and reports on a user study that evaluated RFED by comparing it to walking-in-place and joystick interfaces. The RFED system is composed of two major components, *redirection* and *distractors*. This paper discusses design challenges, implementation details, and lessons learned during the development of two working RFED systems. The evaluation study examined the effect of the locomotion interface on users' cognitive performance on navigation and wayfinding measures. The results suggest that participants using RFED were significantly better at navigating and wayfinding through virtual mazes than participants using walking-in-place and joystick interfaces. Participants traveled shorter distances, made fewer wrong turns, pointed to hidden targets more accurately and more quickly, and were able to place and label targets on maps more accurately, and more accurately estimate the virtual environment size.

Keywords

Virtual Reality; Locomotion; Navigation; Redirection; Distractors; Wayfinding

1 Introduction

Navigation is the combination of wayfinding and locomotion and as such is both cognitive and physical. Wayfinding is defined as the building and maintaining a cognitive map and is used to determine how to get from one location to another, while locomotion is moving—physically or virtually—between two locations [1]. In the real world people often locomote by walking. For most people walking is simple and natural, and enables people not only to move between locations, but also to develop cognitive maps, or mental representations, of environments.

People navigate every day in the real world without problem, however users navigating VEs often become disoriented and frustrated, and find it challenging to transfer spatial knowledge acquired in the VE to the real world [2]–[5]. Navigation is important for VE applications where spatial understanding of the VE must transfer to the real world, such as exploring virtual cities, training ground troops, or visiting virtual models of houses.

Real-walking locomotion interfaces are believed to enable better user navigation, are more natural, and produce a higher sense of presence than other locomotion interfaces [6], [7]. However, because user motion must be tracked, VEs using a real walking locomotion interface have typically been restricted in size to the area of the tracked space. Enabling users to really walk through a virtual environment (VE) that is larger than the tracked space requires that areas of the VE outside the tracked space be remapped to inside the tracked space by translating, rotating, scaling, skewing, or otherwise altering the VE, thus enabling users to walk to new regions in the VE.

Even with such transformation of a larger-than-tracked-space VE, users may find themselves about to walk out of the tracked space (possibly into a real wall!). When a user nears the edge of the tracked space a reorientation technique (ROT) must be used to prevent the user from leaving the tracked space [8].

Redirected Free Exploration with Distractors (RFED) is a locomotion interface that combines transformation of the VE, specifically based on the redirected walking (RDW) system [9], which uses *redirection*—imperceptibly rotating the VE model around the user—to remap areas of the VE to the tracked space, and *distractors*—visual objects and/or sounds in the VE—as a ROT.

In this paper we further develop RFED, first presented in [10], and discuss our final design decisions as well as attempted designs that failed. RFED was implemented in a head-mounted display (HMD) for walking through the VE, however based on [16], variations of RFED could be implemented in CAVEs. The RFED systems presented in this paper handles the general rectilinear tracked space, however future RFED implementations could handle any boundary shape, or moving obstacles such as real people, for collaborative VEs.

A common concern about redirection and distractors is the potential increase in cognitive load, disorientation, and simulator sickness. We have previously reported a user study that evaluated RFED compared to a real-walking locomotion interface that showed RFED participants performed no worse than real-walking participants based on wayfinding and navigation measures [10]. A second study, first presented in [30] and also described in this paper, shows that users perform significantly better on navigation and wayfinding metrics with RFED than with joystick (JS) and walking-in-place (WIP) locomotion interfaces. RFED participants traveled shorter distances, made fewer wrong turns, pointed to hidden targets more accurately and more quickly, and were able to place and label targets on maps more accurately than both (JS) and (WIP) participants. No significant difference in either presence or simulator sickness was found between RFED, WIP, and JS.

Finally, we discuss research areas that may improve future real-walking locomotion interfaces. We invite other implementers to consult with us on their attempted designs and to open the discussion on how to develop promising locomotion interfaces.

2 Background

Previous research suggests that users navigate best in VEs with locomotion interfaces such as real-walking [11] that provide users with vestibular and proprioceptive feedback. Interfaces that stimulate both of these systems improve navigation performance and are less likely to cause simulator sickness than locomotion interfaces that do not stimulate both systems [11], [12]. VE locomotion interfaces such as walking-in-place, omni-directional treadmills, or bicycles [13], [14] require physical input from the user, however they do not stimulate the vestibular and proprioceptive systems in the same way as really walking. In contrast, RFED users really walk, generating both vestibular and proprioceptive feedback.

Since user motion must be tracked, VEs using real-walking locomotion interfaces have been typically restricted in size to the area of the tracked space. Current interfaces that enable real walking in larger-than-tracked-space VEs include redirected walking (RDW) [9], [15], [16], scaled-translationalgain [17]–[19], seven-league boots [20], motion compression [21], [22], and dynamic curvature gain [23]. Each of these interfaces transforms the VE or the user's motion. In addition to transformation techniques, additional methods include dynamically altering the VE [24], harnessing change blindness to manipulate the user's motions [25], and manipulating optic flow to induce self-motion [26].

When freely walking in the locomotion interfaces mentioned above, users may find themselves about to walk out of the tracked space. When a user nears the edge of the tracked space a reorientation technique (ROT) is used to prevent the user from leaving the tracked space [8], [27]. ROTs rotate the VE around the user's current location, returning the user's predicted path to the tracked space. The user must also reorient his body by physically turning so he can follow his desired path in the newly-rotated VE.

RFED transforms the VE by rotation, which has advantages over other transformations because when people turn their heads at normal angular velocities the vestibular system dominates the visual system, thus enabling rotation of the VE visuals without the user noticing [28]. As demonstrated by [9], users can be imperceptibly redirected very little unless the redirection is performed while the user's head is turning, which desensitizes the visual system. Additionally, larger amounts of redirection can be accomplished imperceptibly during head turns [9], [29].

To elicit head-turns, the original RDW used prescribed paths through the VE that at predetermined locations required the user to physically turn her head and body. The principal aim of RFED is to remove this limitation and enable users to walk **freely** about in a VE.

Walking freely in VEs raises a new problem—how to ensure that the user avoids real-space obstacles. Our system uses distractors to guide users away from the tracker boundary.

Additionally, we introduce *deterrents*—objects in the virtual environment that people stay away from or do not cross.

While VE transformations, distractors, and deterrents enable large-scale real-walking in VEs, the effect of the transformations on navigational ability is unknown. The study presented in this paper, first presented in [30], evaluates the effect of rotational transformations, distractors, and deterrents on navigational ability.

3 Basic RFED System

The basic RFED algorithm enables free-walking in large-scale VEs in four steps:

1. **Predict**: At each frame, predict the user's real-space future direction, v_{future}.
2. **Steer**: Steer the user's predicted future direction, v_{future}, to a steer-to location, s, such as the center of the tracked-space, and rotate the VE around the user such that v_{future} is rotated toward *s*.
3. **Distract**: When needed, introduce a distractor to:
    - **a.** a) Stop the user.
    - **b.** b) Elicit head-turns, enabling large amounts of redirection to steer the user's future direction to the steer-to-location of the tracked space.
4. **Deter**: When needed, introduce a *deterrent*, an object in the virtual environment that people avoid, to guide the user away from real-world locations where the user should not walk, such as the boundary of the tracked space.

Our implementation of these four steps was guided by a desire to minimize the number of distractors. Although distractors enable free-exploration, they are indeed distractions and therefore weaken the VE experience. Our implementations of RFED direct the user's path so that it crosses the center of the tracked space, maximizing the distance between the user and the tracker boundary.

To **steer**[1] the user toward the center of the tracked space requires **predicting** the user's real-space future path. In our steer-to-center implementation, if there were no limits to the amount of redirection that could occur at any instant, then users could always be steered directly toward the center of the tracked space, they would never reach a boundary, and a distractor would never be needed.

However, the human visual system limits the amount of imperceptible or non-obtrusive[2] redirection that can occur at any instant. To limit the number of distractions, we must redirect the user away from the boundaries as quickly as possible. We use *efficient redirection*, to redirect as quickly as possible given the limits of imperceptible redirection by:

[1] In this paper we use the term *redirection* to mean the rotation around the global up-axis at the user's real-world location, and *steering* as guiding the user away from the tracker boundary, which is accomplished through redirection.

[2] By non-obtrusive we mean to not increase simulator sickness or cognitive load.

1. Maximizing the *instantaneous redirection*—redirection per frame—within biological limits, to quickly steer the user away from the boundary.
2. Minimizing the total VE redirection, i.e. determining the best direction to rotate the VE, and thus minimizing the time used to steer the user away from the boundary.

Efficient redirection is designed to steer the user away from the tracker boundary, however when steering fails a **distractor** is used to prevent the user from leaving the tracked space. The distractor elicits user head-turns, thus enabling large amounts of redirection to steer the user back toward the center of the tracked space. If the user is very close to the tracker boundary, we **deter** the user from the boundary with a deterrent.

4 Efficient Redirection

At any instance, redirection is a transformation of the VE consisting of a rotation about the user's current location. The redirection transformation is updated **each frame** to steer the user away from the tracker boundary as quickly as possible. For **each frame**, efficient redirection requires answering two questions:

1. Which direction (clockwise or counter-clockwise) should the VE rotate to minimize the total amount of redirection required to steer the user to stay within the tracked space?
2. What is the maximum amount of imperceptible (or non-obtrusive) VE rotation that can be added to the current frame?

Answering these two questions determines θ_{VE}, the magnitude and direction of VE rotation to add to the current frame. θ_{VE} is calculated through six steps that are visually illustrated in Figure 1. As an aid to the reader, a reference of the variables used in the following steps is found in Table 1.

Which direction should the VE rotate?

1. Predict the user's immediate future virtual direction, v_{future}. In the VE, the user will walk in the direction of v_{future}. Rotating the VE redirects the user's real-world future direction, enabling steering of the user to stay within the real space. An inaccurate prediction of v_{future} reduces RFEDs ability to steer the user away from the tracker boundary, and may accidentally steer the user toward the boundary. We discuss algorithms for predicting v_{future} in Section 4.1.

2. To steer the user toward s, the steer-to point, v_{future} must be redirected toward s. Define a vector v_s from the user's real location to s, the steer-to-point. We defined s as the center of the tracked space, however s does not need to be the center of the tracked space or even a fixed location; We discuss our steering algorithm further in Section 4.2.

3a. Calculate the minimum angle from v_{future} to v_s, θ_{ideal}. Rotating the VE by θ_{ideal} will rotate the VE such that v_{future} is in the direction of s.

What is the maximum amount the VE can rotate without being perceived?

3b. Calculate the angular velocity of the user's head since the previous frame, ω_{head}.

4. Calculate θ_{VE}, as a function of ω_{head} and θ_{ideal}. Our algorithm for calculating θ_{VE} is discussed in Section 4.2.

5. Rotate the VE by θ_{VE}.

6. The user physically rotates by θ_{VE}, so as to walk in the direction of *s*. Notice that θ_{ideal} is now smaller.

4.1 Direction Prediction—Having an accurate prediction of the user's immediate future direction, $v_{f\ uture}$, is essential in determining θ_{VE} the minimum VE rotation to keep the user in the tracked space.

In this section, we describe the development of the direction prediction algorithms used in studies presented in [10], [30], and in this paper. Although the final algorithms enabled every participant to successfully walk virtual mazes, participants frequently had to be prevented from leaving the tracked space by distractors or deterrents. Improving the direction prediction algorithm could reduce the number of distractors and deterrents, and thus improve RFED usability.

Direction Prediction, Version 1: Our first direction prediction algorithm which was not used in any system implemented basic direction prediction by defining $v_{f\ uture}$ to be the user's look direction, v_{look}, reported by the head-tracker. This implementation of path prediction was based on results from [31] suggesting that gaze direction and heading direction are the same approximately 70% of the time. The problem with this simple path prediction model arose when people turned their heads, quickly changing v_{look}, and thus changing θ_{ideal}. The rapid change in v_{look} changed the direction of θ_{ideal} when v_{look} moved to the right and left of *s*. The rapid direction change of θ_{ideal} caused θ_{VE} to continually change between clockwise and counter-clockwise VE rotation. If the VE rotates clockwise one frame and then counter-clockwise the next frame (or vise-versa), the net redirection is reduced, thus hindering the system's ability to consistently steer the user away from the edge of the tracked space.

An alternative to using an instantaneous v_{look} to predict v_{future} is to average v_{look} over time. However, pilot experiments suggested that averaging v_{look} over time provided an unusable usable prediction of v_{future} producing the same changes in direction as described above.

Direction Prediction, Version 2: (Used in [10]). This approximation of v_{future} assumes that people continue walking in the same direction in the VE.

We calculated walk direction as the difference between user locations over time as determined from the head-tracker, and it is therefore not an instantaneous measure like look direction. The average direction of the user, v_{user} was calculated as the average of the difference in user's previous virtual locations. Pilot studies guided the selection of parameters for this method. The user's 2D virtual location, l_i, was sampled every 30 frames (about every 0.12 seconds), and v_{user} was averaged using the most recent 30 samples $\{l_0, l_1, \ldots, l_{29}\}$, such that:

$$v_{user} = \frac{\sum_{i=0}^{28} l_{i+1} - l_i}{29} \quad (1)$$

The direction prediction algorithm used in the study presented in [10] defined $v_{f\,uture} = v_{user}$ as calculated from Equation 1. This direction prediction implementation enabled all participants to complete the experiment and freely walk in a VE that was larger than the tracked space. However, errors occurred when users reached the edge of the lab or stood still.

Direction Prediction, Version 3: (Used in [30]). This direction prediction algorithm, presented in Figure 2, uses information about the VE to improve the prediction of $v_{f\,uture}$, and is composed of the following steps:

Step 1: Define a bidirected graph over the VE such that nodes are locations in the VE where people may change direction, and edges are straight paths in the VE. Specifically, for the maze environments used in [30], we defined the edges of the graph as hallways, and nodes as intersections and dead-ends of hallways. Note that a grid could be used as a generic graph of any environment. Defining a graph over the VE does not restrict user movement because the user is not required to walk to nodes and the user can freely change directions. **Step 2**. Determine the nearest node, *p* that is on the user's current path in the bidirectional graph. The user is expected to walk in the direction of one of the nodes connected to *p*. In figure 2, the user is predicted to walk toward p_a or p_b.

Step 3: Define vectors $\mathrm{v_i}$ from the user's virtual location to each node connected to *p*. In Figure 2, define vectors v_a and v_b.

Step 4: Calculate the angles between each v_i and v_{user} (Equation 1). Assume that the user will walk toward the node that has the smallest angle between v_{user} and v_i. In Figure 2, the user is predicted to walk toward node p_a, and we therefore define $v_{future} = v_a$.

Although this algorithm produced better results than using the average direction v_{user} to predict v_{future}, there were stability problems when *p*, the node closest to the user, rapidly changed back and forth between two nodes. We recommend that future developers focus on improving direction prediction. Improvements to direction prediction may include incorporating robotic path planning algorithms and determining a better way to define a directed graph of the environment. Another improvement to direction prediction may include a better estimation of v_{future} by averaging over different times or incorporating the look direction of the user. Work by [32] is also promising and may improve path prediction by extracting the user's instantaneous future direction from head-tracker information.

4.2 Steering

We use a *steering algorithm* to predict the user's future direction and rotate the virtual space to steer the user's real world direction away from the boundaries of the tracked space. The steering algorithm determines the direction and maximum angular rotation to apply to the VE to steer the user's predicted future direction toward a real-world steer-to location.

In the redirection algorithm described in Section 4, we always define the steer-to-point, *s*, to be the center of the tracked space and calculate the direction and angle of the shortest arc between $v_{f\,uture}$ and v_s (Figure 1, 3a.). In theory, this algorithm should steer the user through the longest path across the lab. However, evaluation of different steering algorithms such as steer-to-circle, or steer-to-moving target [9], may produce better results.

After determining the direction to rotate $v_{f\,uture}$ toward the center of the tracked space, we calculate the maximum magnitude of the rotation based on the user's angular head speed around the up axis, ω_{head}. The faster the user turns his head, the less aware he will be of VE rotation. Therefore, we rotate the VE by θ_{VE}, where θ_{VE} is a function of user head-turn speed and a predefined rotation constant, *c*.

$$|\theta_{VE}|=|\omega_{head}|*c \quad (2)$$

That is, for each frame, the faster the user turns her head, the greater the rotation of the VE. Based on pilot experiments and research from [29], we chose *c* to be 0.10 when the VE is rotating in the same direction as ω_{head} and 0.05 when the VE rotates against ω_{head}. Since the ideal amount of rotation is θ_{ideal}, if $\theta_{VE} > \theta_{ideal}$ then we set $\theta_{VE} = \theta_{ideal}$.

We have defined θ_{VE} as a linear function of ω_{head} and capped its value based on θ_{ideal}. Further evaluation of user perception of VE motion during head turns, and determining the maximum amount of acceptable perceptual VE rotation will determine a better maximum value for θ_{VE}.

In summary, for each frame the path prediction algorithm predicts the user's future direction v_{future} and the steering algorithm determines θ_{ideal}, the direction and maximum rotation of the VE to steer v_{future} to a steer-to point *s*. The instantaneous rotation of the VE, θ_{VE} is a function of θ_{ideal} and ω_{head}. The VE is rotated by θ_{ideal} around the user to redirect the user to stay within the tracked space. Although every attempt is made to steer the user away from the edges of the tracked space, there are times when redirection fails and a ROT must be used to steer the user back into the tracked space.

5 Distractors

The distractor implementation described in [8], [27] uses distractors only when redirection is unable to steer the user away from the tracker boundary. Here, we present the implementation described in [8], [27] and presented in [10]. Our distractor was a hummingbird that flew back and forth in front of the user. See Figure 4 A. We further develop an alternative implementation and additionally use distractors to increase redirection before the user nears the tracker boundary.

Both distractor implementations involve continually answering two questions:

1. When should the **distractor appear**?
2. When should the **distractor disappear**?

Distractor Algorithm, Version 1

Distractors are used to stop the user from leaving the tracked space and to elicit head turns, enabling redirection to steer the user's path back into the tracked space.

We defined a boundary region 1*m* from the edge of the tracked space and defined the following criteria to determine when the distractor appeared and disappeared (Figure 3 Version 1.)

- **Distractor-appear**: If the user is located in the boundary region and the distractor is not currently present.
- **Distractor-disappear**: Once the VE has rotated θ_{ideal} to within ε of v_s, where we used ε = 0.005°, and v_s is the vector to the center of the tracked space. That is, redirect the user until their predicted future direction is toward the center of the tracked space.

One problem with this implementation was that the **distractor-disappear** condition required reorienting the user all of θ_{ideal}. The greater the reorientation amount the longer the reorientation. In practice, some reorientations took as long as 30 seconds, an unacceptably long time. Users often reported increased frustration if the distractor did not disappear within 5 seconds after appearing.

Also, after the distractor disappeared, the user had to reorient her body by θ_{ideal} (Figure 1, Step 6.). For large reorientations, some participants noticed that the VE had changed. Although, some participants verbally acknowledged the VE change, it never inhibited any user's ability to turn her body and continue walking in her desired virtual direction.

A third problem was that the **distractor-appear** condition was triggered by the user being within the boundary region. If people did not step out of the boundary before the **distractor-disappeared**, the **distractor-appear** condition would be met and the distractor would reappear. To address this problem, participants were asked to take one step backwards. Since people were being redirected, the backwards step was not always toward the center of the lab and occasionally the experimenter had to physically guide the participant out of the boundary region.

Distractor algorithm, Version 2

In this implementation, distractors are not only used as ROTs, but also to preemptively increase redirection to steer the user away from the boundary (Figure 3 Version 2.). The criteria to determine when the distractor appears and disappears are as follows:

Distractor-appear

The distractor appears based on the inversely related values of *t*, the time since the previous distractor appeared, and *d*, the distance of the participant from the center of the lab. For example, when the participant is near the edge of the tracked space (a large *d*) and a distractor has not appeared within a small *t*, a distractor appears. If the user is near the center of the tracked space (small *d*) then a distractor will appear only if one has not recently appeared (a large *t*).

Distractor-disappear

The distractor disappears once the VE has rotated a percentage of θ_{ideal} based on *d*. The closer the participant is to the center of the lab, the smaller the percentage of θ_{ideal} the VE rotates before the distractor disappears. The values for *t*, *d*, and the percentage of θ_{ideal} used in [30] are in Table 2. For example, if the participant is 2.4 meters from the center of the tracked space and a distractor has not appeared in 16 seconds, the distractor will turn off after the VE rotates by $0.77\theta_{ideal}$.

Overall, this algorithm was promising because participants were kept from the edges of the lab, however the preemptive distractors appeared frequently and many participants complained about the over-abundance of distractions. For anyone implementing a distractor algorithm, we would recommend using distractors to steer people away from the edge of the tracked space, and develop less intrusive distractors that fit seamlessly into the environment. For example, if developing RFED for real estate applications, add a small dog, or children playing within the house. These objects will probably catch the participant’s attention and cause her to turn her head, thus enabling redirection.

Results from the studies in [8] suggests that people are less aware of redirection when a distractor is used. From this result, we piloted increasing the rotation constant *c* from Equation 2 when distractors were visible. The values of *c* used in the following experiment were 0.60 when with head rotation and 0.30 when against head rotation. Further perceptual threshold experiments similar to the experiments run in [29] could determine the most effective values of c with and without distractors. Considerations for determining an appropriate value of *c* include imperceptibility and the likelihood of increasing simulator sickness. However, different levels of *c* may be applicable for experienced and non-experienced users, for example more experienced users may easily tolerate perceptible VE rotation without increased simulator sickness. Participants were instructed that the VE would rotate around them during the experiment. Even though the rotation values were extremely high, participants commented that they noticed VE rotation only after they stopped moving their heads and had to reorient their bodies to continue walking in the same direction in the VE. Although some participants noticed that the VE had moved, no participant had problems reorienting to the rotated VE.

6 Deterrents

In the version 2 distractor algorithm, a distractor appeared every 3 seconds when the user was near the edge of the tracked space. If the user stayed near the boundary, distractors continually reappeared and frustrated many users. To guide the participant away from the boundary, we added deterrents to the environment. *Deterrents* are objects in the environment that people are instructed to avoid. For this implementation, deterrents were virtual horizontal bars that were aligned with the edge of the tracked space. See Figure 4 B. The bars fade in as the user approaches the boundary of the tracked space and fade out as the user walked away from the boundary.

The virtual bars provided participants with a visual cue as to which direction to walk to stay in the real space. No participant complained about the bars.

While the bars appear to inform the user where they cannot walk, they also provide the user with a visual cue about the size of the VE and the orientation of the tracked-space in relation to the participant. We originally thought this would cause a break-in-presence, however people often noted that the orientation of the VE was more dominant than the “real world” orientation of the bars.

A note about implementation. Since the stationary bars mark the location of the tracked space they do not move. If the VE rotates while the bars are visible people will quickly notice the VE rotation. However, in pilot experiments, people perceived the stationary bars to be rotating and not the VE. Therefore in the final version, reorientation of the VE did not occur when stationary deterrents were in view.

7 Evaluation

This study was first presented in [30]. We evaluated RFED by comparing it to Walking-in-place (WIP) and joystick (JS) interfaces, using navigation and wayfinding tasks:

Walking-In-Place

Participants in the WIP system condition locomoted by stepping in place. Advantages of WIP interfaces include: participants receive kinesthetic feedback from the in-place steps that move the viewpoint, and WIP interfaces can be implemented in small spaces. We used the GUD-WIP locomotion interface, a description can be found in [33], because it closely simulates real-walking by using walking biomechanics as inputs. Participants wore shin-guards equipped with Phase Space beacons and shin position was tracked with a PhaseSpace tracker. Heading direction was determined by the participant’s average-forward shin direction, and forward speed was a function of shin movement and stepping frequency.

Joystick

Participants in the JS condition controlled forward speed with a hand-held X-Box 360 controller. Deflection of the analog stick controlled speed. The maximum speed of the participant was chosen to be a moderate walking speed of 3 *mph*. JS participants also used the PhaseSpace tracker to determine heading direction from shin positions. Pushing forward on the joystick translated the participant’s viewpoint forward in the average direction of the participant’s shins.

The evaluation of RFED compared to WIP and JS required participants to locomote through the virtual mazes shown in Figure 5. The mazes were 15.85m × 15.85m, more than twice the dimension of the tracked space. Participants in the RFED condition were restricted to really walking in a space that was 6.5*m* × 6.5*m*, while participants in the WIP and JS conditions were confined to 1.5*m* × 1.5*m* area.

Turning, which stimulates the kinesthetic senses, is believed to aid navigation [11], [12]. Thus, we eliminated turning as a possible confounding factor by requiring users in all conditions to turn, i.e., change heading direction, by physically turning their bodies.

Navigation

Search tasks, which are commonly used in VE locomotion studies [34], are used to evaluate navigational ability and VE training-transfer of spatial knowledge for locomotion interfaces [35], [36]. Search tasks include *naïve searches*, in which targets have not yet been seen, and *primed searches*, in which targets have previously been seen.

Participants performed both na¨ıve and primed searches. Navigational performance was measured by the total distance participants traveled and the number of times participants revisited routes, i.e., returned to previously visited routes of the virtual mazes.

The distance participants travel is a measure of overall spatial knowledge accuracy [37]. Participants who travel shorter distances tend to have a better spatial understanding of the environment enabling them to walk directly to targets without unnecessarily retracing previous steps.

Wayfinding

Point-to-target techniques require participants to point to targets that they have previously seen, but that are currently out of view. Pointing tasks measure a user's ability to wayfind within VEs [12] by testing the user's mental model of target location in relation to the user's current location. Angular pointing errors with small magnitudes suggest that participants have a good understanding of the target locations.

Map completion requires users to place and label targets on a paper map of the VE after exiting the VE. The map target locations correspond to VE target locations. Map completion is often used as a wayfinding metric because maps are a familiar navigation metaphor [1]. Participants with a better mental model of the VE can more accurately place targets in correct locations and correctly label targets on the map.

7.1 Participants

Thirty-six participants, 25 men and 11 women, ($M = 26$, $\sigma = 5.1$), participated in the IRB-approved experiment. Twelve participants were assigned to each condition (8 men and 4 women in RFED and WIP, and 9 men and 3 women in JS).

7.2 Equipment

Each participant wore a stereo nVis nVisor SX head-mounted display with 1280×1024 resolution in each eye and a diagonal FOV of 60°. The environment was rendered on a Pentium D dual-core 2.8GHz processor machine with an NVIDIA GeForce GTX 280 GPU with 4GB of RAM. The interface was implemented in our locally developed EVEIL intermediate level library that communicates with the Gamebryo® software game engine from Emergent Technologies. The Virtual Reality Peripheral Network (VRPN) was used for tracker communication. The system latency was 50 ± 5*ms*.

The tracker space was 9*m* × 9*m* with head and hand tracked using a 3rdTech HiBall 3000.

The WIP and JS systems used an eight-camera PhaseSpace Impulse optical motion capture system with the cameras placed in a circle around the user. The user wore shin guards with seven beacons attached to each shin. PhaseSpace tracked the forward-direction and stepping motion of each leg. The GUD-WIP interface and Joystick direction detection code ran on a PC with an Intel Core2 2.4GHz CPU, NVIDIA GeForce 8600 GTS GPU, and 3 GB RAM.

7.3 Experimental Design

The experimental design is similar to the study presented in [10]. Participants locomoted through three virtual mazes: a training environment and two testing environments. See Figure 5. The virtual environments were 15.85*m* × 15.85*m* mazes with uniquely colored and numbered targets placed at specified locations. See Figure 4 C. All environments used the same textures on the walls and floors, and the same coloring and numbering of targets. The na¨ıve search included seven targets and the primed search included six targets. The VE for the primed search is similar to the na¨ıve search except that the walls are not all placed at 90° angles. This was done to make the experiment more challenging by removing feedback that enables users to determine cardinal directions from axis-aligned walls. The location of the targets changed between the naive and primed searches. All subjects completed the same trials in the same order to control for training effects, and were not given performance feedback during any part of the experiment. Subjects were randomly assigned to the RFED, WIP or JS condition, and completed all parts of the experiment, including training, in the assigned condition.

7.3.1 Training—Subjects received oral instructions before beginning each section of the experiment. The training environment was a directed maze with all walls placed at 90° angles. Subjects walked through the environment and pressed a button on a hand-held tracked device to select each of the seven targets which were placed at eye-height and located along the path. Participants had to be within an arm’s length to select a target. When a target was selected, a ring appeared around it and audio feedback was played to signify that the target had been found. See Figure 4 C.

After subjects completed the training maze, the HMD was removed and participants were asked to complete a 8.5” × 11” paper map of the environment. The map representation of the environment was a 16*cm* × 16*cm* overhead view of the maze with the targets missing. Participants were given their starting location and maps were presented such that the initial starting direction was away from the user. By hand, subjects placed a dot at the location corresponding to each target and labeled each target with its corresponding number or color.

7.3.2 Part 1: Naïve Search—After training, participants were given oral instructions for Part 1, the naïve search. The maze and target locations for Part 1 can be seen in Figure 5.

Participants were instructed to, in any order, find and remember the location of the seven targets within the maze. Participants were also reminded they would have to complete a map, just as in the training session. As soon as subjects found and selected all targets, the virtual environment faded to white and subjects were instructed to remove the HMD. Subjects then completed a map in the same manner as in the training session.

7.3.3 Part 2: Primed Search—After completing the na¨ıve search, subjects were given oral instructions for Part 2, the primed search. The maze and target locations for the primed search can be seen in Figure 5. Participants first followed a directed *priming path* that led to each of the six targets in a pre-specified order. After participants reached the end of the priming path the HMD faded to white, and the participants returned to the starting point in the VE. Participants in the RFED condition had to remove the HMD and physically walk to the starting location in the tracked-space. Participants using WIP or JS were asked if they wanted to remove the HMD, none did, and then they turned in place so they would be facing the starting forward direction in the virtual maze.

Participants were then asked to locomote, as directly as possible, to a specific target. Once the participant reached and selected the target, they were instructed via headphones to point, in turn, to each of the other targets. The instructions referenced targets by both color and number. After participants pointed to each other target, they were instructed to walk to another specific target where they repeated the pointing task. If a participant could not find a target within three minutes, arrows appeared on the floor directing the participant to the target. Arrows appeared in 2% of the trials and did not appear more frequently in any one condition. Once the participant reached the target, the experiment continued as before, with the participant pointing to all other targets.

Participants walked to the targets in the order 3-5-4-1-2-6 and, from each, pointed to each other target in numerical order. At the end of Part 2, subjects had pointed to each target five times, for a total of 30 pointing tasks per subject.

After completing the search and pointing tasks, subjects removed the HMD and completed a map just as in Part 1.

After the experiment, subjects completed a modified Slater-Usoh-Steed Presence Questionnaire [38] and a Simulator Sickness Questionnaire [39].

7.4 Results and Discussion

7.4.1 Part 1: Naïve Search

Navigation: Head-position data for all three conditions were filtered with a box filter to remove higher-frequency head-bob components of the signal. The filter width was three seconds. Participant travel distance was calculated from the filtered head pose data. Figure 6 shows the routes of the participant in each condition who's total distance is closest to the median total distance for that condition. Since participants were asked to find all the targets as directly as possible, our hypothesis is that participants who travel shorter distances have a better spatial understanding of the environment and of previously visited locations. Using a Mixed Model ANOVA with locomotion interface as the between-subjects variable and distance traveled as the dependent variable, we found a significant difference among locomotion interfaces, F(2,35)=4.688, *p*=0.016, *r*=0.353. See Figure 7.

We performed Tukey pair-wise, post-hoc tests on the distance traveled data, and applied a Bonferroni correction since multiple Tukey tests were applied on the same data. Bonferroni corrections were applied to all further Tukey pair-wise tests. Participants using RFED traveled significantly shorter distances than participants using WIP and JS, *p*=0.028 and *p*=0.037 respectively. No significant difference was found in locomoted distance between WIP and JS, *p*=0.992. These results suggest that participants using RFED had a better spatial understanding of the environment.

The number of times participants revisited routes was counted, where a *repeated route* was a route in the maze that a participant walked more than once. See Figure 7. We interpret repeated routes of the maze as indicating that participants were having a harder time building a mental model of the environment. We performed a Kruskal-Wallis test on the number of repeated routes and found a significant difference among locomotion interfaces for the number of times participants repeated routes of the maze when performing a naive search, H(2)=7.869, *p*=0.02. Pair-wise comparison post-hoc tests were performed. We found that participants using RFED revisited significantly fewer routes of the maze than participants using WIP, H(1)=−11.000, *p*=0.026. This suggests that, participants using RFED were not as lost, or had built a better mental model of the environment than participants using WIP. No significant difference was found comparing RFED to JS, or WIP to JS.

Wayfinding: We evaluated participants' ability to place and label each virtual target on a map of the VE. Targets were counted as correctly placed if they were within one meter (scaled) of the actual target and on the correct side of walls. Targets were counted as correctly labeled if they were both correctly-placed and were labeled with either the correct number or color. We performed two Mixed Model ANOVAs with locomotion interface as the between-subjects variable and percentage of correctly placed, and correctly-placed-and-labeled targets, as the dependent variables. No significant difference was found among locomotion interfaces in user ability to place targets on maps. However, there was a trend suggesting a difference between interfaces on ability to correctly place and label targets after the naive search, F(2,30)=2.591, *p*=0.092, ω = −0.683, see Figure 8.

Conclusion: The naive search showed RFED participants traveled significantly shorter distances than both WIP and JS participants and revisited significantly fewer routes in the maze than participants using WIP. These results suggest that, when performing a na¨ıve search, participants using RFED had a better understanding of where they had already been within the VE and had a better spatial understanding of the VE than participants using either WIP or JS. The total time to travel the VE mazes could effect participants' spacial understanding of the environments. Unfortunately we did not measure the total time to completion, however participants in all conditions completed the entire experiment in 40 minutes to an hour. Future studies should explore the total time taken when using different locomotion interfaces.

7.4.2 Part 2: Primed Search

Navigation: We calculated each participant's total travel distance to find each of the six targets for the primed search, in the same way as the na¨ıve search. The real and virtual routes from an RFED participant can be seen in Figure 9. While each participant travels the directed training path he builds a mental model of the environment. We assert that participants who build a better mental model while following the priming path will locomote shorter distances between targets during the search and pointing portions of the task.

We performed a MANOVA with locomotion interface as the between-subjects variable and distance traveled to each of the six targets as the within-subjects repeated measure. See Figure 10. We found a significant difference between locomotion interfaces on distance traveled, F(2,32)=7.150, *p*=0.003, *r*=0.427. Tukey post-hoc tests show that participants using RFED traveled significantly shorter distances than participants using WIP, *p*=0.002. No other significant results were found. This suggests that participants using RFED were better at navigating the VE than participants using WIP.

An additional path evaluation was performed by using a Kruskal-Wallis test on the total number of wrong turns taken by each participant during the primed search. A *wrong turn* occurs at an intersection when the participant does not take the shortest route to the current target goal. A significant difference was found between locomotion interfaces, H(2)=11.251, *p*=0.004. Pairwise comparisons, show that participants using RFED made significantly fewer wrong turns than those using either WIP, H(1)=−13.667, *p*=0.004, or JS, H(1)= −10.708, *p*=0.038. No significant difference was found between JS and WIP users, H(1)=2.958, *p*=1.00. These results suggest that participants in RFED had a better understanding of where they were going in the virtual maze and had a better mental model of the environment after having the same experience in the VE as participants in the WIP and JS interfaces.

Analysis of the routes taken to each individual target show significant differences between walking to target #1, the red target, and target #2 the green target, H(2)=6.505, *p*=0.039, and H(2)=8.881, *p*=0.012 respectively. Post-hoc tests reveal that participants using WIP made significantly more wrong turns when navigating to these two targets than participants using RFED, H(1)=−9.352, *p*=0.034, and H(1)=−11.727, *p*=0.01 respectively. It is interesting to note that during the priming portion of the task, participants visited target #1, the red target first, and visited target #2 the green target last. This may suggest that participants using WIP have problems in the beginning and end of the VE experience. Note: all subjects had to regularly stop and start locomoting to select the targets as they walked the directed path, and participants had to "walk" to get to target #1. Further evaluation of WIP interfaces should be explored, specifically looking at cognitive load at the beginning and end of a virtual experience. There was no significant difference for any of the individual routes between RFED and JS or WIP and JS.

Wayfinding: During the primed search, when subjects reached a target they were then asked to point to each of the other targets. See Figure 11. Small absolute angular pointing error suggests that participants have a better understanding of the location of targets. We ran a Mixed Model ANOVA with locomotion interface as the between-condition variable and absolute pointing error to each target as the repeated measure. There was a significant difference among locomotion interfaces for the absolute angular error when pointing to targets, F(2,28)=5.314, *p*=0.011, *r*= 0.399. Tukey pair-wise post-hoc tests reveal that participants using RFED had significantly smaller absolute pointing errors than both WIP and JS, *p*=0.021 and *p*=0.024 respectively. That is, participants using RFED had significantly better understanding of the location of targets in relation to their current location. There was no significant difference in absolute pointing error between WIP and JS, *p*=0.993.

In addition to evaluating point-to ability, we also analyzed how long participants took to point to each target. See Figure 12. We hypothesized that participants with a clearer mental model would be able to point more quickly to targets. The first pointing trial was also the first time participants pointed, thus we considered this as a training trial and removed it from the data. We ran a Mixed Model ANOVA with locomotion interface as the between-condition variable, and time to point to each target as the repeated measure and found a trend suggesting a difference in pointing time among locomotion interfaces, F(1,19)=2.992, *p*=0.074, *r*=0.369.

Further analysis of the first 14 trials, with the first trial removed, shows a significant difference among locomotion interfaces, F(2,23)=4.636, *p*=0.02, *r*=0.410. Tukey post-hoc tests show a significant difference between RFED and both WIP and JS, *p*=0.031 and *p*=0.050 respectively. This suggests that participants using RFED had a better mental model when pointing, compared to participants in WIP and JS, during the first half of the primed search. This result may imply that participants using RFED train faster than participants in either WIP or JS conditions, however further studies should evaluate interface training time.

We compared the difference in map completion ability using Mixed Model ANOVAs with locomotion interface as the between-subjects variable and percentage of correctly placed, and correctly-placed-and-labeled targets, as the dependent variable, and found a significant difference among interfaces in participant ability to correctly place and label targets, F(2,30)=3.534, *p*=0.042, $\omega = -0.603$. See Figure 8. Tukey pair-wise post-hoc tests revealed a significant difference between RFED and WIP in correctly placing and labeling targets on maps after completing the primed search, *p*=0.034. No other significant differences were found.

Conclusion: The primed search results suggest that participants using RFED navigate and wayfind significantly better than participants using WIP or JS. RFED participants travel shorter distances than participants using WIP, suggesting that RFED participants have a better spatial understanding of the environment and consequently walk more directly to targets. Participants using RFED make fewer wrong turns than WIP and JS participants, providing additional evidence that RFED participants walk more directly to the goal targets, and hence are better at navigating the environment.

Participants in RFED were significantly better at wayfinding than participants in WIP or JS. RFED participants had significantly smaller absolute pointing errors than those using either WIP and JS. In addition to pointing to targets more accurately, participants using RFED are also better at placing and labeling the targets on maps than participants using WIP. This further suggests that participants in RFED develop a better mental model than WIP participants.

Finally, RFED participants point more quickly to targets in the beginning of the experiment than participants in both WIP and JS, suggesting that participants using RFED build mental models faster, however further studies should be run to verify this result. Overall, participants using RFED point to targets more accurately, complete maps with fewer mistakes, and are quicker at pointing to targets in the first half of the experiment.

7.4.3 Post Tests—After completing the final map, participants were asked to estimate the size of the VEs compared to the size of the tracker space they were currently in. See Table 3. Subjects were told that all three virtual environments were the same size and were given the dimensions of the tracked space. We found a significant difference between VE size predictions based on locomotion condition, F(2,31)=6.7165, *p*=0.006, *r*=0.742. Tukey pair-wise post-hoc tests reveal differences between RFED and both WIP (*p*=0.033) and JS (*p*=0.007). The results suggest that people have a better understanding of VE size when using RFED than with both WIP or JS.

One possible confounding factor was that participants in the RFED condition saw virtual bars in the environment that represented the location of the bounds of the tracked space in the real lab. Based on the design of the maze, participants were not able to see more than 25% of the deterrent boundary at any given instant and usually saw less than 10% of the deterrent boundary. This “real world”-sized reference may have given people in the RFED condition an advantage in estimating the size of the VE. However, two participants in the RFED condition asked to walk around the room before making a guess as to the dimensions of the VE. No participants in JS or WIP asked to walk around the room. This suggests that two participants in the RFED condition realized that their physical walking steps could help measure the size of the VE. The two participants were permitted to walk however neither estimated the lab size significantly better than other participants in the RFED condition.

Presence was evaluated using a modified Slater-Usoh-Steed presence questionnaire [38]. The number of “high” presence scores were counted, scores with a 5 or higher, and a Pearson’s chi-square test was performed on the transformed data. No significant difference was found among locomotion interfaces and the number of “high” presence scores, $\chi^2(12) = 14.143$, *p*=0.292. We hypothesized that RFED would produce higher presence scores because it is more similar to real walking [7] than JS and WIP. Possible reasons for the lack of difference in presence scores include: participants in RFED were more likely to be aware of the HMD cables, and RFED participants had a greater number of tracker failures due to walking near the edge of the tracked space where the tracker is less stable.

Participant simulator sickness scores were calculated using Kennedy’s simulator sickness questionnaire [39]. The data was analyzed with an ANOVA and no significant difference was found between locomotion interfaces and simulator sickness scores, *F*(2,31) = 0.403, *p*=0.672.

8 RFED Limitations

The current RFED implementation is limited by distractors appearing too frequently, a result of participants reaching the edge of the tracked space too often. The frequent appearance of distractors is bothersome to some participants and efforts to reduce the number of distractors should be a main focus of future implementations. In the current implementation, on average a distractor appears after the participant travels 5*m*±4*m* and remains visible for 8*s*±2*s*. However, participants often travel longer distances than 5*m* before being stopped by a distractor. Once the participant stops, 2 or 3 distractors occasionally appear before the participant's future path is redirected to the center of the tracked space and resumes walking. Although the frequency of distractors is quite high, participants were able to successfully navigate the VEs and develop mental models, suggesting that people have a high tolerance for distractors. However, for improved usability only one distractor should appear to fully redirect the user, and any reduction in the current amount of distractors would improve the system.

The frequent occurrence of distractors is an obvious drawback of the current implementation. Improving the current redirection design and implementation, as well as determining how to encourage users to turn their heads, will reduce the occurrence of distractors and deterrents and the number of times participants reach the boundary of the tracked space.

Additionally, the implementation of RFED requires a large tracked area to enable redirection. With an average step length of 0.75*m*, in four steps a person can travel 3*m*, a typical tracking width. Although it is currently unknown as to how much redirection can be added at any instant, results from [29] suggest that, with head turns, the virtual scene can rotate 1.87°/sec and [40] suggests that people can be reoriented up to 30° when performing a 90° virtual rotation, i.e., the rotation could be 60° or 120°. From observation, if a person is walking straight and not turning her head, very little unobservable redirection can be added to the scene. Therefore, in a 3*m*×3*m* tracked space, the user may be stopped every 4 or 5 steps. In the experiment described above participants were rotated on average more than 2° for every traveled meter in the RFED condition. During the course of the experiment participants were turned completely around more than 800 times!

Based on the limitations of the amount of redirection that can be applied, redirection techniques are only feasible in large tracked spaces. With the current implementation of RFED, we recommend at least 6*m* × 6*m*, however a larger space should produce better results. For tracked spaces smaller than 6*m*×6*m*, we recommend using joystick or WIP interfaces, and only using RFED if the VE is designed such that on average the user travels 4 or 5 steps at a time, such as a building with small rooms. Future implementations of redirection techniques, or combinations of redirection techniques may produce better results in smaller tracking spaces.

9 Future possibilities

We presented a basic redirection algorithm and the parameters used in the RFED implementation. Participants were able to successfully locomote and navigate VEs that were larger than the tracked space. However in the current RFED implementations distractors appeared too frequently, a result of participants reaching the edge of the tracked space too often. Improving the current redirection design and implementation, as well as determining how to encourage users to turn their heads quickly, will reduce the number of times participants reach the boundary.

9.1 Redirection

Redirection can be thought of as determining the instantaneous rotation, θ_{VE}, of the VE around the user such that the total VE rotation over time is minimized, while the instantaneous per frame VE rotation is maximized. One area of future study is to determine the maximum per frame θ_{VE} to maximize efficient redirection. Research by [29] suggests lower bounds for imperceptible rotation that can be added to the VE during head turns, however the rotation amounts used in RFED were larger than those presented in [29]. Participants did not complain about VE rotation, nor did the VE rotation significantly increase simulator sickness compared to participants using walking-in-place or joystick interfaces. This suggests that rotation in RFED may not need to be imperceptible, however the gold standard would demand imperceptibly and an upper bound for imperceptible redirection is currently not known.

Direction prediction—One of the most difficult parts of redirection is predicting the user’s future direction. Having accurate direction prediction enables minimization of VE rotation. Based on user feedback and experimenter observation, we would recommend experimenting with enabling the user to interact with the system to define her own future direction. For example, if the user is at the tracker boundary, with a hand held wand, she could select her future VE location. This user selected location would then be steered into the tracked space. User interaction would remove the guess work and inaccuracies from the path prediction algorithm as well as reduce the time required to redirect the user. This technique would be specifically useful for non-na¨ıve users, while providing the benefits of really walking.

Designing path prediction without user input to the system is more challenging. Creating a statistical model of the environment to determine the most likely future user path could drastically improve the current direction prediction algorithm. We also recommend using motion planning algorithms, or previous user path information of specific environments to determine common user paths within environments.

Steering—The current implementation of steering always directs the user’s predicted future path to the center of the tracked space. Steer-to-farthest-corner, steer-to-circle, or steer-to-moving-targets may be able to steer the user to stay within the tracked space better than steer-to-center. Simulations of different steering algorithms on different virtual paths may provide insight into the best steering algorithm for RFED. Additionally, information from the direction prediction bidirected graph may also improve steering algorithms.

9.2 Distractors

The goals for distractor implementation are, (1) minimizing distractor appearance frequency and duration, and (2) minimizing distractor-related instruction. Future research areas for distractors include:

Minimize awareness—Results from [41] provide promising results for encouraging head turns through image modulations in peripheral vision. These results could remove the requirement of instructing users to watch distractors.

Naïve users—Current distractor implementations require initial instruction causing users to be non-naïve to distractors. Requiring users to be non-na¨ıve may increase cognitive load or have unknown negative effects on usability. Developing distractors for a completely na¨ıve user without requiring initial instruction will enable visually and cognitively imperceptible reorientation. A first approach could focus on intuitively natural distractors such as people walking by, children crawling on the ground, or playing dogs.

Appearance—Results from [8] suggest that the appearance of distractors has an effect on user preference. Additional study of distractor appearance should include studying animated versus rigid-body distractors, realistic versus non-realistic distractors, and looking at different colors, shapes, sounds, or sizes of distractors.

Motion paths—Current distractor implementations only move distractors in arcs located directly in front of the user. Further evaluation of different distractor motion paths may reduce user frustration or may result in more effectively encouraging user head turns. Additional research should focus on the motion and appearance of distractors in different parts of the FOV, specifically in the user's periphery when using a wide FOV HMD.

9.3 Deterrents

Deterrents were implemented as horizontal bars that marked the edge of the physical tracked space. Currently deterrents have been implemented as stationary virtual objects, however implementing deterrents as dynamic objects will provide additional ways to "steer" the user away from the edge of the tracked space. One implementation of deterrents could be avatars walking around the environment, such as visitors at a museum or shoppers in a store, whose personal spaces would deter the user from the boundary of the tracked space.

9.4 Combining redirection techniques

RFED is not the only interface to combine multiple redirection techniques. [42] has combined scaled translational gains and resetting, and [23] developed a dynamic redirection controller based on the user's walking speed, and added avatars to redirect the user. Different combinations of redirection and reorientation techniques are likely to enable different results and experiences. Promising future work would compare different combinations of techniques to guide the VE designer. For example, if a mental model of the environment is not a requirement than dynamically altering the VE [24] may produce a more usable interface since it would not use distractors. For training transfer applications where fatigue is important, scaled translational gain methods may not be feasible, however scaled translational gain may be most appropriate for a novice user walking through a virtual city. Possible design goals may include: accurate development of a mental model, usability, user enjoyment, speed of travel, training transfer, and designing for experienced vs. novice users.

10 Conclusion

The user study presented here shows that RFED is significantly better than walking-in-place based on the same navigation metrics that were used in [10]. Researchers have shown that walking interfaces are significantly superior to joystick interfaces on many kinds of measures [7], [36]. Trends have been seen suggesting that real walking is superior to walking-in-place but no previous results have been able to show a statistically significant superiority. The study presented in this paper does so. We compared to a stat-of-the-art WIP system, GUD-WIP [33]. We developed a real-walking system, RFED, that enabled free exploration of larger-than-tracked-space virtual environments. Pairwise comparisons showed that RFED was significantly superior to GUD-WIP and a JS interface on several navigation measures. Moreover, RFED was never significantly worse than either the JS and WIP interfaces on any measured metric in this study.

Further, we found no significant differences between our JS implementation and GUD-WIP. This was a surprising result, as we expected GUD-WIP to out-perform JS. We believe one reason for the lack of a significant differences stems from the challenges of stopping and starting in WIP systems, including GUD-WIP. The virtual mazes required many starts and stops to select and point to targets and WIP participants occasionally walked through targets when trying to stop in front of them. Additionally, when starting to walk, participants intentionally "walked" through targets in both JS and WIP interfaces instead of walking around them, while RFED participants did not walk through targets.

Although no navigational differences between WIP and JS were found, our results further support that real walking is critical for VE navigation. Even though RFED continuously rotates the VE around the user and frequently stops the user with distractors, RFED participants were significantly better at navigating VEs than both WIP and JS participants. The physical difference between the systems is the stimulation of the proprioceptive system. There was no kinesthetic difference between interfaces since heading direction was controlled by physical heading direction. Although WIP stimulates the proprioceptive system, RFED more accurately stimulates the proprioceptive system as it is more similar to real walking.

Our results support that accurate stimulation of the proprio-ceptive system is critical to navigation and even with rotation of the VE around the user and distractors, participants were able to navigate VEs significantly better than without accurate proprioceptive stimulation. Further development of RFED and RFED-like systems to improve usability will further aid VEs and the intended goal of free exploration of large spaces without user awareness of the enabling techniques.

Acknowledgments

Support for this work was provided by the Link Foundation. The authors would like to thank the EVE team, especially Jeremy Wendt for the use of his GUD-WIP system, and the anonymous reviewers for their thoughtful and helpful comments.

References

1. Darken, .; Rudolph, P.; Peterson, .; Barry, . Spatial Orientation, Wayfinding, and Representation. Mahwah, NJ: Lawrence Erlbaum Associates; 2002. p. 493-518.The Handbook of Virtual Environments
2. Darken , Rudolph P, Sibert , John L. “Wayfinding strategies and behaviors in large virtual worlds,” CHI ‘96. Proceedings of the SIGCHI conference on Human factors in computing systems. 1996:142–149.
3. Durlach, NI.; Mayor, AS. Virtual reality: scientific and technological challenges. Washington, DC: National Academy Press; 1995.
4. Stuart C, Grant , Lochlan E. Magee. “Contributions of proprioception to navigation in virtual environments”. Human Factors. Sep.1998 40.3:489. [PubMed: 9849106]
5. Joseph, Psotka. “Immersive training systems: Virtual reality and education and training”. Instructional Science. 1995; 23:405–431.
6. Mel Slater, Martin Usoh; Anthony, Steed. “Taking steps: the influence of a walking technique on presence in virtual reality”. ACM Trans. Comput.- Hum. Interact. 1995; 2:201–219. no. 3.
7. Martin, Usoh; Kevin, Arthur; Mary C., Whitton; Rui, Bastos; Anthony, Steed; Mel, Slater; Frederick P., Brooks, Jr. “Walking> walking-in-place > flying, in virtual environments.”. SIGGRAPH 99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. 1999:359–364.
8. Peck, Tabitha C.; Fuchs, Henry; Whitton, Mary C. “Evaluation of Reorientation Techniques and Distractors for Walking in Large Virtual Environments”. IEEE Transactions on Visualization and Computer Graphics. 2009; 15:383–394. no. 3. [PubMed: 19282546]
9. Sharif, Razzaque. “Redirected Walking”. PhD Dissertation, University of North Carolina at Chapel Hill: Department of Computer Science; 2005.
10. Peck TC, Fuchs H, Whitton MC. “Improved Redirection with Distractors: A Large-Scale-Real-Walking Locomotion Interface and its Effect on Navigation in Virtual Environments”. IEEE Conference on Virtual Reality. 2010:35–38.
11. Ruddle, Roy A.; Lessels, Simon. “The benefits of using a walking interface to navigate virtual environments”. ACM Trans. Comput.-Hum. Interact. 2009; 16:1–18. no. 1. [PubMed: 22563232]
12. Chance S, Gaunet F, Beall A, Loomis J. “Locomotion mode affects the updating of objects encountered during travel: The contribution of vestibular and proprioceptive inputs to path integration. Presence. Apr; 1998 7(2):168–178.
13. Darken RP, Cockayne WR, Carmein D. “The omni-directional treadmill: a locomotion device for virtual worlds”. UIST 97: Proceedings of the 10th annual ACM symposium on User interface software and technology. 1997:213–221.
14. Hollerbach, JM. Locomotion Interfaces, ser. The Handbook of Virtual Environments. Vol. ch. 11. Mahwah, NJ: Lawrence Erlbaum Associates; 2002. p. 493-518.
15. Razzaque, S.; Kohn, Z.; Whitton, MC. EUROGRAPHICS 2001 /. Jonathan C. Roberts Short Presentation The Eurographics Association; 2001. “Redirected walking.
16. Razzaque S, Swapp D, Slater M, Whitton MC, Steed A. “Redirected walking in place. EGVE 02. 2002:123–130.
17. Robinett W, Holloway R, “Implementations of flying. scaling and grabbing in virtual worlds. Interactive 3D graphics. 1992:189–192.
18. Williams B, Narasimham G, McNamara TP, Carr TH, Rieser JJ, Bodenheimer B. “Updating orientation in large virtual environments using scaled translational gain. APGV 06: Proceedings of the 3rd symposium on Applied perception in graphics and visualization. 2006:21–28.

19. Williams B, Narasimham G, Rump B, McNamara TP, Carr TH, Rieser J, Bodenheimer B. "Exploring large virtual environments with an hmd when physical space is limited. APGV 07: Proceedings of the 4th symposium on Applied perception in graphics and visualization. 2007:41–48.

20. Interrante V, Ries B, Anderson L. "Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. IEEE 3D User Interfaces. 2007:167–170.

21. Nitzsche N, Hanebeck UD, Schmidt G. "Motion compression for telepresent walking in large target environments. Presence: Teleoper. Virtual Environ. 2004; 13:44–60. no. 1.

22. Su J. "Motion compression for telepresence locomotion. Presence. 2007; 16:385–398. no. 4.

23. Neth CT, Souman JL, Engle D, Kloos U, Bülthoff HH, Mohler BJ. "Velocity-dependent dynamic curvature gain for redirected walking. IEEE Conference on Virtual Reality. 2011:151–158.

24. Suma E, Clark S, Krum D, Finkelstein S, Bolas M, Warte Z. "Leveraging change blindness for redirection in virtual environments. IEEE Conference on Virtual Reality. 2011:159–166.

25. Steinicke F, Bruder G, Hinrichs KH, Willemsen P. "Change blindness phenomena for stereoscopic projection systems. IEEE Conference on Virtual Reality. 2010

26. Bruder G, Steinicke F, Wieland P. "Self-motion illusions in immersive virtual reality environments. IEEE Conference on Virtual Reality Conference. 2011:39–46.

27. Peck T, Whitton M, Fuchs H. "Evaluation of reorientation techniques for walking in large virtual environments. IEEE Conference on Virtual Reality. Mar.2008 :121–127.

28. Duh HB-L, Parker DE, Phillips J, Furness TA. ""Conflicting motion cues at the frequency of crossover between the visual and vestibular self-motion systems evoke simulator sickness. Human Factors. 2004; 46:142–153. [PubMed: 15151161]

29. Jerald J, Peck T, Steinicke F, Whitton M. "Sensitivity to scene motion for phases of head yaws. APGV 08: Proceedings of the 5th symposium on Applied perception in graphics and visualization. 2008:155–162.

30. Peck TC, Fuchs H, Whitton MC. "An evaluation of navigational ability comparing redirected free exploration with distractors to walking-in-place and joystick locomotion interfaces. IEEE Conference on Virtual Reality. 2011:55–62.

31. Hollands MA, Patla AE, Vickers JN. ""Look where youre going!: Gaze behaviour associated with maintaining and changing the direction of locomotion. Experimental Brain Research. 2002; 143:221–230. [PubMed: 11880898]

32. Wendt, J. "Real-walking models improve walking-in-place systems. PhD Dissertation, University of North Carolina at Chapel Hill: Department of Computer Science; 2010.

33. Wendt J, Whitton MC, Brooks F. "GUD WIP: Gait-understanding-driven walking-in-place. IEEE Conference on Virtual Re-ality. 2010:51–58.

34. Bowman, DA. Principles for the Design of Performance-oriented Interaction Techniques, ser. Vol. ch. 13. Mahwah, NJ: Lawrence Erlbaum Associates; 2002. p. 277-300.The Handbook of Virtual Environments

35. Waller D, Hunt E, Knapp D. "The transfer of spatial knowledge in virtual environment training. Presence. Apr.1998 7:129–143. No. 2.

36. Witmer BG, Bailey JH, W Knerr B, Parsons KC. "Virtual spaces and real world places: transfer of route knowledge. Int. J. Hum. Comput. Stud. 1996; 45:413–428. No. 4.

37. Ruddle RA. Navigation: Am I really lost or virtually there, ser. 2001:135–142. Engineering psychology and cognitive ergonomics - Volume 6. Ashgate.

38. Slater M, Steed A. "A virtual presence counter. Presence. 2000; 9:413–434. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/105474600566925no. 5.

39. Kennedy R, Lane N, Berbaum K, Lilenthal M. "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. The International Journal of Aviation Psychology. 1993:203–220.

40. Steinicke F, Bruder G, Hinrichs K, Jerald J, Frenz H, Lappe M. "Real walking through virtual environments by redirection techniques. Journal of Virtual Reality and Broadcasting. 2009 Feb.6 no. 2.

41. Bailey R, McNamara A, Sudarsanam N, Grimm C. “Subtle gaze direction. SIGGRAPH 07: ACM SIGGRAPH 2007 sketches. 2007:44.

42. Xie X, Lin Q, Wu H, Narasimham G, McNamara TP, Rieser J, Bodenheimer B. “A system for exploring large virtual environments that combines scaled translational gain and interventions. Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization. 2010:65–72.

Biographies

Tabitha C. Peck is currently a post-doctoral researcher at the Event Lab in Barcelona, Spain. She received her PhD from The University of North Carolina at Chapel Hill focusing on loco-motion interfaces in virtual environments. She is currently working in the European project Virtual Embodiment and Robotic Re-Embodiment (VERE) studying the psychological effects of embodiment in virtual environments. Her research interests include: Virtual reality, virtual embodiment, human-computer interaction, 3D user interfaces, locomotion, navigation, system design and evaluation, and human perception.

Henry Fuchs is the Federico Gil Distinguished Professor of Computer Science and Adjunct Professor of Biomedical Engineering at The University of North Carolina at Chapel Hill. Fuchs is a co-director, with Nadia Thalmann of NTU Singapore and Markus Gross of ETH Zurich, of the NTU-ETH-UNC “BeingThere” International Research Centre for Tele-Presence and Tele-Collaboration. In 1975 he received a Ph.D. in Computer Science from the University of Utah. He has been active in computer graphics since the 1970s, with rendering algorithms (BSP Trees), hardware (Pixel-Planes and PixelFlow), virtual environments, tele-immersion systems and medical applications. He is a member of the National Academy of Engineering, a fellow of the American Academy of Arts and Sciences, a fellow of the ACM, and recipient of the 1992 ACM-SIGGRAPH Achievement Award.

Mary C. Whitton is a research associate professor of computer science at the University of North Carolina at Chapel Hill. She, with Fred Brooks, co-leads the Effective Virtual Environments research group that investigates what makes virtual environment systems effective and develops and evaluates techniques to make them more effective for applications such as simulation, training, and rehabilitation. Before joining UNC in 1994, she was co-founder of two companies that produced high-end hardware and software for graphics, imaging, and visualization. Ms. Whitton has held leadership roles in ACM SIGGRAPH including serving as President 1993–1995. She is a member of ACM, ACM SIGGRAPH, and a senior member of IEEE.

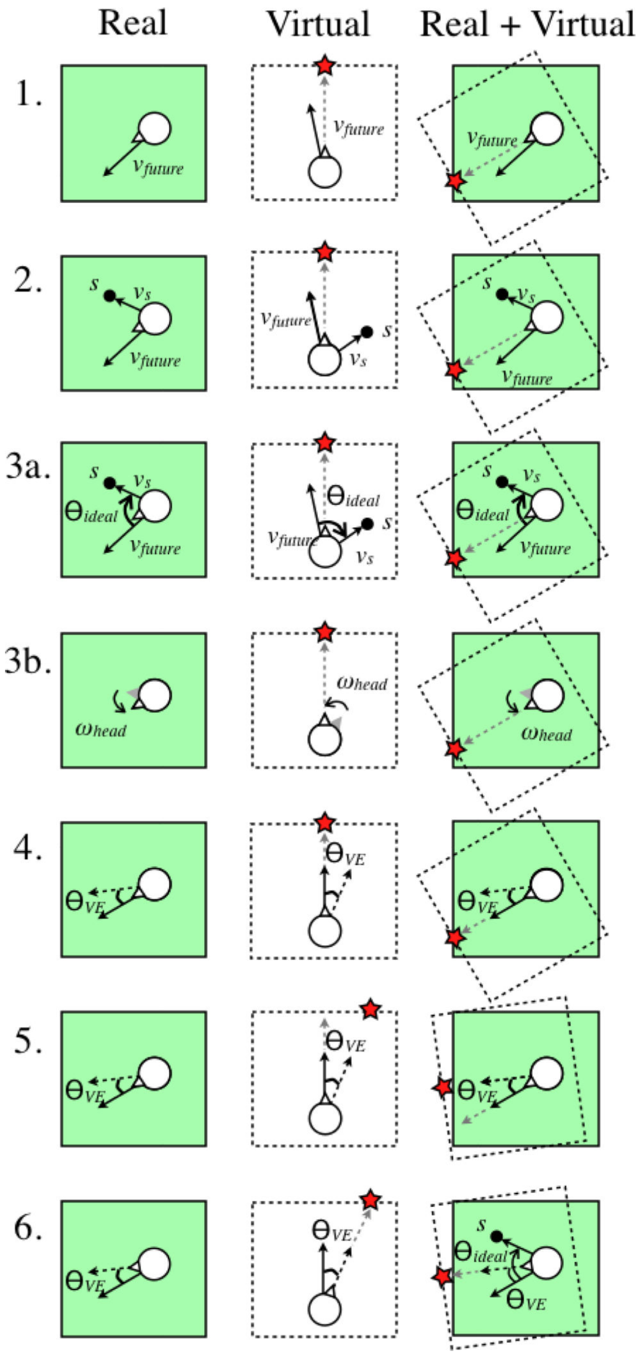

Fig. 1.

The six steps of *efficient redirection*. These steps are discussed in Section 4. The star is a virtual reference point. Notice that the star moves with the VE in **Step 5**.

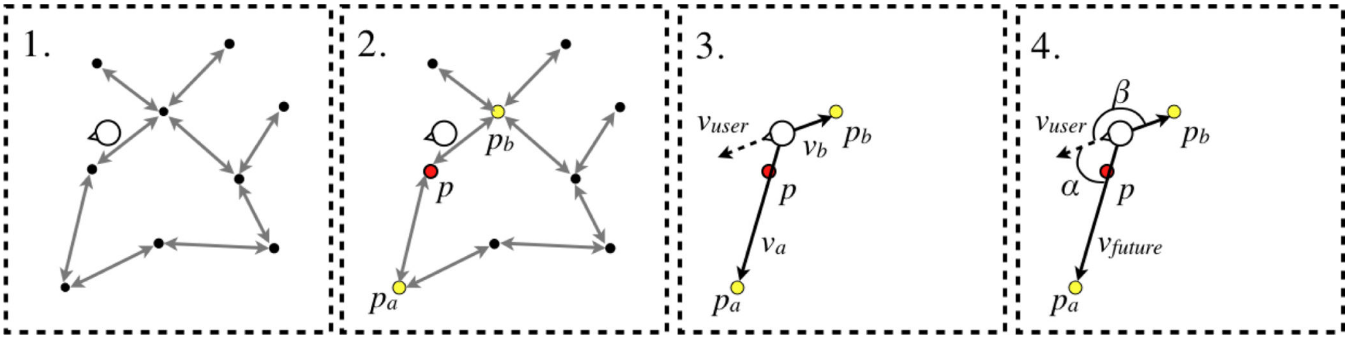

Fig. 2.

Step 1. Define a bidirected graph over the VE. **Step 2**. Identify the node closest to the user (p), and the nodes connected to p, (p_a and p_b). **Step 3**. Define vectors v_a and $v_b$from the user to connected nodes p_a and p_b**Step 4**. Calculate and compare the angles α and β between the user direction vector, v_{user} (Equation 1) and vectors v_a and v_b. Since α is smaller than β, set v_a as v_{future}.

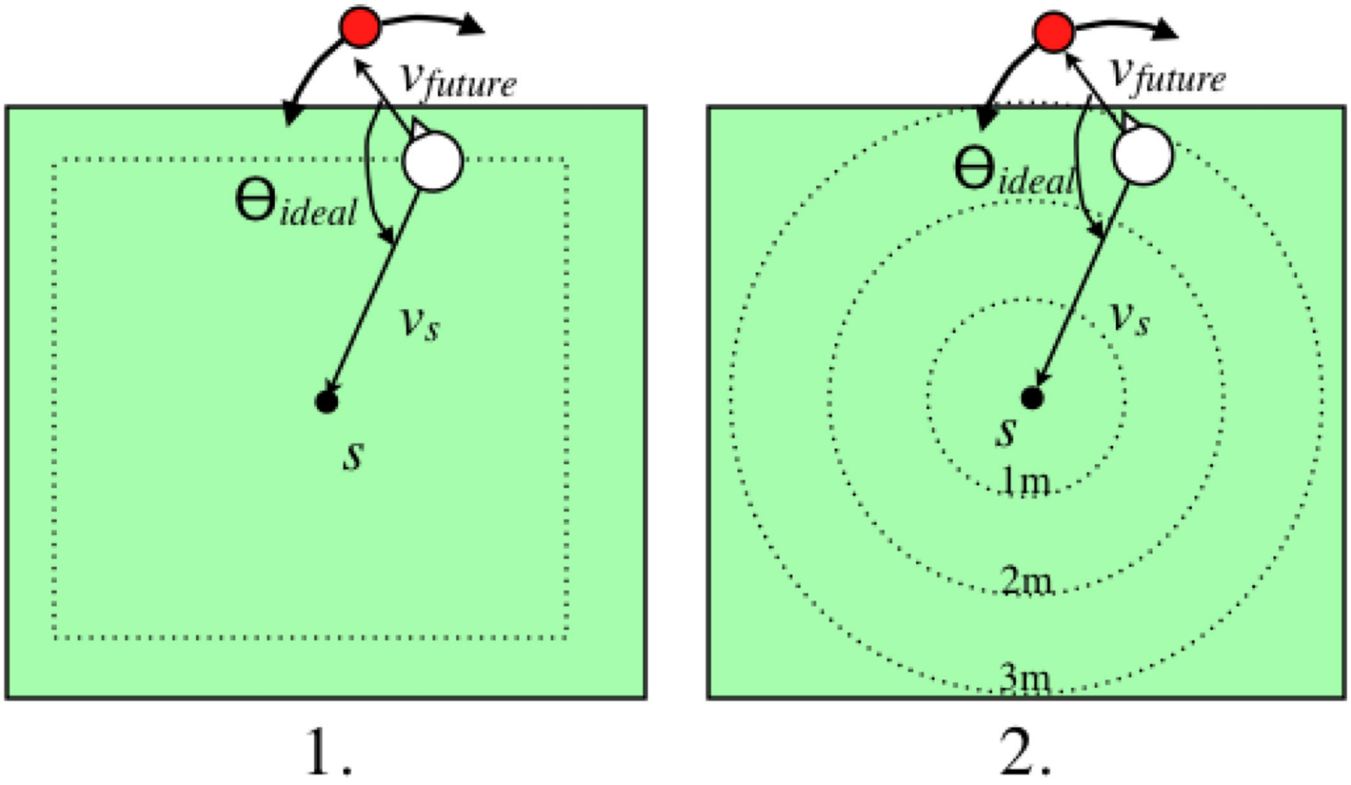

Fig. 3.

Version 1. Turn distractor on: The user is stopped by a distractor when he crosses a boundary near the edge of the tracked space. Turn distractor off: When the VE has rotated by θ'_{ideal} around the user. **Version 2**. Turn distractor on: The distractor appears based on a function of the user's distance to the center of the tracked space and the time since the previous distractor. Turn distractor off: When the VE has rotated a fraction of θ_{ideal} based on the distance of the user from the center of the tracked space.

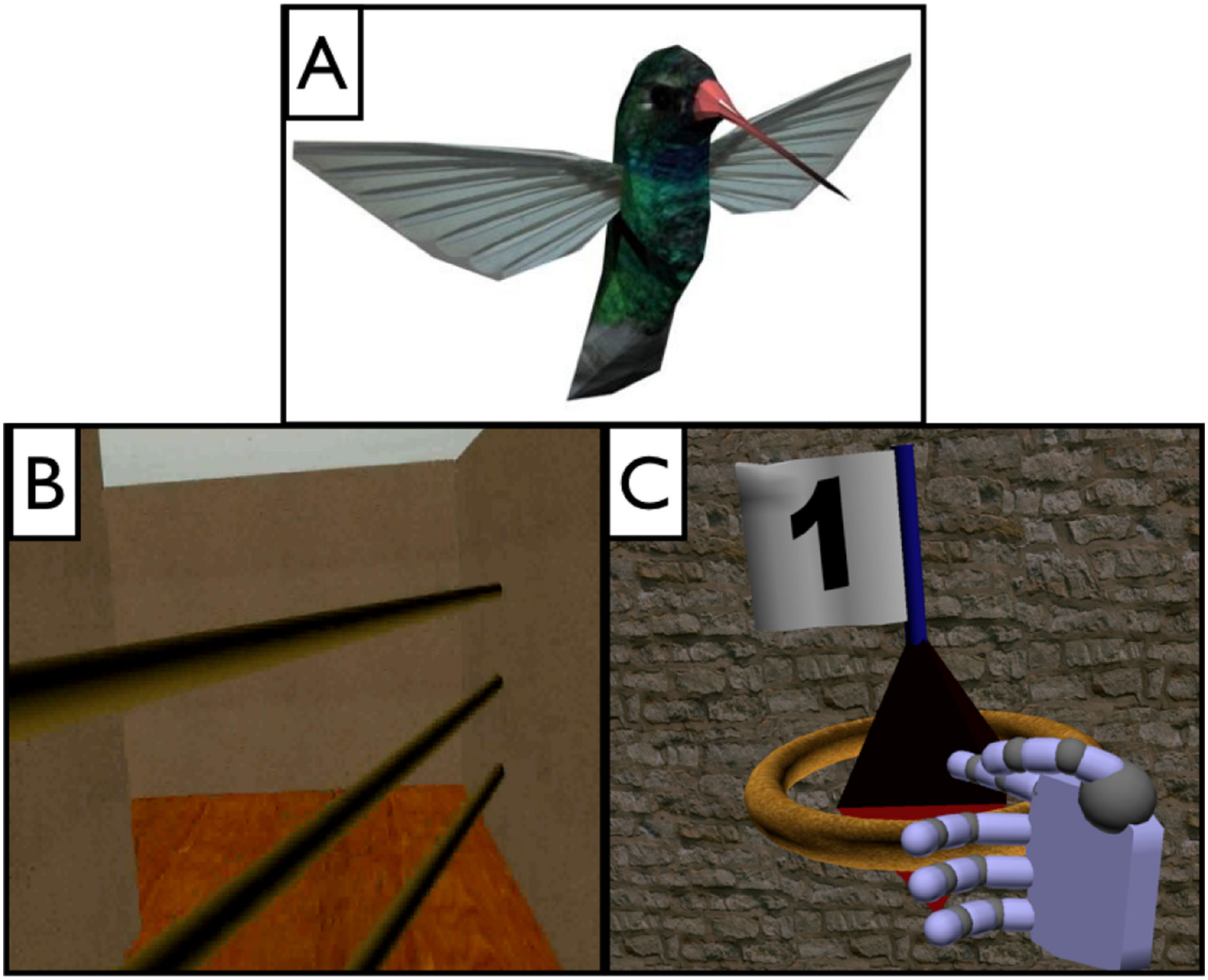

Fig. 4.

Screen shots of **A**: the hummingbird distractor, **B**: the horizontal bars used as deterrents, and **C**: the virtual avatar hand selecting a target.

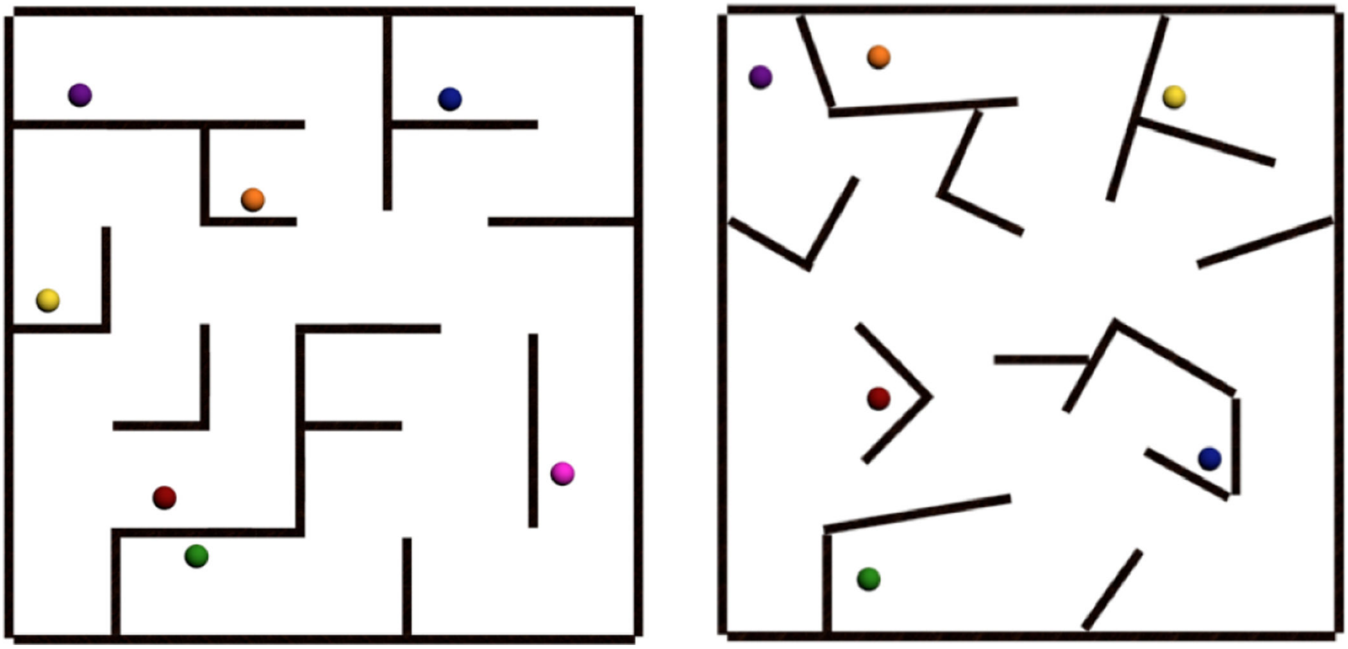

Fig. 5.

The 15.85m × 15.85m virtual mazes used in this study. Left: the maze used during the naive search with seven targets. Right: the maze used during the primed search with six targets. Participants started each maze in the bottom left corner.

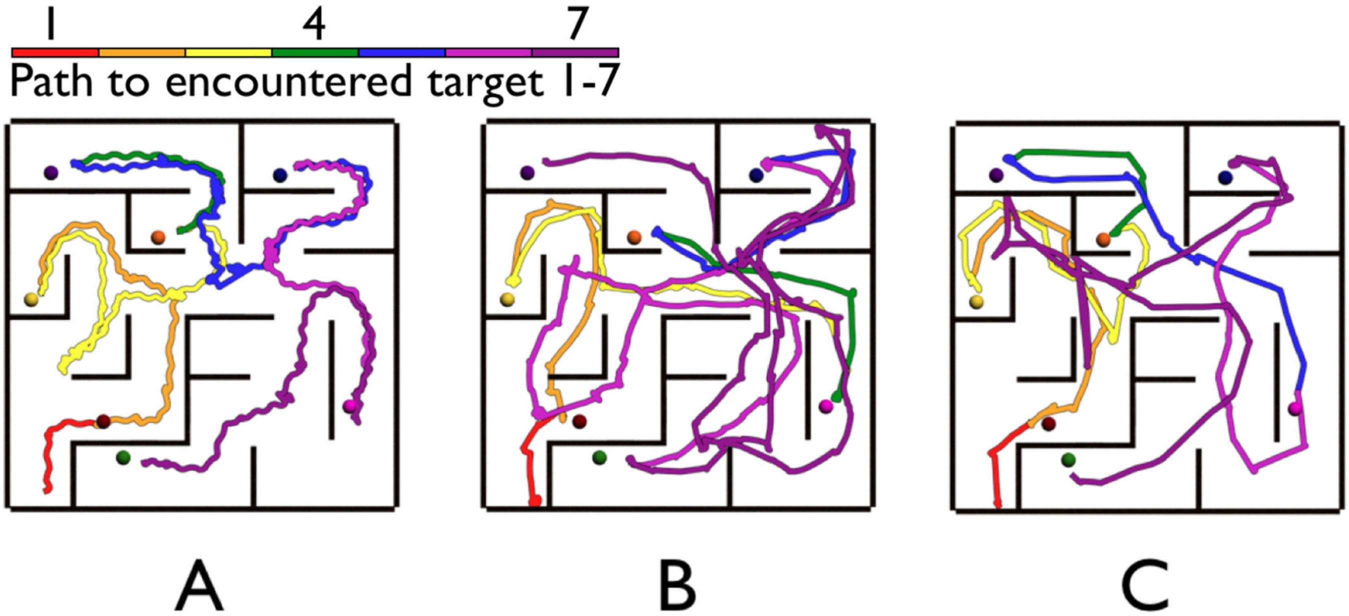

Fig. 6.
The virtual routes of three participants performing the naïve search, one using each of the three locomotion interfaces. The routes of the participant who traveled the median distance in each locomotion interface is displayed. **A**. RFED. **B**. WIP. **C**. JS.

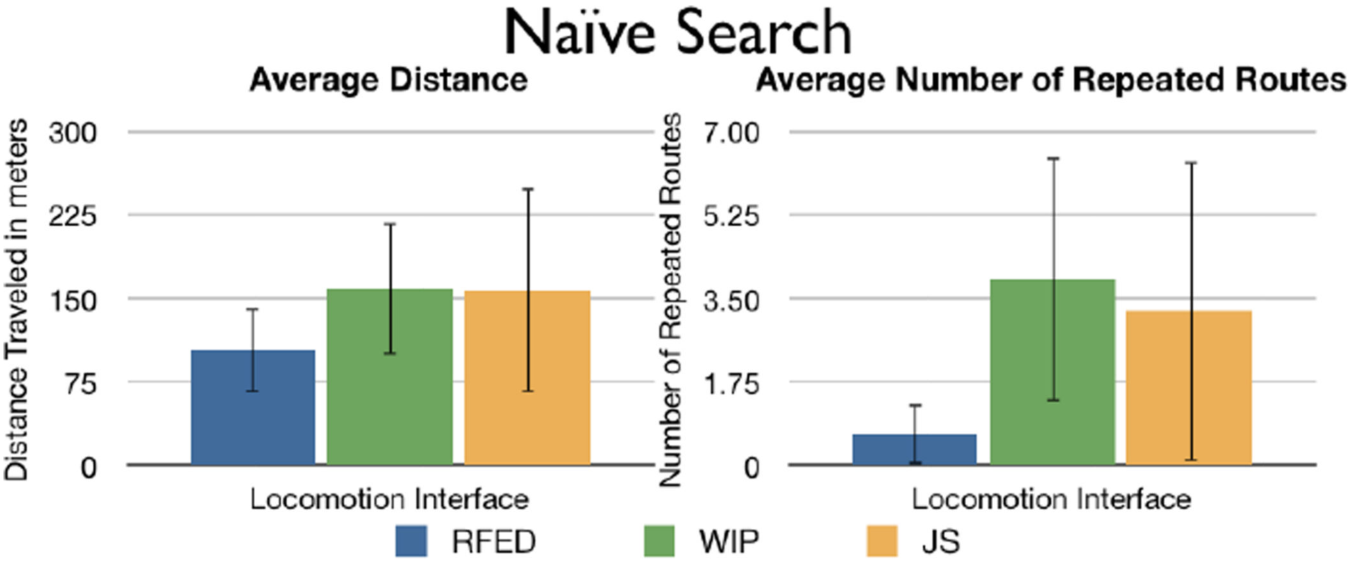

Fig. 7.

The average total distance traveled and the average number of repeated routes, by locomotion interface, when performing the naive search to find seven targets within the maze, with ±1 standard deviation error bars.

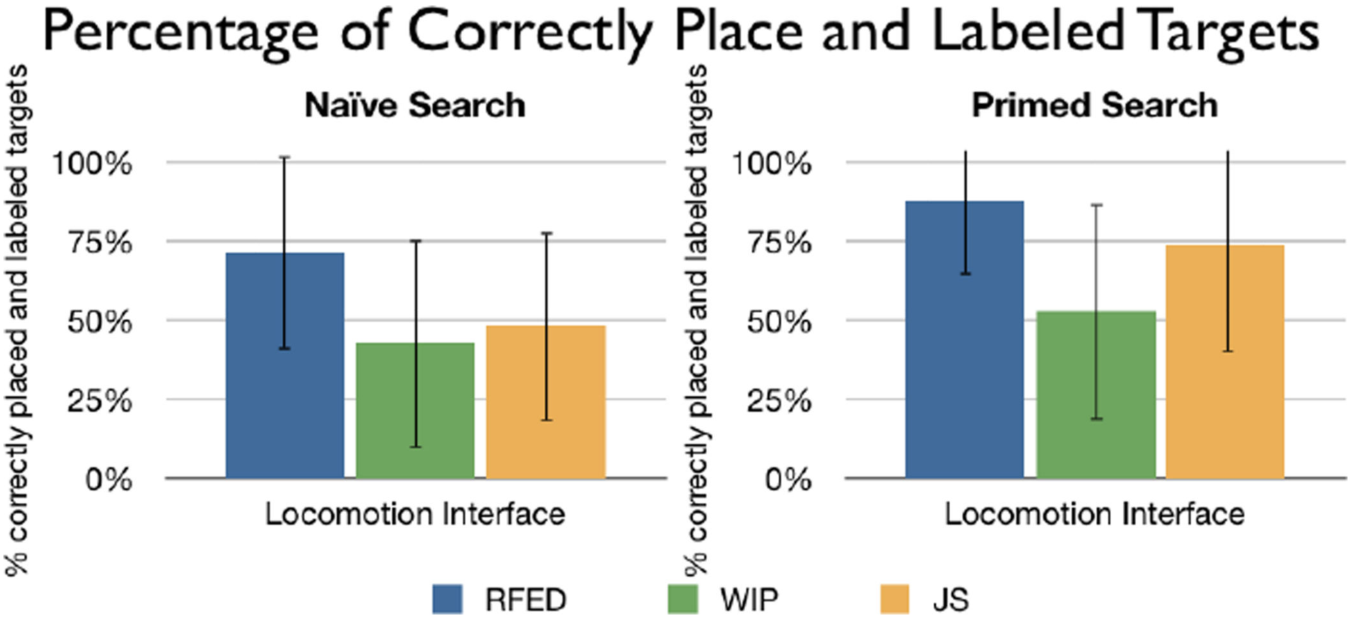

Fig. 8.

The average percentage of correctly placed and correctly labeled targets on paper maps after completing the naive and primed searches. ±1 standard deviation.

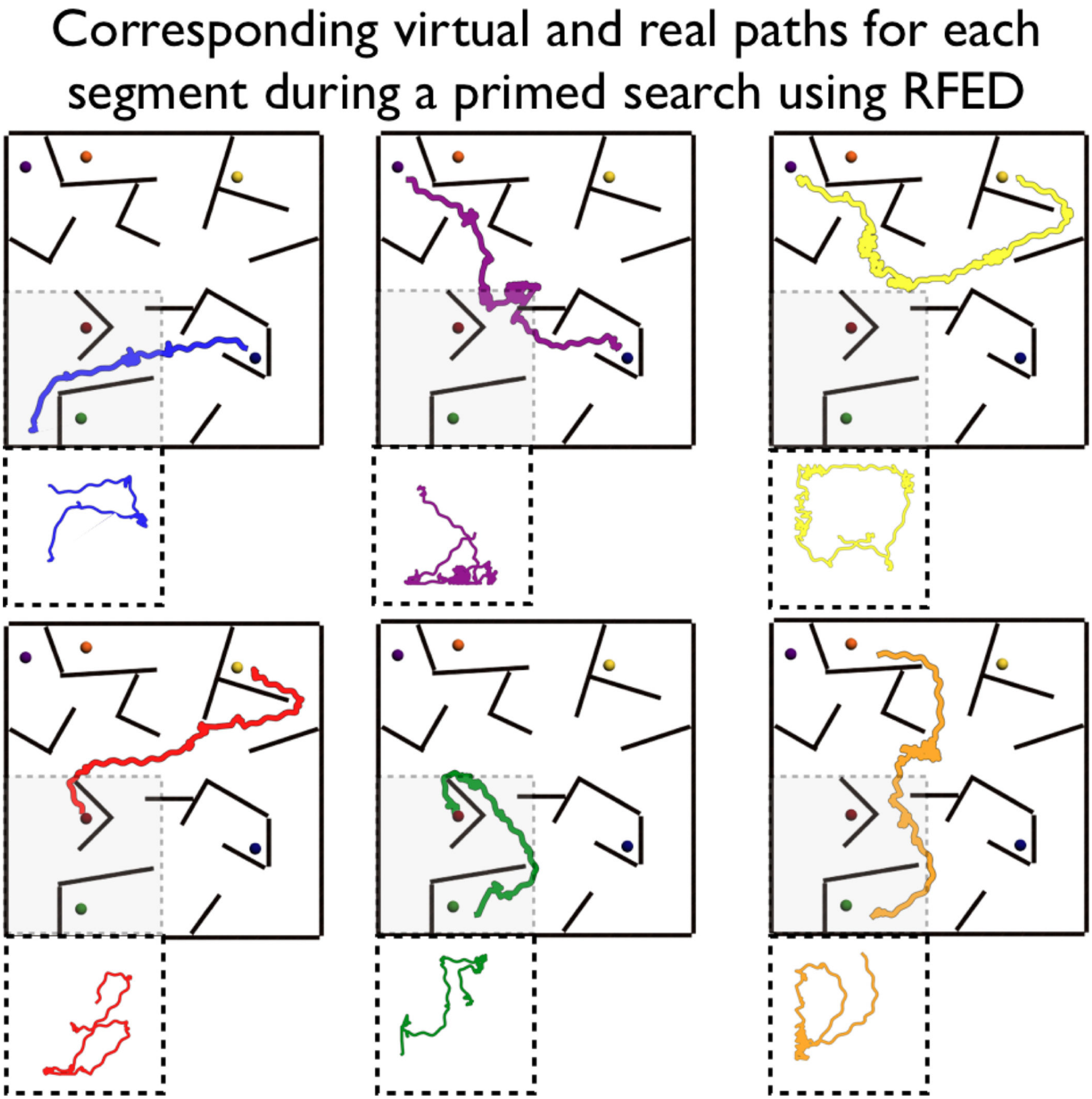
Corresponding virtual and real paths for each segment during a primed search using RFED

Fig. 9.

The virtual routes with corresponding real routes taken by a participant in the RFED condition during the primed search part of the experiment. Participants were really walking in one-quarter of the area of the VE. In the large boxes are the virtual routes, and in the small dashed line boxes are the corresponding real routes. Routes are displayed to scale.

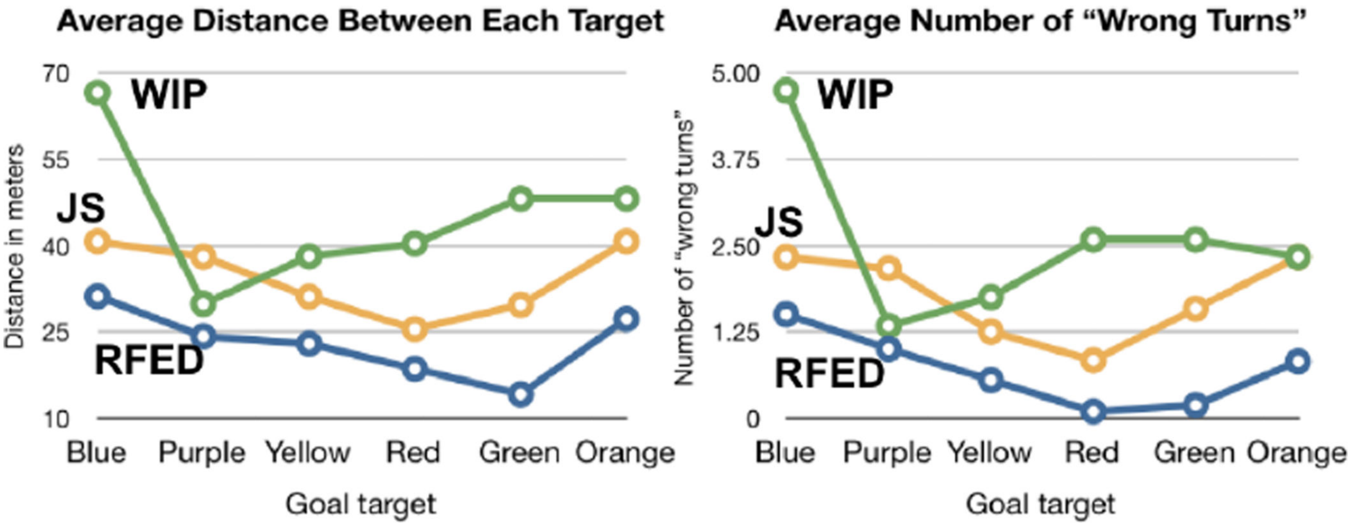

Fig. 10.

The average distance traveled between targets (in visited order) and the average number of "wrong turns", by locomotion interface when performing the primed search for each of the six targets within the maze.

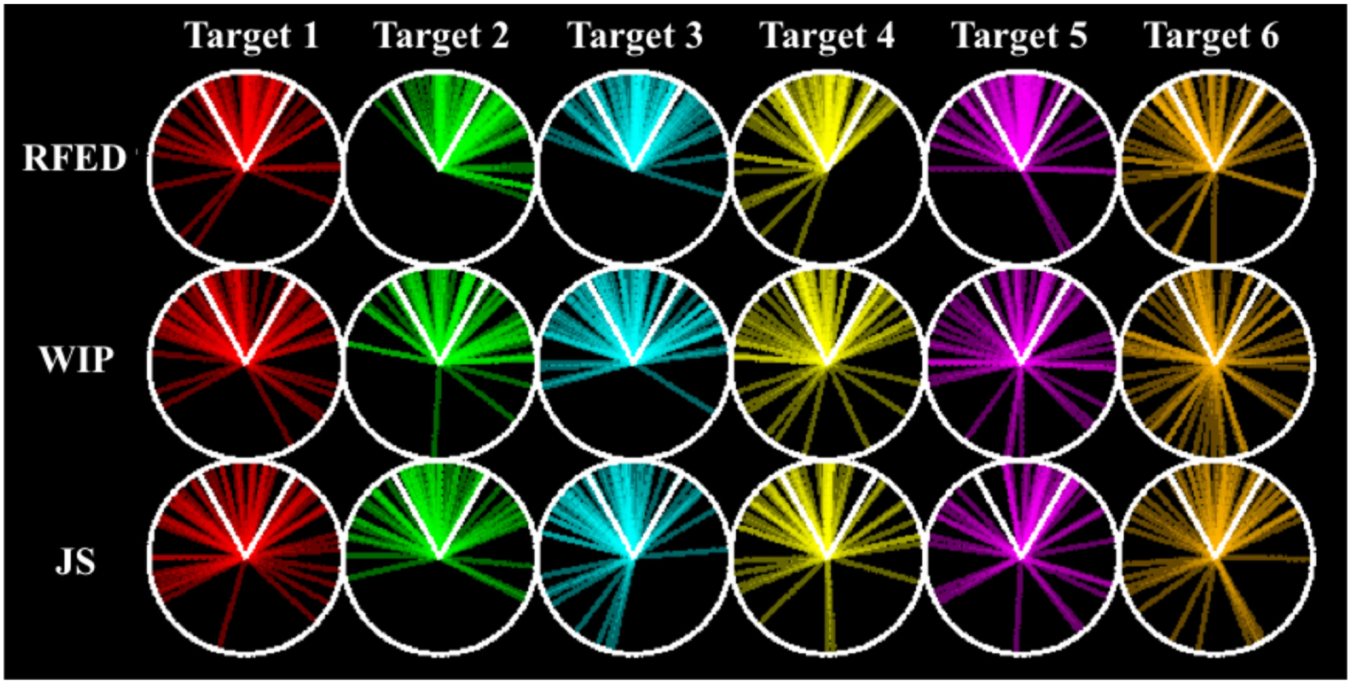

Fig. 11.

The pointing data for all participants to each of the six targets (columns) by each locomotion interface (row). The white lines denote ±30°, a standard real-world point error.

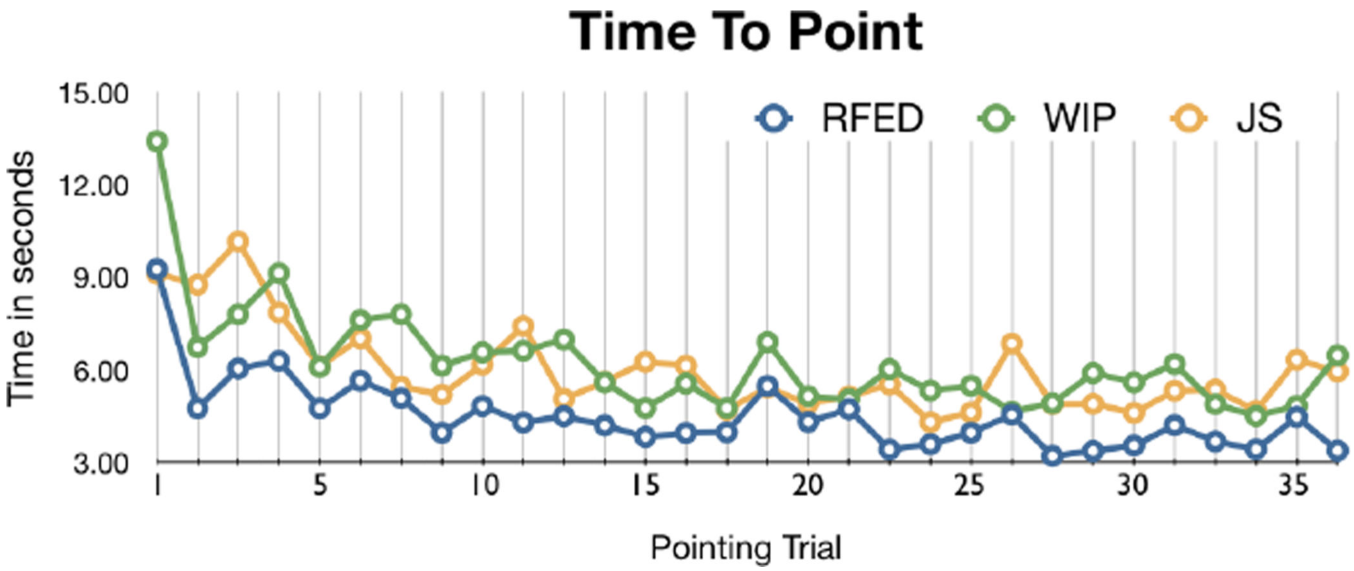

Fig. 12.

The average pointing time for each pointing trial by locomotion interface.

TABLE 1

Efficient Redirection Variables

Variables	
θ_{VE}	instantaneous magnitude and direction of VE rotation
v_{future}	unit vector of the user's immediate future direction from the user's current location
s	a real-world location toward which the user is steered
v_s	unit vector from user's real location to s
θ_{ideal}	ideal magnitude and direction of VE rotation to redirect user toward s

Note: a transformation consisting of a rotation and translation can convert between the real world and the virtual world.

TABLE 2

The distractor on and off values for distance and time used in [30] as determined by pilot experiments.

Distance from center (d)	Time since previous distractor (t)	percentage of θ_{ideal}
$< 1.5m$	$> 40s$	0.67
$< 2m$	$> 35s$	0.71
$< 2.5m$	$> 15s$	0.77
$< 3m$	$> 5s$	1.0
$< 3.25m$	$> 3s$	1.0

TABLE 3

The average VE size estimate and area underestimate by locomotion interface.

Locomotion Interface	**Dimension Estimate**	**Area Underestimate (%)**
RFED	15.0 *m* × 15.0 *m*	10%
WIP	10.5 *m* × 10.5 *m*	56%
JS	9.1 *m* × 9.1 *m*	67%
Actual	**15.85** *m* × **15.85** *m*	**0%**