NAVIGATING VIRTUAL ENVIRONMENTS AT SCALE

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of Master of Science in
Computer Science and Engineering

by
Jiwan Bhandari

Dr. Eelke Folmer / Thesis Advisor
August 2017
We recommend that the thesis prepared under our supervision by

JIWAN BHANDARI

Entitled

NAVIGATING VIRTUAL ENVIRONMENTS AT SCALE

be accepted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Eelke Folmer, Ph.D. – Advisor

Feng Yan, Ph.D. – Committee Member

Paul Macneilage, Ph.D. – Graduate School Representative

David Zeh, Ph.D. – Dean, Graduate School

August 2017
ABSTRACT

Using real walking for virtual exploration generally delivers most immersive VR experience, however, it is limited by the availability of physical tracking space. Virtual spaces beyond the size of tracked space are inaccessible. Limiting virtual world to match the shape and size of physical space is regressive and usually not possible. As such, to explore virtual space inaccessible with real walking, alternative free form navigation techniques like Walking-in-place, teleportation, joystick etc. should be used in-conjunction with real walking input or just by themselves. We present studies on two alternative navigation techniques (Legomotion & Dash) that enable immersive, efficient and scaleable virtual navigation.

Combining real walking with techniques like joystick input involves repeated transition from leg-based navigation to hand-based navigation that breaks immersion and could potentially lead to users abandoning real walking input as hand based navigation is often faster, easier and less tiring. Legomotion is a hybrid handsfree locomotion technique that lets users switch between real walking and walking-in-place input.

Regular teleportation, because it discontinuously translates the viewpoint does not generate any optical flow, limits path integration, e.g., estimating the total distance travelled, which can cause spatial disorientation. Dash is a modified teleportation technique that quickly but continuously displaces the viewpoint as to retain some optical flow and significantly improves path integration while it does not increase cybersickness.
ACKNOWLEDGEMENTS

I owe a huge debt of gratitude to my advisor, Dr. Eelke Folmer for all his support and guidance. His encouragement and feedback has been very crucial to me. I would like to thank the members of my committee Dr. Feng Yan and Dr. Paul Macneilage. I want to thank my parents, family members and friends for their love and support.
# TABLE OF CONTENTS

Abstract ................................................................. i
Acknowledgements ....................................................... ii
Table of Contents ....................................................... iii
List of Tables ............................................................ iv
List of Figures ........................................................... v

1 Introduction
   1.1 Legomotion ...................................................... 3
   1.2 Dash .............................................................. 5

2 Background and Related Work 8
   2.1 Legomotion - Walking-based scalable locomotion ............... 8
   2.2 Dash - VR Teleportation without spatial disorientation ........ 10

3 Legomotion: Walking-based Scaleable Locomotion 13
   3.1 Design of Legomotion ........................................... 13
   3.2 Implementation .................................................. 14
   3.3 Methodology ..................................................... 15
      3.3.1 Instrumentation ............................................. 16
      3.3.2 Virtual Environment ....................................... 16
      3.3.3 Procedure .................................................. 19
      3.3.4 Data collection ............................................ 21
      3.3.5 Participants ............................................... 21
   3.4 Results .......................................................... 22
   3.5 Discussion ...................................................... 24

4 Dash - VR Teleportation without spatial disorientation 29
   4.1 Experiment Design ............................................... 29
      4.1.1 Instrumentation ............................................. 30
      4.1.2 Virtual Environment & Tasks .............................. 31
      4.1.3 Implementation ............................................ 34
      4.1.4 Procedure .................................................. 35
      4.1.5 Measures ................................................... 37
      4.1.6 Participants ............................................... 38
   4.2 Results .......................................................... 38
   4.3 Discussion & Limitations ....................................... 40

5 Conclusion
   5.0.1 Legomotion ................................................... 43
   5.0.2 Dash .......................................................... 43
   5.0.3 Future Work .................................................. 44

Bibliography .......................................................... 46
LIST OF TABLES

3.1 Total time traveled averaged per participant in seconds. Standard deviation listed between parenthesis. Method is either WIP or controller. ................................................................. 22

3.2 Total distance traveled averaged per participant in meters. Standard deviation listed between parenthesis. Method is either WIP or controller. ................................................................. 22

4.1 Quantitative results. ................................................................. 39
LIST OF FIGURES

1.1 Left: walking input as enabled using positional tracking, is considered most immersive for VR navigation, but it is bounded by available tracking space. Right: Legomotion lets users switch to walking-in-place to navigate beyond the confines of available tracking space. ................................................................. 4

1.2 Teleportation discontinuously translates the user’s viewpoint over a distance (A & B). The absence of optical flow reduces VR sickness, but also limits the users’ ability to perform path integration, i.e., estimating the distance travelled, which can lead to spatial disorientation. Dash merges teleportation with regular locomotion by quickly and continuously moving the user to a destination, which generates the optical flow that allows for path integration. ................. 6

3.1 Finite state machine listing the 3 defined input states and transitions between them for legomotion to avoid false input. ............... 15

3.2 Birds eye view showing the VE used for the study, far waypoints are rendered in red and lines connect each consecutive waypoint. .. 17

3.3 First person view showing the navigation task showing a far waypoint (Left) and a visual grid appears when the user runs into the border of the tracking space (right) which requires the participant to switch to an artificial locomotion method. ......................... 18

3.4 Combined graph showing discomfort scores (on a scale 0-10) in the first 2 columns. The remaining columns show Likert scores (scale 1-5) for each navigation method based on criteria: efficiency, learnability, errors, likeability and transition ease. ......................... 25
3.5 Ranking of navigation methods where each number states how many participants preferred that navigation method for each usability criterion.

4.1 Virtual environment used for the user study. Left: users have to teleport to the green circle (red starting circle is visible) Right: after teleporting, a sign indicates to the participant to point back to their start location.

4.2 The two triangle completion tasks used in our study: left: 1-teleport, right: 2-teleport. S=start location, ∈=1st teleport location, ∈=2nd teleport location and E is the user’s estimate of S. V3 is the vector between S and E. With optimal path integration, V3=0 and V1 = V2.

4.3 Blocks for each trial for each group.

4.4 Ranking of teleportation methods on four criteria.
CHAPTER 1
INTRODUCTION

Virtual reality has existed for several decades mostly as a niche and fringe technology but lately virtual reality is making its way to the main stream [47]. The latest resurgence in the interest in virtual reality been fostered by years of innovation in high resolution displays and computing power. Furthermore, advances in motion tracking system over the last decade have enabled 6DOF input devices that provide rich interaction opportunities based on human gestures. In the realm of virtual reality, motion tracking controls/consoles have enabled rich set of immersive interaction experiences. Exploring virtual environment using real world walking is one such affordance of positionally tracked VR consoles.

Real walking based virtual locomotion have several benefits. Real walking based VR navigation generally delivers most immersive experience because virtual world action and events correspond to that of real world[49, 59]. As, walking comes naturally to human beings using real walking for virtual navigation imposes no cognitive load. However, real walking input as form of virtual navigation is limited by the availability of size and shape of the tracking space. Any reasonable virtual reality experience is bound to happen in a virtual world that is usually larger than commonly available in- door tracking space. Limiting the shape and size of the virtual world to match the available tracking space is not just regressive in terms of VR experience but also impossible for many application. Thus, in order to navigate beyond what’s feasible using real walking, we need to resolve to alternative navigation techniques like teleportation, walking-in-place, scale walk, joystick etc.

Teleportation is the most widely adopted technique in virtual reality applica-
Teleportation allows free-form omni-directional navigation and its usage doesn’t involve much cognitive burden. The only requisite for adopting teleportation is the ability to precisely point in virtual environment, thus it can be adopted on low-end VR setup like Google cardboard [2] as well as high-end VR setups like Vive [1]. In regular teleportation (point-and-jump), since the virtual viewpoint gets translated discontinuously there is no optical flow during the transition. The absence of optical flow, though, beneficial in terms of VR sickness, hurts spatial awareness and path integration ability. We did a comparative study on regular point-and-jump teleportation and modified teleportation technique (Dash), that doesn’t discontinuously translate the viewpoint thereby preserving some optical flow, to study the effects of limited optical flow in path integration ability.

Walking-in-place (WIP) [58] is a virtual navigation technique based on inertial sensing. Data from the IMU unit is taken as the users perform in-place step like movements. These steps are then used as continuous input to VR to translate the viewpoint in the direction of the gaze. WIP has been found to be more immersive form of input as it closely approximates real walking. Recent studies have found conjunction of head gestures (tilt) with WIP enables omni-directional navigation [57]. We propose a scaleable navigation technique (Legomotion) that couples WIP with real walking. We hypothesize that Legomotion offers more immersive navigation as compared to the combination of joystick input and real walking.

In the following, we will first introduce each technique in detail and in the rest of this symposium we expound further on the two projects on virtual navigation. Either of these project is a research study one of the aforementioned two alternative navigation techniques, their adaption, efficacy, efficiency, path integration and VR sickness. The research work on both projects culminated into two papers that have
been submitted to VRST 2017 conference.

1.1 Legomotion

VR is considered most immersive when it closely mimics interaction with the real world. Because walking is most natural to humans, using real walking input for VR navigation, generally delivers the most natural and immersive experiences [49, 59], while it reduces cognitive load [70] and minimizes the occurrence of VR sickness [34, 29]. Unfortunately, virtual locomotion using physical walking is bounded by the size of available positional tracking space [44].

Current consumer VR platforms support reasonable tracking areas, e.g., 15x15 feet. In practice, because these systems are installed in home environments, the available tracking space is limited by the available space in a living room or office, with many users freeing up only enough space to meet the minimum tracking space requirements. Because of these space constraints, some VR games limit gameplay to the available tracking space or they involve no navigation at all by offering an on-rails or wave-shooter like experiences [38]. It has been argued that such experiences offer limited engagement and lower presence[52].

Typically, the virtual environment that users can explore exceeds the boundaries of available physical tracking space and a different solution needs to be used.

Currently, to navigate beyond the confines of available tracking space, many VR applications require users to switch to an alternative locomotion technique (ALT) to reach destinations that they cannot walk to. ALTs are activated by the user using their controller, and include popular techniques, such as teleportation, full
Because walking is most natural to us, a possible solution to the virtual locomotion problem could be in the usage of locomotion techniques that resemble the human gait and which generate all or partial proprioceptive/vestibular afferents associated with real walking. Most notable walking-like locomotion techniques include treadmill walking [63] and walking-in-place (WIP) [54, 58]. Omnidirectional treadmills [23] enable locomotion at scale. A major limitation besides issues with presence and limited commercial availability [67] is that treadmills cannot be combined with positional tracking input. Because users are strapped into a harness, it impedes certain movements, such as kneeling down to pick up an object, which reduces presence. WIP closely mimics walking, e.g., users provide step-like motions while remaining stationary [54] and it has found to offer a higher presence.
than controller input [49, 45]. Because WIP can be implemented using cameras [66] or inertial sensors [58] it can be easily integrated in current positional tracking systems, without restricting the users’ movements in any way.

This paper presents legomotion; a leg-based locomotion technique that lets users navigate beyond the confines of available tracking space by switching between WIP and positional tracking input. We evaluate the usability, presence and effectiveness of legomotion versus conventional controller based navigation and discuss limitations of both approaches.

1.2 Dash

Teleportation is considered a risk-free way of navigating in VR as it discontinuously translates a user to a specified destination; which doesn’t generate any optical flow or vection and thus reduces VR sickness [16]. However, it has been argued that teleportation breaks presence [16] since it lets users do something that doesn’t exist in real life and it usage can also disrupt intended gameplay [39].

Besides issues with presence, a current limitation of teleportation is that it can cause spatial disorientation [16, 11]. Humans generally navigate using a combination of two skills [60]: (1) path integration, where users update their current position based on an estimate of the direction and distance travelled obtained from visual, vestibular and proprioceptive senses and (2) landmark navigation, where users update their current position when a known landmark is identified [35].

When exploring an unfamiliar environment, humans rely entirely on path integration but build a cognitive map by observing landmarks [50]. Optical flow and
Figure 1.2: Teleportation discontinuously translates the user’s viewpoint over a distance (A & B). The absence of optical flow reduces VR sickness, but also limits the users’ ability to perform path integration, i.e., estimating the distance travelled, which can lead to spatial disorientation. Dash merges teleportation with regular locomotion by quickly and continuously moving the user to a destination, which generates the optical flow that allows for path integration.

vestibular and proprioceptive feedback provide powerful cues about self-motion [12] and are essential for effective path integration [60]. Though path integration is possible without vision it then relies solely on vestibular/proprioceptive cues [36].

Path integration while using teleportation as a VR locomotion method is limited, as users typically stand or sit still so there are no proprioceptive cues, while the discontinuous translation doesn’t generate any optical flow. Though heading can be acquired using vestibular feedback, for assessing the distance travelled, a user must rely entirely on a change in the distance perception of landmarks. Landmark perception can be difficult in environments where they are sparse or which contain lots of similar landmarks. A complicating factor is also that the field of view (FOV) of consumer VR headsets (up to 110°) is considerably smaller than the human FOV (i.e., up to 180°) –which significantly impedes landmark perception.

To allow for path integration and to minimize spatial disorientation, we evaluate Dash; a modified teleportation method that instead of a discontinuous translation uses a quick but continuous translation of the viewpoint to retain some optical
flow.
CHAPTER 2
BACKGROUND AND RELATED WORK

2.1 Legomotion - Walking-based scalable locomotion

Gait negation devices, such as omnidirectional treadmills [28, 7], enable navigation at scale. A limitation of treadmills is that they don’t support a very natural gait and walking input is thus perceived as a form of skating [67], which is detrimental to presence. To avoid users falling, users require to be strapped in a harness that is suspended from an overhead boom or hold onto a railing system, but this impedes certain movements, for example, kneeling down and picking up an object. Given that positional tracking is standard on all PC VR platforms and emerging on mobile VR platforms [5], we provide an overview of locomotion techniques that can be integrated with positional tracking systems.

Currently, the two most widely used mechanisms to enable virtual locomotion at scale – and that can be used in conjunction with positional tracking – are teleportation and vehicle movement [39, 52], but various other innovative locomotion mechanisms have been developed as well. Teleportation allows users to instantly move to a location that is selected by the user by pointing their controller or by stepping into a map that is rendered at the user’s feet [33]. It is a risk-free way of moving in VR as it doesn’t involve any vection, and is therefore is less likely to cause VR sickness [16]. It has been argued that since it lets users do something that doesn’t exist in real life, it has a low presence [16]. Teleportation also causes user disorientation [16] and significantly alters intended gameplay [39]. Though many VR games launch with teleportation as the default artificial locomotion method, there seems to be an active community of gamers that develop mods for these
games that add full locomotion using a controller [19].

Vehicle movement has users enter a vehicle or platform to move larger distances. The vehicle is either controlled by the user using their controller or their gaze or it provides an on-rails experience. On-rails experiences offer limited interaction and presence and vehicle movement is more likely to induce VR sickness, as users have no control over the direction of movement [38].

Redirected walking techniques [44] apply gains to viewpoint rotations and translations to give users the impression that they are walking in a straight line, while they actually walk in a circle. Though they offer a high presence, they require a large tracking space (> 22 meters) [51] to be imperceptible to the user, which not only exceeds the current tracking limitations of current consumer VR systems but also the available space in a typical home. Redirection techniques not only increase cognitive load [20] but also require knowing the user’s intended path to work effectively and to maintain a high presence [42]. As a result, they limit locomotion speed and don’t support real-time VR applications, such as games, where users frequently turn [61].

Alternative approaches [65] rotate the virtual world by 180° when the user approaches the tracking space boundary. Doing so forces users to turn around to continue their intended path, but such system-initiated interruptions are considered to break immersion [27] and may cause user disorientation. This approach doesn’t seem very effective for small tracking spaces [65]. The scaling factor of navigation [64] can also be modified to more optimally use limited tracking space, though it may break presence as high scaling values can cause user discomfort and VR sickness.
Walking-in-place enables handsfree VR navigation by closely mimicking walking, i.e., users provide step-like motions while remaining stationary [54] while they steer using gaze. Compared to controller input, studies have found walking-in-place to have a higher presence [49], and to allow for better spatial orientation [32].

A few approaches implement the partial gait of walking, e.g., head bobbing [55] and arm swinging [41]. Where earlier walking-in-place implementations relied on wearable magnetic sensors [25] or bulky sensors [15], recent walking-in-place implementations use cameras [66] or inertial sensors [58]. Given that cameras are central to positional tracking systems, walking-in-place can be easily integrated with positional tracking to enable virtual locomotion at scale. Yan et al. [69] has explored the use of WIP for 3D cave navigation, but their study doesn’t perform an evaluation of WIP when it is combined with positional tracking.

Most closely related to Legomotion are the following works. Jumper [14] lets users physically jump as to traverse a large virtual distance that is selected using their gaze. Jumper was designed to be used in conjunction with positional tracking input. RIPmotion [53] is a recent a demo for the HTC Vive that lets users switch between walking and running-in-place to navigate beyond the confines of available tracking space. No results from user studies are reported or comparisons made to other locomotion methods, which is what this paper does.

2.2 Dash - Teleportation without spatial disorientation

Being able to navigate beyond the confines of available tracking space while minimizing VR sickness, cost and maintaining a high presence is considered a major
barrier for the mass adoption of VR [38]. Directional movement using a controller (often known as full locomotion) has found to induce VR sickness [17, 59], and therefore the two most widely used mechanisms to enable virtual locomotion at scale –and that can be integrated with existing positional tracking systems without impeding movement– are vehicle movement and teleportation [39]. Vehicle movement has users enter a vehicle or platform to move larger distances. The vehicle is either controlled by the user using their controller or their gaze or it provides an on-rails experience. On-rails experiences offer limited interaction and presence and are more likely to induce VR sickness, as users have no control over the direction of movement [38]. Though many VR games launch with teleportation as the default artificial locomotion method, there is an active community of gamers that develop mods that add directional movement using a controller to these games [19].

Bowman et al. [16] was the first study to discover that teleportation increases spatial disorientation. Spatial awareness was tested using a visual search task, e.g., users first cognitively mapped a space that contained different colored cubes by freely moving around. Then users were automatically moved from one location to another using a different velocity and then had to find a cube of a particular color. This study explored four different velocities (slow, fast, S-curve, infinite) and found that using an infinite velocity (e.g. teleportation) significantly increased target search time but no difference between the other velocities was found. Bakker et al. [11] conducted a similar experiment with participants cognitively mapping various rooms using teleportation or controller input and then asking them to point to a specific object. This study also found that the use of teleportation leads to a worse spatial mapping performance. Cliburn et al. [21] used a similar experimental setup but used a desktop computer instead of an HMD and found that when
given a map teleportation users are faster at object recollection than users using a keyboard for navigation.

A few approaches have aimed to improve teleportation. Laviola [33] presents a modification that requires users to step into a location on a map that is rendered at their feet in order to teleport to that location. Freitag et al [27] presents a teleportation mechanism where users have to walk through a portal that appears behind them in order to teleport as to optimize the usage of limited tracking space. Point and teleport [18] allows users to specify their post-teleport orientation.

Most closely related to Dash is the following. Jumper [14] is a hands-free form of teleportation on PC VR platforms where users physically jump forward to a location specified by their gaze. The optical flown shown during this jump is similar to Dash but it uses a variable velocity. This study investigated spatial awareness but did evaluate path integration or its effect on VR sickness (though because users have to physically jump forward it does generate vestibular/proprioceptive afferent that may minimize VR sickness). Raw Data [6] is a popular VR game for the HTC Vive/Oculus Rift that implements a teleportation mechanisms very similar to Dash. However, Raw Data uses a fixed amount of time for the teleport transition regardless of the distance traveled, which in our opinion impedes spatial integration ability. Though this teleportation mechanism is widely praised there are currently no studies that analyze its effect on spatial orientation or VR sickness, which this paper does.
CHAPTER 3

LEGOMOTION: WALKING-BASED SCALEABLE LOCOMOTION

3.1 Design of Legomotion

WIP [48] approximates real walking input in terms of presence [49] and efficiency [45]. Locomotion techniques generally allow users to control velocity and direction. Using real walking, users steer with their legs but also control velocity. We argue that switching between real walking and WIP input may offer a lower cognitive load because the same limbs are involved. Because this transition is more seamless than having to switch to a different limb (i.e., controller input) it may offer a higher presence. Extensive usage of a controller can lead to arm fatigue or “gorilla arm” syndrome. A significant benefit of Legomotion is that it leaves the user’s hands free, allowing users to lower their controllers or use them for other purposes (like shooting), which can improve efficiency.

To guarantee collision free navigation, consumer VR systems (i.e., HTC Vive’s Chaperone) will render a visual grid that indicates the size and location of the available tracking space. This grid becomes visible when the user approaches the boundary of the tracking space. Legomotion supports two usages:

- **As needed.** A user navigates freely using walking input (see fig 1.2 left). When approaching the boundary of the available tracking space the user transitions to WIP if they want to continue in the direction they were heading.
- **As anticipated.** If a user is aware of their current position in the available tracking space, they can assess whether their destination lies outside of it and immediately switch to WIP to avoid having to transition when reaching the edge.
of the tracking space. This scenario predominantly uses WIP to traverse larger
distances while positional tracking is used for detailed/precise navigation.

For both cases, users can switch back to walking input at any time (tracking space permitting). Users control velocity of navigation using their step frequency and steer either using their gaze (when using WIP) or using their legs (when using real walking).

### 3.2 Implementation

To implement legomotion, we assume a 3D positional tracking system and a real-time step detection algorithm that uses Y-positional or inertial sensor data. An implementation challenge here is that users should only be in one of three states {stationary, walking, WIP}, but positional movement (\(pos\)) can also be detected as a WIP step (\(step\)) leading to input from both states and an extra virtual displacement. To avoid a double displacement, we check every sample duration of the step detection algorithm whether \(pos\) in the lateral plane has exceeded threshold \(\sigma\). If the user is walking \((pos > \sigma \land step = 0)\) and while WIP \((step \geq 1 \land pos < \sigma)\). Figure 3.1 defines valid transitions between the three states using a finite state machine to avoid false input.
3.3 Methodology

For our study, we decided to compare Legomotion to full locomotion rather than teleportation as it has been argued that it breaks presence [16] since it lets users do something that doesn’t exist in real life and it usage can also disrupt intended gameplay [39]. Teleportation also requires a motion sensing controller which isn’t widely available on mobile VR platforms yet. Though many VR games launch with teleportation as the default ALT, there is quite an active community of gamers that develop mods that add full locomotion to these games [19]. Full locomotion can also be considered a proxy for vehicle/cockpit locomotion that is also widely used in VR games. Because teleportation is instantaneous, it isn’t possible to compare performance or steering-accuracy and our evaluation would only be able to compare presence.
3.3.1 Instrumentation

We initially implemented legomotion using the HTC Vive headset, but we found that because this HMD is tethered it allows the user to feel where they are in the defined tracking space based on the cable tension. Because this could affect the performance of our user study, we explored an untethered approach using mobile depth sensing technology and mobile VR adapter. However, legomotion can be used with any VR platform that supports positional tracking. Google Tango is a 7" Android tablet that features integrated motion-tracking (using inertial sensing), area learning and depth perception using time-of-flight sensors. The depth map is 320x180 pixels and is updated with a frequency of 5Hz. Using the Durovis 7 adapter, Tango becomes a VR display offering a 960x1200 per-eye resolution at 60Hz with a 100° FOV. Though these specs are slightly lower than the HTC Vive (1080x1200,90Hz,110°FOV), the overall weight of this setup (795 grams) is similar to the HTC headset with cables. We replaced the strap of the Durovis 7 headset with a 3-point strap for more comfort and better support. For our controller input, we used the VR-PARK portable Bluetooth 3.0 controller that features a thumb stick.

3.3.2 Virtual Environment

Prior WIP studies [49, 62] have mostly included navigation tasks with users doing straight trajectories; which isn’t very realistic for many VR applications [56]. Using a navigation task that has users freely explore a VR environments makes it difficult to compare results between participants, therefore our navigation task had users follow a path that is defined using a series of waypoints (similar to Fernandes and Feiner [26]). We designed a virtual environment (VE) in Unity 5.0
Figure 3.2: Birds eye view showing the VE used for the study, far waypoints are rendered in red and lines connect each consecutive waypoint.

featuring a house and several trees. A $\delta > 1$ of 1.0 was used to make sure 1.0 meter displacement in the real world corresponds to a 1.0 meter viewpoint translation in the VE. To implement legomotion, Tango’s real world observed delta displacements are used to move the user’s avatar.

Real time step detection and translation to virtual motion was implemented using an algorithm presented in [58] and which was available as a Unity plugin called VR-step [8]. This plugin detects steps using inertial data, but we modified it to use the positional Y data, which slightly improved step detection accuracy. Authors of this plugin do not report an error rate, but experimental trials revealed a fairly high step detection accuracy (>98%). The virtual locomotion velocity of WIP depends on the step frequency. To minimize for any differences in performance, the maximum velocity for both WIP and joystick input was set to $2m/s$, which is reached with a step frequency of 1.4 steps a second (average human walking
Figure 3.3: First person view showing the navigation task showing a far waypoint (Left) and a visual grid appears when the user runs into the border of the tracking space (right) which requires the participant to switch to an artificial locomotion method.

speed/step frequency) or when the thumb-stick exceeds a specified threshold. For step frequencies below 1.4 the velocity is set to a value according to a linear transfer function specified in [58], where the velocity for joystick below the thumb-stick threshold is also linearly interpolated. The step detection threshold was set to the recommended values. Directional movement was implemented, such that the thumb-stick moves the user in the direction relative to their current gaze (e.g., if the thumb-stick is moved to the left, the user moves sideways, but forward moves the user in the direction of their gaze).

We implemented the FSM (see figure 3.1) for legomotion and to avoid false
input we used a value of $1.3cm$ for $\sigma$ that was determined experimentally. Experimental trials showed that transitioning between WIP and walking could be performed fairly smoothly, leading only to a minor drop in velocity, with no effect on latency. Because controller input doesn’t interfere with positional tracking, both types of input can be used simultaneously.

The size of tracking space affects how much walking or WIP/controller input is used. The HTC Vive’s tracking space ranges from $1.5x2.0$ to $3.5x3.5$ meters. Most tracking ranges are rectangular, but because navigation in any direction is equally likely, we used a square tracking space of $2.5x2.5$ meters. User studies were held in a large room with at least 1 meter of accessible space outside the defined tracking space. We used Tango’s area learning feature to map the visual features of the room and to define the dimensions of our tracking space. This information was stored on the Tango to boost the performance of its real-time depth sensing. With depth sensing the VE running in Unity we got a frame rate close to 60fps. The grid becomes visible when a user gets within $0.4m$ of the border and we modulate its alpha value.

### 3.3.3 Procedure

To force participants to transition between locomotion techniques, we defined two different types of waypoints:

- **Far** waypoints ($distance > 3m$) lie outside the tracking space and require WIP or controller input to be reached.
- **Near** waypoints ($distance = 1.3m$) lie within the tracking space and can be reached using walking input.
Figure 3.2 shows a birds eye view of the 30 far waypoints defined for our VE. Figure 3.3 shows the first person view of the navigation task. Only one waypoint is visible at a time and it is rendered as a tall and distinctively colored (magenta) column that is easily noticed and not obscured by other objects. When the participant collides with the waypoint, it disappears, a sound is played and the next waypoint becomes visible. Participants started their trial for a specific locomotion method by navigating to a number of green colored waypoints in the beginning so that they could adjust things like focus of the lenses, tension of the straps, etc.

Participants navigated to a far waypoint and then need to navigate to a predefined number (between 0 and 2) of near waypoints. Because we can’t predict the participant’s location in the available tracking space, a near waypoint is generated randomly at $1.25m$ radius distance from the participant but such that the participant can always walk to it. This waypoint is sometimes generated outside their current field-of-view to lure participants away from the edges of the tracking space.

After navigating to the near waypoints, the next far waypoint is generated and the sequence repeats. Participants were fitted with the Tango headset and we ensured the HMD was firmly attached to their head using the straps. We did not observe WIP to affect HMD fixation. For each navigation technique, a built-in tutorial explains how to use it and participants practiced 5 waypoints which included 2 far and 3 near waypoints to control for any learning effects. Each participant tests both locomotion combinations (e.g., walking+WIP and walking+controller) which were counterbalanced to control for order effects. Participants were placed in the center of the tracking space and asked to navigate to a total of 60 waypoints, which included 30 far and 30 near ones. We did not tell participants about far or near
waypoints and merely suggested participants use WIP/controller input when a waypoint was outside the tracking space. We use the same sequence of waypoints for both trials, with the location of the near waypoints varying.

3.3.4 Data collection

We decoupled the visual search from the navigation task, e.g., and we measure time and distance traveled to each waypoint only once the landmark is visible to the participant. If the lateral angle between the user’s gaze pointer and the landmark is less than 40° for 0.75 seconds, we assume that the user is looking at the landmark and we start measuring time and distance. For every type of waypoint, we record total time and distance distinguished by whether they used positional tracking or WIP/Controller input. Before the trial, basic demographic was collected using a questionnaire. Participants rested 10 minutes between trials while they took off the headset. After both trials users were asked to qualitatively rank each navigation method based on a number of criteria. The entire session took about 40 minutes.

3.3.5 Participants

22 participants were recruited of which 18 successfully completed our study (4 females, average age 25.44, SD=5.4). Four participants dropped out during the first trial due to VR sickness, but it seemed not specific to a locomotion method but a general adverse reaction to VR. A recent study [40] also found females to be more susceptible to VR sickness, which could have played a factor given that two of the
participants who dropped out were female. All participants had experience with navigating 3D environments using a joystick with three participants having lots of VR experience and the remaining 15 had experienced VR before (e.g., Google Cardboard or Oculus Rift). We considered our group of participants to be homogeneous regarding VR experience. None of the participants self-reported any non-correctable impairments in visual/audio perception or mobility limitations. The user study was IRB approved.

### 3.4 Results

<table>
<thead>
<tr>
<th></th>
<th>Legomotion</th>
<th>Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Far walking</strong></td>
<td>30.86 (24.1)</td>
<td>7.33 (8.5)</td>
</tr>
<tr>
<td><strong>Far method</strong></td>
<td>122.84 (23.6)</td>
<td>90.52 (9.7)</td>
</tr>
<tr>
<td><strong>Near walking</strong></td>
<td>24.03 (10.9)</td>
<td>10.18 (11.9)</td>
</tr>
<tr>
<td><strong>Near method</strong></td>
<td>6.92 (5.3)</td>
<td>17.14 (13.6)</td>
</tr>
<tr>
<td><strong>Total walking</strong></td>
<td>54.89 (31.7)</td>
<td>17.51 (18.6)</td>
</tr>
<tr>
<td><strong>Total method</strong></td>
<td>129.75 (23.6)</td>
<td>107.66 (17.2)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>184.65 (55.3)</strong></td>
<td><strong>125.17 (35.78)</strong></td>
</tr>
</tbody>
</table>

Table 3.1: Total time traveled averaged per participant in seconds. Standard deviation listed between parenthesis. Method is either WIP or controller.

<table>
<thead>
<tr>
<th></th>
<th>Legomotion</th>
<th>Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Far walking</strong></td>
<td>14.54 (11.8)</td>
<td>3.74 (4.8)</td>
</tr>
<tr>
<td><strong>Far method</strong></td>
<td>167.19 (13.8)</td>
<td>191.24 (18.9)</td>
</tr>
<tr>
<td><strong>Near walking</strong></td>
<td>15.77 (7.3)</td>
<td>6.24 (8.4)</td>
</tr>
<tr>
<td><strong>Near method</strong></td>
<td>15.99 (12.1)</td>
<td>48.02 (38.4)</td>
</tr>
<tr>
<td><strong>Total walking</strong></td>
<td>30.30 (17.8)</td>
<td>9.98 (12.2)</td>
</tr>
<tr>
<td><strong>Total method</strong></td>
<td>183.18 (24.4)</td>
<td>239.26 (45.4)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>213.48 (42.1)</strong></td>
<td><strong>249.23 (46.0)</strong></td>
</tr>
</tbody>
</table>

Table 3.2: Total distance traveled averaged per participant in meters. Standard deviation listed between parenthesis. Method is either WIP or controller.
A Shapiro-Wilks test was used to test for normality and a Grubb’s test found one outlier for controller data (time travelled) and we removed this datapoint. Table 3.1 and 3.2 list the results for total time and total distance travelled averaged per participant. A two-way repeated measures ANOVA was conducted to determine the effect of input method and waypoint type on total time travelled. There was a statistically significant two-way interaction between input method and waypoint type \((F_{1,17} = 18.09, p = .05)\). Total time traveled was statistically significantly different for locomotion method \((F_{1,17} = 24.10, p < .05)\) and waypoint type \((F_{1,17} = 453.38, p < .05)\). The same ANOVA found no statistically significant two-way interaction between input method and waypoint type \((F_{1,17} = .85, p = .369)\). Total distance travelled was statistically significant different for locomotion method \((F_{1,17} = 11.24, p < .05)\) and waypoint \((F_{1,17} = 797.89, p < .05)\).

For both types of waypoints, participants used a combination of walking and WIP/controller input, e.g., a participant can start walking to a far waypoint but then switch to WIP when they figure out they can’t reach the far waypoint using positional tracking. For near waypoints distance, legomotion participants used 50% walking input but controller input only used 11% walking input. For time the usage of walking input was 77% (legomotion) versus 37% (controller). A closer analysis of controller input revealed that 44% of participants stopped using positional tracking input after the first two far waypoints and the other participants’ usage of positional input dropped from 46% to 18% between the first 10 and the last 10 near waypoints.

**Qualitative feedback.** Similar to a study by Fernandes [26] participants rated their discomfort level on a scale of 1-10 after each trial. Average scores were 3.22 (2.6) for legomotion and 2.50 (2.6) for controller but a Mann-Whitney U test didn’t
detect a significant difference \( (U = 130, Z = 1.01, p = .313) \). Then, using a 5-point Likert scale, users evaluated each input method on 4 usability criteria: efficiency, accuracy, learnability and likability. Additionally, users were asked about the ease with which users could transition from walking to controller or WIP input. Figure 3.4 lists the results. A Mann-Whitney test found no statistical significant difference between input methods for any of the criteria, except for the transition criteria \( (U = 93, Z = 2.17, p < .05) \). After both trials participants ranked each input type on four usability criteria (efficiency, learnability, accuracy, likeability) and presence, where participants could select “No preference” as a 3rd option. Figure 3.5 lists the results, and a \( \chi^2 \) test found the rankings for efficiency, accuracy and presence to be significantly different \( (p < .05) \).

### 3.5 Discussion

Real walking input using positional tracking takes significantly more time than WIP or controller based navigation. Both quantitative and qualitative results found controller input to be significantly faster, despite that the velocities for both input techniques were set to the same value. A close analysis revealed that when using controller input, users use significantly less positional tracking input (nearly 3x less when looking at time) than when using legomotion. Our collected data revealed that over time participants transition to only using their controller to navigate to near waypoints, though these were generated within walking distance. Qualitative results confirm our hypothesis that switching between walking and WIP was considered more seamless than having to switch to a controller. Given that the navigation velocities of both input techniques were paired, the only explanation for users abandoning walking input when using a controller was that
Figure 3.4: Combined graph showing discomfort scores (on a scale 0-10) in the first 2 columns. The remaining columns show Likert scores (scale 1-5) for each navigation method based on criteria: efficiency, learnability, errors, likeability and transition ease.

having to frequently switch between leg and hand input was considered to more tedious than switching between different types of leg input. This assumption was further confirmed by a significantly higher transition ease score for legomotion. Legomotion only uses a single limb (ignoring steering) with walking and WIP having similar motor actions. Because controller based navigation uses legs and an arm, we believe switching between these has a significantly larger cognitive load associated than using legomotion.
Several participants said that they preferred controller input as they were most familiar with it. Though this result wasn’t significant, most users found controller input easiest to learn. Because WIP is also faster than real walking input, we observed a few participants to also use WIP to reach near waypoints but this occurred to a significantly smaller extent than was the case for controller input.

Though WIP generates vestibular/proprioceptive afferents, we did not detect any significant differences for VR sickness, e.g., discomfort scores, between both locomotion methods. Prior studies [29, 34] had found significantly higher VR sickness scores for using a joystick. Factors that could have played a role include that we used a high-fidelity platform (Tango) and that most participants were familiar with VR.
Though experimental conditions were the same, the total distance travelled to each type of waypoint varied significantly between navigation methods. This indicates a difference in accuracy, as more precise techniques generally lead to shorter distances travelled. Of specific interest is that participants ranked controller input to be the most accurate, but quantitative results revealed that participants travelled significantly longer distances using controller input than using legomotion (see table 1). These results corroborate results from a related study [56] that compared WIP and joystick trajectories, and found that WIP allows for better velocity manipulation and more precise steering. Though our difference isn’t huge (+16%) a visual analysis of trajectories showed that controller users swerved significantly more.

We implemented legomotion using the Google Tango platform, because it allows for untethered usage, but legomotion is not tied to this specific platform. We initially implemented legomotion using the HTC Vive which shows it is feasible to be implemented on any VR platform capable of positional tracking.

Our study only evaluated one particular navigation task, but we found some evidence for the notion [39] that the type of ALT offered in conjunction to positional tracking can significantly affect the usage of positional tracking input which then affects presence. This finding could have significant implications. If walking input is abandoned and users primarily use a controller, there is no real reason to remain standing. Within the VR community there have been extensive discussions on whether VR is best enjoyed standing up or sitting down [3]. For example, a seated VR experience can be maintained much longer and doesn’t require room size positional tracking, but it puts constraints on interacting with VR and what kind of VR experiences are possible. For example, picking up an object from the
floor is harder when sitting than when standing and developers end up having to implementing all sort of presence-breaking-shortcuts (object will appear on a table when dropped [43] ) to avoid such types of interaction. With the market and features of consumer VR still being fluid, what pose its users end up using will fundamentally affect the presence of its resulting VR experiences but it was important for our paper to identify what important role virtual locomotion has in it.
CHAPTER 4

DASH - VR TELEPORTATION WITHOUT SPATIAL DISORIENTATION

Teleportation is a popular locomotion technique that lets users navigate beyond the confines of limited available positional tracking space. Because it discontinuously translates the viewpoint, it is considered a safe locomotion method because it doesn’t generate any optical flow, and thus reduces the risk of vection induced VR sickness. Though the lack of optical flow minimizes VR sickness, it also limits path integration, e.g., estimating the total distance travelled, which can cause spatial disorientation. Dash; is a modified teleportation technique that quickly but continuously displaces the viewpoint and which retains some optical flow. We hypothesize that Dash significantly improves path integration while it doesn’t increase VR sickness.

4.1 Experiment Design

The objective of our study was to analyze whether a small amount of optical flow during teleportation enables path integration as to improve spatial orientation while not causing any VR sickness. Prior teleportation studies [16, 11, 21, 31] have all evaluated spatial disorientation using a visual search task where users first cognitively map a space containing landmarks or objects. The use of a visual search task is subject to a number of limitations as it can be argued that this task relies to a large extent on a user’s ability to successfully cognitively map a space and to a lesser extent on path integration—which is where differences in the type of locomotion method used matter. Using a visual search task that relies on an existing cognitive map doesn’t seem very realistic for many VR applications such as games, as its players predominantly explore new environments without a cog-
nitive map. Maps are not often used in VR as it breaks presence. For this study, we wanted to isolate the effect that optical flow has on path integration ability. To assess path integration, we use the “triangle completion” navigation task that was described in early work on path integration [36]. This task requires participants to travel from a start location to two waypoints, which are non-collinear with the start location. After arriving at the second waypoint, users have to turn around and try to navigate back to their start location.

4.1.1 Instrumentation

For this study, we decided to evaluate Dash on a mobile VR platform. Mobile VR platforms are considered to have limited interaction options [58], but since they only require a smartphone, they have a much larger potential to bring VR to the masses than PC VR approaches [24]. Mobile VR devices, like the Gear VR, have also eclipsed PC VR platforms in terms of sales [22]. Mobile VR platforms currently only support 3 degrees-of-freedom (DOF) input. It has been suggested that mobile VR platforms are more likely to induce VR sickness, because of the lack of apparent translational motion when users move their physical bodies. These specific constraints and popularity of this platform defines a relevant real-world context for Dash to investigate.

We implemented our experiment using Google Daydream; a VR platform for Android devices. This platform offers a 1080 x 960 pixels per-eye resolution at 60Hz with a 90° FOV using the Google Pixel smartphone (Snapdragon 821 2.15Ghz Quad-Core). For interaction, Daydream features a wireless inertial sensing 3DOF remote controller with a touchpad and several buttons. Because Daydream doesn’t
Figure 4.1: Virtual environment used for the user study. Left: users have to teleport to the green circle (red starting circle is visible) Right: after teleporting, a sign indicates to the participant to point back to their start location.

feature positional tracking, teleportation is a recommended for locomotion by Google’s VR guidelines [9]. With the exception of a few, the majority of Daydream apps use teleportation.

4.1.2 Virtual Environment & Tasks

We designed our virtual environment for our study using the Unity 3D engine (version 5.5.1) and Google VR SDK. To be able to exclusively focus on path integration, our virtual environment was designed to be devoid of any landmarks or distinguishable visual features. We used a scaling factor of 1:1 to model our virtual environment. The ground plane consisted of a desert like texture and the sky-box
featured uniformly distributed and shaped clouds, both generate optical flow but did not contain any specific visual features that would allow for landmark based navigation (see figure 4.1). We used a bright green circle as a way-point that the user needs to teleport to. Users can only teleport to the green circle so their exact location is always known.

At the start of each navigation task, a red circle is rendered at the participant’s feet (see figure 4.1:left) and a short audio feedback is given to make participants aware of their latest start location. After teleport, a message appears at the pointer that asks participants to point back towards their start location, i.e., the red circle, which has then disappeared (see figure 4.1:right). For our study, we deviate from the “triangle completion” navigation task [36] in two ways, e.g., early experiments found that pointing back to the start location after two teleports (2-teleport) with a random angle between them was already quite challenging. We decided to include an easier navigation task that only uses a single waypoint to teleport to (1-teleport). This task allows us to isolate path integration from the user’s ability to correctly rotate towards their start location as this does not rely on optical flow but only on vestibular cues.

Rather than having users navigate back to the start location, we have participants point to their start location, which we felt was more precise and also helps us maintain an exact location of the user. For the experiment, we used a predefined random sequence of 10 waypoints for 1-teleport and a random sequence of 20 waypoints for 2-teleport. The distance between consecutive waypoints was uniformly randomly varied between 5 - 11 meters as early experiments showed selecting a way point that was more than 13 meters away was difficult. For the 2-teleport task, we assume users are most likely to use teleportation to travel in one direc-
Figure 4.2: The two triangle completion tasks used in our study: left: 1-teleport, right: 2-teleport. S=start location, ①=1st teleport location, ②=2nd teleport location and E is the user’s estimate of S. V3 is the vector between S and E. With optimal path integration, V3=0 and V1 = V2.

tion, therefore we constrain the placement of the second waypoint to be within in a 180° FOV of the first way-point. We achieve this by splitting, the 180° range into 10 intervals of 18° and each second waypoint was placed at a random but unique value of \( q \times 18° \) with \( q \in [0, 10] \). Because we have participants switch between 1-teleport and 2-teleport for each teleportation method, the possibility of having participants memorize both sequences is low.

The Daydream controller is used to select the waypoint using a circular pointer (see figure 4.1), upon which the track-pad needs to be pressed to activate teleportation. Though the controller is only tracked in 3DOF, using it for selecting a destination to teleport to was not challenging. To avoid accidental input, when pointing back at the start location using the pointer, participants would need to hold down the track pad for 2 seconds. Many existing teleport implementations show a visual arch or line from the controller to the pointer, which helps in manipulating it. Because this visual cue conveys distance, we remove it, as to focus on path integration using optical flow only. Early trials showed participants were able to easily manipulate the pointer without this visual aid.
4.1.3 Implementation

To implement *Dash*, we modified an existing popular open source Unity teleportation asset [13]. Using regular teleportation, the player’s viewpoint is instantly translated in the next frame update. To implement *Dash* a number of design considerations had to be considered regarding the velocity as this affects the amount of optical flow and involves trade-offs between efficiency, spatial integration ability and VR sickness. Using a lower velocity generates longer-lasting optical flow but increases the risk of VR sickness due to vection while it also increases the time for users to arrive at their destination.

Our study did not evaluate what velocity was optimal for *Dash* because a closely related study on teleportation by Bowman et al. [16] had already explored several different velocities but did not detect a significant difference in spatial awareness between them. Because this study did not assess VR sickness or path integration, it made most sense to select a high velocity to limit optical flow exposure as to minimize VR sickness. We needed to consider how to adjust velocity during the translation. Raw Data [6] offers optical flow for a fixed amount of time irrespective of distance, which in our opinion doesn’t allow for path integration as users must gauge varying changes in velocity. Locomotion techniques like walking-in-place benefit from using an S-curve for ramping up and ramping down velocity depending on step frequency, as this closely models human locomotion and increases presence [58]. Bowman et al. [16] also explored the use of an S-transition but did not find a difference in spatial awareness with using a constant velocity. Mackinlay et al. [37] explored using a logarithmic curve but do not present results from a user study. Both the Oculus [10] and Google [4] VR design guidelines recommend using a fixed velocity for locomotion and to avoid acceleration.
Using a number of pilot studies, we experimented with different velocities to provide a better insight into how it affects path integration. Because the optical flow duration is fairly short (1.3s at most as selecting targets further away than 13m was difficult) the use of an S-curve or logarithmic is nearly imperceptible and for it to be more noticeable we would have to increase the duration, which to minimize VR sickness was undesirable. We also felt that path integration was more difficult if users would also have to gauge changes in velocity. We found a constant velocity of 10m/s to work best as this closely approximates the efficiency of instant teleportation while the amount of optical flow optical seemed to sufficiently allow for path integration.

Vection is known to induce VR sickness but we hypothesize that since users are only exposed to optical flow for a very brief time, the likelihood of inducing VR sickness is low. FOV reduction is an effective strategy to minimize VR sickness [26] during locomotion. This strategy is also used by Raw Data which blurs the screen during the teleportation. However, FOV reduction or blurring diminishes the perception of optical flow which is needed to perform path integration, so we decided not to implement this.

4.1.4 Procedure

We used a within-subjects factorial design with independent variables teleportation_method, i.e., (regular, Dash) with dependent variables; 1-teleport and 2 teleport path integration error (PI-error). The two tasks both measure path integration ability and are not independent variables. To control for order effects, we counterbalanced the order of independent variables tested, e.g., each participant
was randomly assigned to one of four groups (A, B, C, D) such that each group contained an equal number of participants. To analyze whether Dash induces VR sickness, we used the Simulator Sickness Questionnaire (SSQ) [30]; a standardized questionnaire that quantifies various aspects of simulation sickness. Before the trial participants filled in an SSQ to get a baseline reading. Each trial consists of four blocks, with participant testing one particular teleportation method in the first two blocks and the other in the last two blocks. Each block had 10 navigation tasks, and the first and last two blocks were then randomized for 1-teleport and 2-teleport. We did this so that participants could fill in another SSQ after they completed two blocks using the same teleportation method. Figure 4.3 shows an overview of the blocks for each trial for each group.

User studies were held in a large open lab space free of any obstacles or interference, and participants were fitted with the Google Daydream headset. Prior to the trial participants performed a brief built-in tutorial containing two 1-teleport tasks and two 2-teleport tasks using their first assigned first teleportation technique to familiarize them with the navigation task. After the first two blocks, participants took off the headset, filled in an SSQ and rested for 15 minutes to let any VR sickness effects subside. Because the Daydream controller tends to drift over time, we
re-calibrated prior to each trial and after the first two blocks. The whole trial took
about 35 minutes per participant. After the last two blocks, participants filled in a
third SSQ followed by a questionnaire that collects demographic information and
which aimed to determine a ranking between teleportation methods, based on a
number of criteria.

4.1.5 Measures

For every navigation task, we collect the coordinates of E (see Figure 4.2). Given
the known coordinates of S, 1 and 2, we calculate V1, V2 and V3 depending on
what teleportation method was used (see Figure 4.2). We analyze PI-error for both
navigation tasks. V3 is the difference vector between the start (S) and the pointed
location (E) and its length reflects the PI-error in distance as well as rotation (see
Figure 4.2). Rotation error relies on vestibular cues and optical flow [46] but since
our virtual environment was designed to be devoid of any visual features, rota-
tion primarily depends on vestibular cues. However, to focus on the benefits of
optical flow during teleportation, we exclude for errors in rotation as this doesn’t
depend on optical flow cues during teleportation and we only evaluate linear path
integration (distance travelled) for which optic flow is the only cue.

For 1-teleport, we project V2 on V1 and calculate PI-error for 1-teleport as |V1 −
V2|/|V1| (we normalize the difference in length between waypoints by dividing by
V1). For 2-teleport, the error in the second rotation needs to be included as this
relies on path integration when the user travels from 1 to 2 (given that S, 1 and
2 are non-co-linear). To assess PI-error for 2-teleport, we use |V3|/|V1| which is
the normalized length of the difference vector V3 and which embodies the error in
both distance and rotation.

### 4.1.6 Participants

We recruited 16 participants (5 female, average age 25.0, SD=4.6) for our user study. All participants had experience with navigating 3D desktop environments. Six participants had no experience with VR, five had some VR experience and five had lots of VR experience. None of the subjects self-reported any non-correctable impairments in perception or limitations in mobility. The user study was approved by an institutional review board.

### 4.2 Results

Table 4.1 list the average PI-error for 1- and 2-teleport for each teleportation method (e.g. Dash or Regular). A one-way repeated measures MANOVA found a statistically significant difference in path integration error between teleportation techniques, \((F_{2,158} = 7.353, p < .05, \text{Wilk's } \lambda = .915, \text{ partial } \varepsilon^2 = .085)\), with a significantly smaller error using Dash. There was homogeneity of variances, as assessed by Levene’s Test \((p > .05)\). A Tukey post-hoc analysis found a significant difference for teleportation methods for both PI-errors \((p < .05)\). An analysis of pointing errors, by analyzing the sign of \((V1 − V2)\), using a \(\chi^2\) test found that users significantly \((p < .05)\) underestimated the distance to their start location, though there was no significant difference between teleportation methods.

Regarding VR sickness, a Mann-Whitney U test didn’t detect a significant dif-
Table 4.1: Quantitative results.

<table>
<thead>
<tr>
<th></th>
<th>Dash (SD)</th>
<th>Regular (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI-error (1-teleport)</td>
<td>.149 (.10)</td>
<td>.172 (.12)</td>
</tr>
<tr>
<td>PI-error (2-teleport)</td>
<td>.527 (.20)</td>
<td>.594 (.25)</td>
</tr>
<tr>
<td>Nausea ((max: 200))</td>
<td>6.56 (10.3)</td>
<td>6.56 (7.6)</td>
</tr>
<tr>
<td>Oculomotor (160)</td>
<td>6.16 (13.6)</td>
<td>7.58 (9.2)</td>
</tr>
<tr>
<td>Disorientation (292)</td>
<td>10.44 (18.7)</td>
<td>11.31 (13.7)</td>
</tr>
<tr>
<td>SSQ-total (235)</td>
<td>7.71 (13.1)</td>
<td>9.12 (9.1)</td>
</tr>
</tbody>
</table>

Figure 4.4: Ranking of teleportation methods on four criteria.

ference ($U = 109.5, Z = .678, p = .496$) between total SSQ-total scores or any of its sub-scores, e.g., nausea, oculomotor and disorientation. The average SSQ-Q scores, 7.71 for Dash and 9.12 for regular, which had a maximum possible value of 235, would rank between no and mild VR sickness [30].
We also asked participants to rank each teleportation method on four criteria where participants could select “no difference” as a 3rd option. Fifteen participants thought Dash helped best maintain spatial awareness, with 1 participant stating there was no difference. Regarding which technique caused VR sickness the most, opinions were split with 5 participants stating Dash, 5 participants stating regular teleport and 6 participants stated that none of the techniques caused VR sickness. 11 participants said regular teleport was most efficient, and 2 stated Dash was more efficient with 3 saying there was no difference. Regarding preference, 12 participants liked Dash the best and the other 4 liked regular teleportation the most. Figure 4.4 lists the results, and a $\chi^2$ test found the rankings for spatial awareness, efficiency and preference to be statistically significantly different ($p < .05$).

4.3 Discussion & Limitations

Our study innovates over existing teleportation studies [16, 11, 21, 31] in that we assess spatial disorientation using a path integration navigation task that doesn’t rely on prior knowledge of the environment that users navigate in. Our navigation tasks don’t rely on cognitive mapping ability and therefore better assess the benefits of optical flow during locomotion. In addition, this task seemed more suitable, as many existing VR applications allow users to explore large new environments, which are difficult to cognitively map.

Dash offers a significantly lower PI error than regular teleport, but this difference for 1-teleport isn’t very large –though it significantly increases for 2-teleport. Because users often use teleportation multiple times, this difference in PI error will continue to accumulate and grow unbounded over time, as this is an inherent char-
acteristic of dead-reckoning localization. To exclusively focus on path integration, our virtual environment did not contain any visual landmarks, but we are confident our results hold for more realistic virtual environments. Such environments will have more visual landmarks, which reduces the PI error for each teleport, but since these errors accumulate and propagate over time anyway, the benefits of using Dash to reduce the PI error for each teleport are evident.

Our study focused on mobile VR platforms because of their popularity and potential to bring VR to the masses. Because we didn’t observe a significant increase in VR sickness we believe Dash can be used on PC VR platforms as they have higher refresh rates, lower latency and offer 6DOF tracking and are less likely to induce VR sickness than mobile VR platforms. Because we didn’t observe an increase in VR sickness, the use of field-of-view reduction in a similar teleport method used in Raw Data [6] seems superfluous and may actually impede path integration as it blocks optical flow.

Most participants (n=11) found regular teleport to be most efficient, which made sense as Dash was slightly slower. Dash moves the viewpoint with a speed of 10m/s and our waypoints varied between 5-11 meters, so it added a fixed amount of time .5 and 1.1 seconds to each teleport, which is negligible given that it enables path integration. Though we designed our user study to be devoid of any visual information that conveys distance, the pointer used for selecting a waypoint does convey distance to some extent.

There was no consensus among the participants’ ranking which method caused VR sickness the most (see Figure 4.4), but quantitative results only found a slight increase in SSQ scores for both methods, with no significant differences between them. Seven participants (all males, all experience with VR) in our study showed
no increase in SSQ scores from their baseline. A recent study [40] has found that females are more susceptible to VR sickness. All five participating females showed a small increase in their SSQ scores (no to mild VR sickness), but we didn’t enroll enough females to allow for testing whether this difference was significant. Participants were exposed to VR for approximately 20 minutes, which should be long enough to induce VR sickness if participants were susceptible to it [40]. Future work will investigate whether there may be a gender difference in SSQ scores using Dash.
5.0.1 Legomotion

Implementing optimal locomotion methods for navigating large VR environments has remained a challenge. Real walking input—implemented using positional tracking—generally delivers the most immersive experiences as it most closely mimics interaction with the real world, but it is limited by available tracking space and doesn’t allow for exploring large VR environments, without switching to a controller. We present a novel solution to the VR Locomotion problem: legomotion; a VR locomotion technique that is a hybrid of walking and walking-in-place. A user study with 18 participants shows that legomotion offers a higher presence but slower performance because the use of controller/walking based navigation leads to users abandoning walking input. legomotion enables handsfree locomotion at scale at low cost.

5.0.2 Dash

Teleportation is known to cause spatial disorientation, because the absence of optical flow during the instant translation doesn’t allow for estimating the distance using path integration. We evaluate Dash—a modified version of teleportation—that quickly but continuously translates the user’s viewpoint, and which generates a small amount of optical flow. A user study that compared Dash to regular teleport found Dash to allow for significantly better path integration, and amount of optical flow generated does not induce VR sickness.
5.0.3 Future Work

In the future we intend to focus on precisely understanding this trade-off to identify the smallest tracking space, that does not lead to users abandoning the use of positional tracking input.

We purposely did not compare legomotion to teleport although this method is widely used. Navigating using teleport is significantly faster than using real walking. Because it is also significantly faster than full locomotion it is at an even higher risk of being adopted by users as their primarily locomotion method even when positional tracking is available. Future work will try to substantiate this assumption.

Overloading the hands with navigation functionality not only breaks presence, but also increases cognitive load which impedes efficiency [33]. We will evaluate how legomotion affects cognitive load for tasks that require users to use both hands and compare this to a controller based locomotion method.

Because vestibular and proprioceptive feedback plays a major role in spatial mapping [68], their absence in controller-activated artificial locomotion techniques has been found to limit users’ ability to spatially map virtual environments [11]. Future work will investigate whether the proprioceptive feedback generated using legomotion, will allow for better spatial mapping.

In FaceIT application, the current study provides a starting point for the exploration of face at any distance and at any angle. Better face detection algorithms could also be tested or new ones could be explored to reduce the problem of background face detection. This application could be extended to other head mounted
devices which could be cheaper than Google Glass but have higher efficiency. Non-verbal cues given by conversation partner could also be incorporated in this application to increase its effectiveness. The application could also be expanded to detect expressions and body language as well.
BIBLIOGRAPHY


Mel Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. Philosophical Transactions of the Royal Society B: Biological Sciences, 364(1535):3549–3557, 2009.


