University of Nevada, Reno

EXPLORING HANDS-FREE ALTERNATIVES FOR TELEPORTATION IN VR

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

by

Walker Spurgeon

Dr. Eelke Folmer / Thesis Advisor

August 2018

UNIVERSITY OF NEVADA RENO

THE GRADUATE SCHOOL

We recommend that the thesis prepared
under our supervision by

WALKER SPURGEON

entitled

EXPLORING HANDS-FREE ALTERNATIVES FOR TELEPORTATION IN VR

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 2018

ABSTRACT

In recent years, Virtual Reality (VR) technology has reached a point where it is powerful enough to be immersive, yet cheap enough to be commercially available. Both high-end headsets designed for desktop computers and low-cost peripherals for smartphones are seeing increased usage by consumers and developers alike. While there are differences in motion tracking capabilities between various VR devices, locomotion remains a common problem due to space constraints, VR sickness, limited input on low-cost devices, and the need for immersion in VR. Currently, a popular technique for locomotion in VR is teleportation. For headsets with positional tracking, teleportation allows users to navigate beyond the tracking space without a high risk of inducing VR sickness; for devices without positional tracking, teleportation can allow users to move in the virtual world even if the device has limited input options or low computational power. The most common teleportation methods rely on controller input, usually with motion controllers. This kind of input has downsides in that it can lead to arm fatigue, and it is not a viable method for devices without motion controllers or for people who cannot use motion controllers because of a disability or injury. We evaluated four hands-free methods for teleportation and compared their performance with teleportation using motion controllers. Two of the methods - teleporting using a voice command and teleporting by having the user's gaze dwell on the desired destination - had been used previously. The other two methods - teleporting via foot stomp and teleporting via blink - were novel. We performed a study in which users would teleport to waypoints in a VR environment, and the speed and accuracy of their teleportation was recorded and compared between the various methods. The study compared teleportation via controller with the blink, stomp, voice, and dwell methods. Data analysis of our results suggest that the blink and dwell methods have comparable results to controller teleportation and may serve as viable hands-free alternatives. Both methods had comparable accuracy, and blink in particular was well-received by users and did not have a large time increase.

ACKNOWLEDGEMENTS

I would like to thank Dr. Eelke Folmer for supporting and advising me since I was an undergraduate, as well as the other members of my committee, Dr. Feng Yan and Dr. Paul MacNeilage. I also owe thanks to all the members of the HumanPlus lab and my other friends and family who have supported me along the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The ability to move around freely and explore has been one of the fundamental appeals of 3D games since their inception, but implementing this free motion in virtual reality (VR) has proven challenging. In other 3D games, walking is a common means of traversing the game world. Walking input for VR has been facilitated with positional tracking, which offers high presence [1], but this method is limited by the tracking space which is itself limited by the available physical space. Thus, artificial locomotion techniques (ALTs) become necessary to navigate large VR environments, and these are typically controller-activated. Common examples of ALTs include teleportation, linear movement in any direction using a thumb stick or trackpad, and vehicular transport. ALTs such as linear movement and vehicle transport usually generate optical flow without providing any vestibular or proprioceptive stimuli, which can confuse the senses and lead to vection-induced VR sickness [1, 2]. With teleportation, however, the user's viewpoint discontinuously translates and no optical flow is generated. The absence of optical flow cues has shown to impede path integration (i.e., estimating distance travelled) [3], but it mitigates sensory conflict. Typically, teleportation involves using a motion-sensing controller to select a teleportation destination using a ray-cast.

In recent years, people have been able to experience VR –often for the first time– using low-cost VR platforms and peripherals such as Google Cardboard and Gear VR. Since such platforms only require a smartphone which is appropriated into a VR display, they have the largest potential to bring VR to the masses [4]. Mobile VR platforms have limited input options [5]; most do not support the use of controllers and rely largely on head tracking and a single button for user input. Since the one button cannot be easily have multiple uses in a single application, there is a limit to the type of VR experiences that can be developed for mobile VR [6]. There is also a limit to the amount of engagement that these experiences can provide.

The standard input devices for PC VR platforms are controllers with 6 degrees of freedom (DOF). For some mobile VR platforms, such as the Daydream or Gear VR, a 3-DOF controller is available. While most gesture-based interactions for PCs, smartphones, and other devices are bounded by a physical touch surface, VR interactions largely rely on mid-air arm movements using a controller. Prolonged use of these mid-air interactions can lead to arm fatigue [7, 41, 9, 10, 11]. This makes physical ergonomics a significant design consideration for VR, as this arm fatigue can be detrimental to the user experience [9]. Given its wide use in VR games and other applications, teleportation may be a significant contributor to gorilla arm syndrome, which occurs when a user experiences arm fatigue from having to lift their arms for long periods of time. Pointing with a controller to teleport can be done while resting the arm on the hip, but in practice, users will typically raise the controller inside their field-of-view (FOV) so they can see it in the virtual world. This allows users to reduce any mismatch between their visual and proprioceptive senses, but may lead to arm fatigue.

Low cost external depth-sensing cameras such as the Leap Motion [12] are already available and allow for articulated hand tracking. Future VR headsets may feature integrated depth cameras and may not have motion sensing controllers available if users prefer using their hands for input. Some VR applications may also use controllers that designed to be immersive but may not be suitable for teleportation; controllers in the shape of a gun for first-person shooters or controllers in the shape of musical instruments are two possible examples. With these possibilities in mind, hands-free teleportation can still allow for immersive experiences on both mobile and PC VR platforms without limiting mobility or inducing arm fatigue. Hands-free teleportation might also allow users with severe motor impairments increased access to VR.

Eye gaze has recently been explored for use in teleportation [13] but a controller was used for activation. The work presented here explored using head gaze to select a teleportation destination with four hands-free methods of activating the teleport. The four activation methods are blinking, stomping with feet, speaking a voice command, and dwelling on the desired destination for a set period of time. We had users teleport to set waypoints in a VR environment and evaluated the speed, accuracy, and usability of the hands-free teleportation methods with controller teleportation as a benchmark for comparison.

The contribution of this paper is as follows: (1) we present designs for hands-free teleportation methods that can be used on mobile VR platforms; and (2) we present results from a study comparing the performance of these teleportation methods with controller teleportation, and evaluate the feasibility of these methods for use in VR.

CHAPTER 2

BACKGROUND AND RELATED WORK

One of the major barriers for the mass adoption of VR is giving users the ability to navigate beyond the available tracking space while maintaining a high presence, all while minimizing cost and VR sickness. Natural walking can be immersive and offer high presence, but unfortunately it can't easily scale to navigate large environments. and physical space and cost may also be limiting factors. Redirected walking techniques [15] apply gains to viewpoint rotations and translations, which can give users the impression that they are walking in a straight line, while they are actually walking in a circle. These techniques can be immersive and offer high presence, but they require a large tracking space (over 22 meters) [16] to be imperceptible. This space exceeds both the current tracking limitations on existing consumer VR systems and the available physical space in a typical home.

There have been various proposals for hardware ALTs that can allow navigation in a large VR environment. These proposals include omni-directional treadmills [17], leaning chairs [18], motorized roller skates [19], and a human-sized hamster ball [20]. Hardware-based ALTs often suffer from high response times which can cap movement speeds or keep users from being able to make abrupt turns; the omni-directional treadmills currently in development suffer from these issues [14]. These ALTs usually require users to be strapped into them or onto them, which may be incompatible with existing positional tracking systems. This constraint may also impede user movement, such as limiting the ability of the user to kneel and pick up an object [21] or swing an arm or controller. There are other concerns that could limit the usage of some of these ALTs. Physical space may make them less feasible, particularly in the case of treadmills or a giant hamster-ball, and the cost may be prohibitive. They may also be unavailable for mobile VR platforms; the treadmills being developed are designed for use with PC VR, and may require too much processing power to function properly on a mobile platform if calculating the movement direction is complex.

Since natural walking offers the highest presence in VR, there are several walking-based ALTs that have been developed that can function alongside existing positional tracking systems. Some examples are walking-in-place [22, 5], arm swinging [23], and leaning input [24]. These ALTs have yet to be widely adopted, but have benefits in their low implementation cost and their ability to generate some or all of the vestibular and proprioceptive stimuli generated by natural walking. Since VR sickness can be caused by a visual/vestibular conflict [25], the ability to generate such vestibular cues can reduce the likelihood or effect of VR sickness [21, 26]

Vehicle movement, full locomotion with a joystick or trackpad, and telportation are the most widely used ALTs that can be integrated with positional tracking systems without impeding movement. Vehicle movement has users enter a vehicle or board a platform to move larger distances, and full locomotion requires a controller. Both of these ALTs generate optical flow which can induce VR sickness [2, 1].

Teleportation, on the other hand, doesn't cause vection-induced VR sickness because there is an absence of optical flow when moving from one point to another [27]. The lack of optical flow does have downsides in that in doesn't allow for path integration (estimating the distance travelled) and can cause spatial disorientation [28, 3]. There have been a few approaches aiming at improving teleportation. LaViola [29] presented a modified teleportation in which a map is rendered at the user's feet and the users must step into a location to teleport to that location. Freitag et al [30] presented a teleportation method in which a portal to the teleport destination appears behind the user, and the user walks through the portal to teleport, optimizing usage of the limited tracking space. With point and teleport [25], users can specify their post-teleport orientation. Dash [3] adds a small enough amount of optical flow to the viewpoint transition, and found in a user study that this improved path integration without inducing or increasing VR sickness.

The following works are most closely related to the work presented here. Beckhaus et al [31] and Zielasko [32] both presented results from comparative studies between multiple hands-free locomotion methods, but these methods involved continuous movement rather than teleportation. Jumper [33] was a hands-fre teleportation method for PC VR platforms; this method used positional tracking and had users physically jump forward. They would take a giant leap forwar in the virtual world to a location specified their gaze. A user study with 11 participants found that jump had no performance difference when compared with natural walking and teleport, but was easier to learn than teleport. Jump did, however, worsen spatial orientation when compared with natural walking. Linn [13] evaluated teleportation based on eye gaze using a Tobii eye tracker integrated into an HTC Vive headset. Users would select their teleportation destination using their eye gaze and would activate the teleport with a button on their controller. In this paper a study with 12 subjects found that the eye gaze-based teleport had no significant difference in performance when compared to controller teleportation, but users preferred teleporting using eye gaze over teleporting with a controller.

CHAPTER 3

THE DESIGN OF HANDS-FREE TELEPORTATION

We specifically explored hands-free solutions that require no or a minimal amount of instrumentation. This is to keep potential costs low and to allow for the methods to be adopted on mobile VR platforms. Teleportation consists of the following two distinct tasks: Selecting a destination to teleport to, and activating the teleport.

3.1 Destination Selection

Assuming users engage in grounded navigation –in other words, they are not flying or swimming– then selecting a surface coordinate (X,Y) requires at least 2-DOF input. Without using a controller, and without requiring additional sensors, this task can only be performed using gaze input, since most VR headsets on both PC and mobile platforms are capable of at least 3-DOF head tracking using inertial sensing. Selection using head gaze is already widely used on mobile VR platforms; some newer VR headsets also support eye tracking [34]. Eye gaze can be faster than mouse-based 2D selection [35] but for 3D selection, eye gaze is slower than a controller for distant objects [36]. Eye gaze has been explored for navigating in VR for methods such as steering [37], but all of these instances involve locomotion methods that generate continuous optical flow, rather than teleportation. Head gaze was used because it is simple to implement on all platforms and requires no extra attachments or devices. Head gaze also allows users to look around at the environment without having to move their teleportation destination.

3.2 Teleport Activation

Four techniques were explored for hands-free activation of teleportation.

3.2.1 Dwell

Dwell is the de facto selection technique for gaze input; it requires users to fixate their gaze for a specified time duration in order to select an item [35]. A long time duration for dwell can significantly affect performance by making relatively simple actions such as selection and teleportation take too long to perform. However, using a short dwell time can harm selection accuracy as the result of a "Midas Touch" problem, where users will teleport to any location where they briefly fixate their gaze. A remedy we implemented took advantage of the fact that users are constrained to ground navigation. The teleportation cursor is only rendered when the user's gaze intersects with the ground plane. Then the user can teleport by fixating their gaze on any location where the cursor appears and can raise their gaze if they do not wish to teleport. This largely circumvents the Midas Touch problem, allowing users to look around freely without unexpectedly teleporting. This approach does have limitations in that it only works for environment that are fairly flat, and not for environments that have slopes, elevated surfaces, or stairs. Depending on the environment, this problem could also be mitigated by making the cursor disappear after a certain distance.

3.2.2 Voice

Voice input is a popular hands-free interaction technique for mobile devices, and it is widely used in intelligent personal assistants. For use in teleportation, a specific word or phrase can be set as the teleport activation command. Voice commands can be recognized with a good degree of accuracy on smartphones because they usually feature a pair of microphones(one near the mouth, one near the ear) which enables noise cancellation. However, with mobile VR platforms the smartphone is often embedded inside the VR adapter, which means both microphones could be occluded which could impede speech recognition accuracy since the smartphone can't take advantage of its noise cancellation [6]. Another possible issue with using voice commands is that it may be detrimental to presence, particularly if the VR experience already features sound and music, or has situations where one would normally want to be silent (e.g. hiding from an enemy). However, this method may be particularly immersive in other instances, such as games featuring magic where the voice command could serve as a spell incantation to teleport. This method may be less viable on PC VR platforms as the headsets may not have microphones and likely will not have noise cancellation.

3.2.3 Foot Stomp

Smartphones feature Inertial Measurement Units (IMUs), and these sensors can be appropriated for VR input. Examples of this include enabling walking-in-place [5] or head-tilt navigation [21]. Steps, foot stomps, and jumps generate unique acceleration patterns that can be detected with reasonable accuracy [5]. These patterns, once detected, can serve as the activation command for teleportation. Similar to the Jumper metaphor [33], a foot stomp may offer users a way to activate the teleportation in a natural and intuitive way. A possible issue with this technique is that differences in physique may affect the detection accuracy. There may also be issues with the stomp motion causing head movement, changing the teleportation destination. This technique could also be difficult or impossible for users who are physically impaired.

3.2.4 Blink

Some teleportation implementations apply a fade in/fade out affect to teleportation. Though this takes slightly longer, it can make the viewpoint transition less abrupt. Blinking your eyes in an exaggerated way can create a similar visual effect, and may be useful for activating teleportation in a natural way with high presence. Eye blinks can be detected using an eye tracker, but these are not available on mobile VR platforms; however, smartphones have sensors that could be appropriated for this purpose. To use blinking for teleportation, the device needs to be able to detect if an eye is open or closed. The teleport would activate when the user's eyes are closed for a specified amount of time. This could be advantageous compared to techniques like dwell, since the time threshold can be much lower. The time threshold for this method has potential to cause problems. If the threshold is too low, then regular blinks could cause the user to teleport unintentionally, and since the user's eyes are closed when the teleportation occurs, a high threshold could lead to uncertainty as to whether a teleport has occurred unless it is accompanied by other feedback such as sound.

CHAPTER 4
USER STUDY

The goal of the user study described here was to evaluate the performance (in the form of speed, accuracy, and usability) of the hands-free teleportation methods versus controller-based teleportation to identify which method worked best.

4.1 Instrumentation

For this study, we used the Google Daydream mobile VR platform. The Daydream setup offers a 960 x 1080 pixels per-eye resolution at 60Hz with a 90 degree field of view using the Google Pixel smartphone (Snapdragon 821 2.15Ghz Quad-Core). For interaction, the Daydream features a wireless inertial sensing remote controller with a touchpad and several buttons. Because the Daydream doesn't feature positional tracking, many Daydream apps primarily rely on teleportation for locomotion. An additional benefit of focusing on mobile VR platforms is that our results could significantly increase interaction options of first generation mobile VR platforms like Gear VR and Google Cardboard since these platforms do not feature a controller. We implemented teleportation using the Unity 3D engine (version 5.5.1) and Google VR SDK.

Many VR experiences that use teleportation let their users select a location using a teleport cursor that is tethered to the ground plane. On PC VR platforms, controllers are usually tracked using 6-DOF, but mobile VR platforms typically only support 3-DOF tracking. Because of this tracking limitation, many mobile VR implementations use an indirect selection mechanism rather than having the user directly point at a location. Typically, this is in the form of a parabolic arch rendered from the controller to the cursor, and users can manipulate the cursor by tilting the controller. Gaze selection, on the other hand, typically uses a ray-cast. In order to avoid having differences in performance as a result of using a different selection mechanism, we also implemented a ray cast for selecting a location using a controller.

We implemented each teleportation method and fine-tuned their implementation using preliminary experiments. Our teleportation cursor was a green circle with an orange outline (see Figure 4.2: left). Since we controlled the cursor using ray-casting, we did not render a visual arch from the controller to the cursor as this would obscure the user's view. The maximum distance from the user to the ray-cast was set to 15 meters for all techniques, as pointing precisely beyond that distance was found to be difficult in preliminary experiments. For regular controller teleportation, after users selected a location using a ray-cast from their controller, users teleported by pressing the touchpad. To implement dwell, we found a dwell time of 1.5 seconds to work well. Any longer than about 2 seconds, and the threshold felt too long, but 1.5 seconds was still enough time to prevent most accidental teleports. To further mitigate the Midas touch problem, we only rendered a teleportation cursor when the ray-cast intersected with the ground plane (this implementation was used for voice, stomp, and blink as well). The Midas Touch problem was also somewhat alleviated by the 15 meter limitation for the cursor to appear. Voice activation was implemented using an open source speech recognition library called pocketsphinx for Android. To activate the teleportation, we ended up using the phonetically distinct keyword of "Okra", which was recognized with a much higher accuracy than using the words "Go" or "Teleport". Though this library and implementation worked, there was some noticeable latency of about 1.5 seconds.

To implement stomp teleportation we used the VR-step Unity asset [38] that can be used to implement walking-in-place on mobile VR platforms using inertial sensing. It was difficult to find a stomp detection threshold setting that worked for everyone, as this was subject to individual preference, physique, and ability. Thus we implemented a default stomp threshold acceleration value (0.28g), but let users adjust this using a slider inside the application we used for our user study.

Implementing Blink on mobile VR platforms was a challenge as current mobile VR headsets do not support eye tracking. We originally implemented a hack for the Gear VR platform [39] that enables blink detection. This hack requires a small fibre optic cable that is placed inside the VR headset close to the eye. The cable and is then connected to the smartphone's front camera. The concept behind this modification is that light from a scene reflects of the user's eye. When the eye is closed a change in luminance of the light reflected from the eye can be detected. We implemented this for the Daydream headset but due to the location of the camera, we instead had to appropriate the Pixel's light sensor. We were eventually able to implement it successfully, however we struggled to get this method to work reliably. Actual eye blinks were detected with a reasonable accuracy, but false positive were also generated often when the luminance of the scene abruptly changed. For example, if users were looking at the sky or a lighter color on the ground, and then looked at a tree or darker section of ground, both of which were dark green, then it would be detected as a blink. Another issue with the Daydream headset is that it was not large enough and did not stick to faces well enough to block out all light. Light could enter through the sides of the headset, which would also trigger false positives, and this could only be mitigated by turning off all external lights.

As a result of these technical difficulties, evaluating blink teleportation using a mobile platform proved to be unrealistic. Instead, to evaluate blink for this specific study, we ended up using the Fove headset [34], which is a tethered VR headset with a 1280 x 1440 pixels per-eye resolution at 90Hz with a 100 degree field of view. This headset was connected to a high-end desktop PC (Intel Core i7 3.8GHz 32GB RAM, NVIDIA's GTX 1070). The Fove headset features an integrated infrared eye tracker that, aside from tracking eye gaze, can detect blinks with a high degree of accuracy. We explored different values for the time threshold for which one eye needed to be closed to activate the teleport. We found a value of 170ms to be the best compromise between generating false positives due to regular eye blinks, which take about 100ms–150ms according to a UCL researcher [40], and having to excessively blink to teleport. For preliminary experiments, this was a short enough threshold that simply blinking manually was usually sufficient, and users did not have to consciously exaggerate the blink, while most automatic blinks did not teleport the user.

Unfortunately, the Fove does not support any type of motion sensing controller. As a result, we had to use the Fove for evaluating blink and the Daydream for the other conditions. To mitigate for any cross platform differences we set the resolution and refresh rate of our Unity application on the Fove to use the Daydream specifications (e.g. 960 x 1080, 60Hz), while their field-of-view for the Daydream and the default field-of-view setting for the Fove were the same at 90 degrees. Although the Fove headset weighs 91 grams more, we didn't observe any significant differences in comfort levels between wearing both headsets. We also tested dwell teleportation on both platforms to check for any noticeable differences in application performance (such as changes in frame rate or delays in responding to head motion) and found none. Since our study didn't involve any physical translation, we attached the cable of the Fove headset to the ceiling to offer users the same ability to freely rotate as they can on the tether-less Daydream headset.

Some teleportation implementations in games and other existing applications apply a fade in/fade out or motion blur to the instant viewpoint transition during teleportation. It has been argued that this could mitigate VR sickness but since there are no studies that have analyzed the effectiveness of such a transition on spatial orientation or VR sickness, we implemented teleportation using the simple instant viewpoint transition without added effects, which itself is commonly used. While blinking does offer a sort of fade in/fade out transition as a result of users closing and opening their eyes, we determined that it is not likely to make a significant difference in this experiment due to the short duration of blinking and the relatively short duration of each trial.

4.2 Virtual Environment & Navigation Task

Our virtual environment consists of a low-poly terrain peppered with low-poly trees (see Fig 4.2) as well as some roads, paths and water features to make it look somewhat like a realistic environment. The use of a low-poly environment helps boost the performance of the application, and we did not observe any noticeable latency in rendering or tracking between the Fove and the Daydream platforms used for the study.

For each teleportation method, the participants performed a virtual navigation task along a virtual path defined by sequence of 24 waypoints. We used tall and distinctively colored (purple) cylindrical objects each with a 1.40m radius as waypoints to make them easy to spot and to prevent them from being occluded by

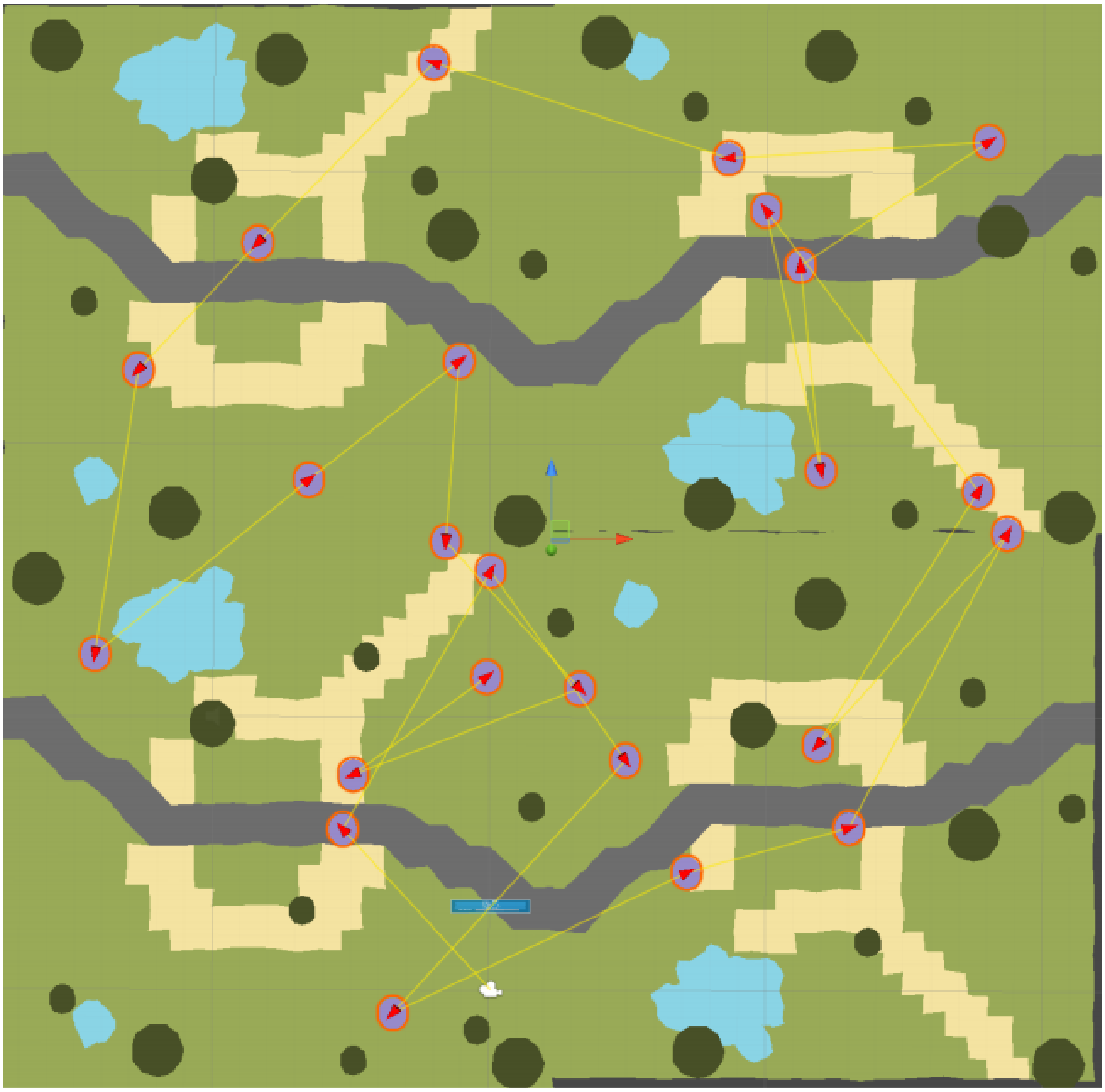

Figure 4.1: A top-down view of the virtual environment showing the path users had to navigate and the locations of each waypoint that defines the path

trees. The distance between consecutive waypoints was predefined but randomly selected from the range 5-13m, which we deemed reasonable minimum and maximum distances a user would teleport, based on the size of VR tracking spaces and the distance at which precise teleportation becomes difficult. Figure 4.1 shows an overview of the path and the 24 waypoints.

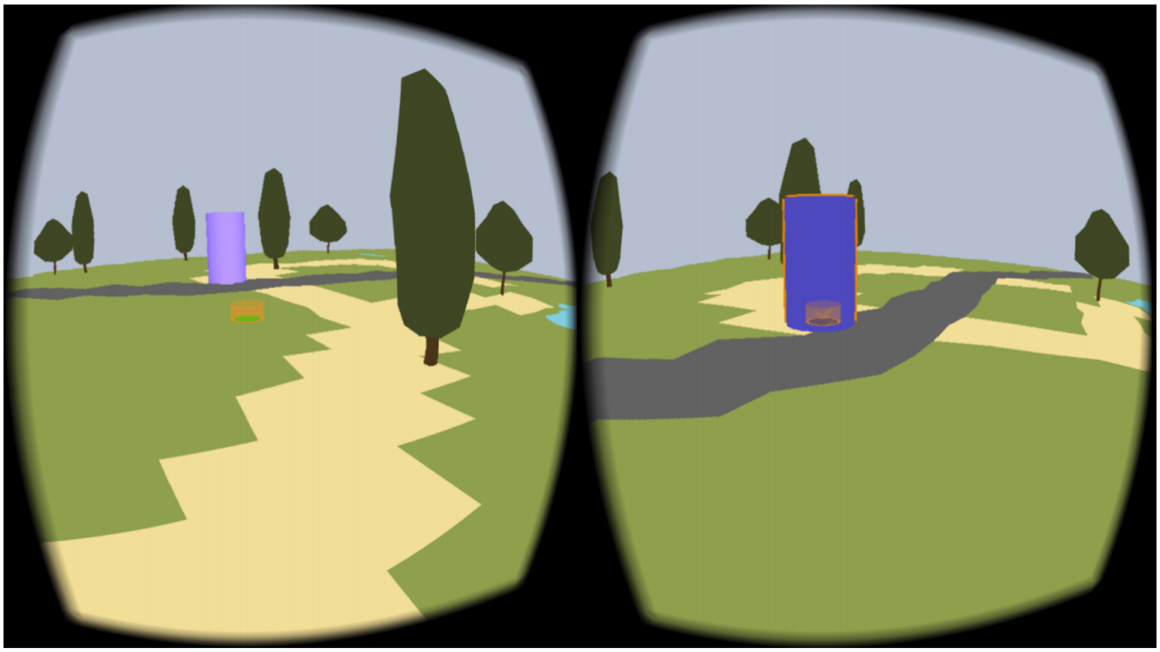

Figure 4.2: A view of the virtual environment used for the user study as seen from a mobile VR headset. The purple column indicates a waypoint to navigate to. Left: a green circular teleportation pointer is visible. Right: a column turns dark purple when the teleportation cursor is placed inside it.

The navigation task was to go from the current waypoint to the next as quickly as possible using the specified teleportation technique for that trial. To get to the currently visible waypoint, participants first pointed their cursor at the waypoint using either the controller pointer or the gaze pointer and then activated their teleportation. Selection feedback was given by highlighting the waypoint. Upon activation, the virtual viewpoint was instantly teleported to the pointed location. If this location was within 1.40m of the center of the waypoint, the waypoint would disappear with an auditory feedback and the next waypoint would appear. If the pointed location outside this radius, the participant would need to make another teleportation attempt until they had reached the waypoint. This would repeat until all waypoints had reached and there was a visual indication that the task was complete.

4.3 Procedure

We used a within-subjects factorial design with teleportation method (i.e. controller, voice, dwell, stomp, blink) as the independent variable. To control for order effects we counterbalanced the order of independent variables tested; that is, each participant was randomly assigned to a group such that each group contained an equal number of participants (plus or minus one participant). Each group then started with a particular teleportation method first (e.g., group 1 started with controller-based teleportation, group 2 started with voice teleportation, etc). To allow for comparison between methods, we used a predefined but randomly generated sequence of waypoints and the same sequence was used across all teleportation techniques and participants. To minimize the possibility of participants memorizing the sequence of waypoints from one method to the next, we changed the start position for each method and the order in which the waypoints were presented. Participants using the controller method would start with waypoint 1 then go to waypoint 2, and so on. For voice teleportation, users would start at waypoint 8 and then go to waypoint 7 to traverse the path in reverse order. For dwell teleportation, users would start at waypoint 16 and then go to waypoint 17, and for stomp teleportation they would reverse the path starting at waypoint 24. For blink teleportation, users started at waypoint 12 and would go to waypoint 13. 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, or the Fove headset when conducting the blink trial. Prior to the trial, participants performed a brief built-in tutorial where each teleportation method was explained and participants had to complete a 10- waypoint navigation task using each method. This allowed the participants to get some familiarity with each method so they could reasonably teleport as they desired without risking too many false positives. For stomp teleport, participants were allowed to adjust the step sensitivity from a default value to reduce false positives or to increase step detection, accounting for differences in physique and physical ability.

4.4 Data Collection

To allow for a fair comparison between techniques, we decoupled the visual search from the navigation task, i.e., if the lateral angle between the user's gaze pointer and the landmark is less than 40 degrees for 0.75 seconds, we assume that the user is seeing the waypoint. For every waypoint, and for every teleport issued to get to the waypoint we recorded: user location as a 2D coordinate, teleportation cursor location as a 2D coordinate, visual search duration in seconds, and teleport travel time in seconds. Additionally, for voice and stomp teleportation, an observer recorded the total number of commands issued in order to better asses the accuracy of the techniques. This was not possible for blink detection. The whole trial took about 20 minutes per participant. Given the short duration of our study, we did not attempt to measure arm fatigue as this typically only manifests itself during prolonged periods of VR interaction [41]. After the last trial, participants filled out a questionnaire that collected demographic information and which aimed to collect qualitative feedback on each teleportation method, based on a number of criteria. Evaluations on each teleportation technique were collected in the form of 5-point Likert scales, with sections for uses to write any comments regarding the teleportation methods.

4.5 Participants

We recruited 16 participants (3 female participants, 13 male, with an average age of 23.43 years and a standard deviation of 5.4 years) for our user study. All of the participants had experience with navigating 3D desktop environments. Three participants had no prior experience with VR, ten had a some experience and three had lots of experience with VR. None of the subjects reported any non-correctable impairments in perception or limitations in mobility. The user study was IRB approved.

4.6 Quantitative Results

A Shapiro-Wilks test was used to test for normality and a Grubb's test found one outlier, which was for selection accuracy with controller teleportation, that we replaced with the mean. For our comparative analysis we analyzed: (1) efficiency as the total time to get to all waypoints; (2) accuracy as the total number of teleports used; and (3) selection accuracy as the distance between each waypoint and the destination of the first teleport performed when that waypoint became active (i.e. how close the user got to each waypoint on the first try). 4.1 lists the total results. For stomp teleportation, an average of 45.0 stomps with a SD of 7.9 were recorded; for voice teleportation, there were an average of 37.42 voice commands recorded with a SD of 8.1. Though not all of these commands were necessarily recognized, there are still a greater number of teleportations on average, indicating there were multiple false positives.

Method	Time	SD	# Teleports	SD	Accuracy	SD
Controller	81.62	11.3	26.81	2.3	19.71	6.0
Dwell	100.80	9.7	24.69	.4	14.31	3.0
Stomp	109.39	38.0	50.19	9.8	100.39	29.2
Voice	137.92	28.9	24.56	.6	19.42	10.4
Blink	89.42	15.7	26.31	2.4	22.12	10.6

Table 4.1: Quantitative results for each teleportation method.

There was a statistically significant difference between teleportation methods for total time as determined by a repeated measures ANOVA ($F_{4,60}$ = 13.96, $p < .0001$). A Tukey posthoc analysis showed a significant difference between controller and voice ($p < .01$), controller and stomp ($p < .05$), blink and voice ($p < .01$), dwell and voice ($p < .01$), and voice and stomp ($p < .01$). For the total number of teleports used to get to the waypoint, the same ANOVA found a significant difference between teleportation methods ($F_{4,60}$ = 73.25, $p < .0001$). A Tukey Post hoc test found significant differences between controller and stomp ($p < .01$), blink and stomp ($p < .01$), dwell and stomp ($p < .01$), and voice and stomp ($p < .01$). Selection accuracy was calculated to be the Euclidean distance between where the first teleportation cursor was placed and the waypoint that the participants needed to reach. If users used more than one teleport to get to the waypoint (such as when over or undershooting), we only used the data from the first teleportation they made to calculate this distance, since those following are assumed to be corrections. For selection accuracy, a one-way ANOVA found a significant difference between teleportation methods ($F_{4,60}$ = 99.41, $p < .0001$). A Tukey Post hoc test found significant differences between controller and stomp ($p < .01$), blink and stomp ($p < .01$), dwell and stomp ($p < .01$), and voice and stomp ($p < .01$).

4.7 Qualitative Results

We asked each user to rate every method they tested (including controller) in terms of efficiency, learnability, accuracy, and likability using a 5 point Likert scale. Usability is generally decomposed into these four attributes [42]. Questions were phrased in the following manner: "Teleporting using a controller was efficient" and "Teleporting using Blink was easy to learn". The results are summarized in 4.3.

A Kruskal-Wallis test found a significant difference for efficiency (χ^2 = 21.92, p < .01), learnability (χ^2 = 18.29, p < .01), accuracy (χ^2 = 21.85, p < .01) and likeability (χ^2 = 16.48, p < .01). For efficiency, a Mann-Whitney post-hoc tests found a significant difference in Likert scores between controller and voice (p < .05), controller and stomp (p < .05), controller and dwell (p < .05), blink and voice (p < .05), blink and stomp (p < .05), and blink and dwell (p < .05). For learnability, a Mann-Whitney post-hoc tests found a significant difference in Likert scores between controller and voice (p < .05), controller and stomp (p < .05), controller and dwell (p < .05), blink and stomp (p < .05). For accuracy, a MannWhitney post-hoc tests found a significant difference in Likert scores between controller and stomp (p < .05) and blink and stomp (p < .05). For likeability, a Mann-Whitney post-hoc test found a significant difference in Likert scores between controller and voice (p < .05), controller and stomp (p < .05), blink and dwell (p < .05), blink and stomp (p < .05), and blink and voice (p < .05).

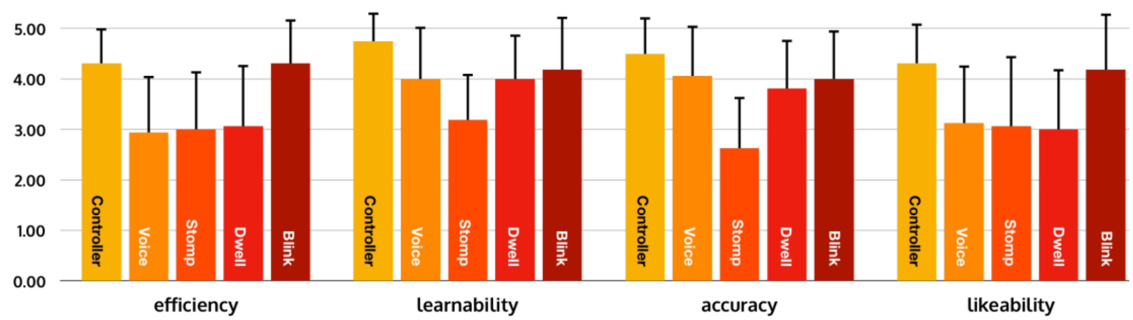

Figure 4.3: These columns show the Likert scores (ranging from 1 to 5) for each teleportation method based on the following criteria: : efficiency, learnability, accuracy, and likeability. Error bars indicate the standard deviation.

CHAPTER 5
DISCUSSION AND FUTURE WORK

5.1 Discussion

We expected controller teleportation to be the fastest method due to the fact that participants were most familiar with this method. While controller teleportation did have the lowest average time, for dwell and blink we found no significant differences in total trial time compared to using a controller. Voice commands ended up being the slowest method for activating teleport. This likely occurred because of a combination of factors; the command itself can take longer to vocalize than a button press, blink, and possibly stomp, and the time was further increased by the noticeable latency of the speech recognition software. Cases where the participant spoke the command but it was not recognized (false negatives) also contributed to the time. When looking at total number of teleports issued as one measure of accuracy, we observed that stomp required approximately twice as many teleports as when using the other techniques, which had no significant differences compared with each other. For stomp and voice, we must also take false positives/negatives into consideration. The average number of stomp commands recorded by the observer was 45.0 (SD=7.9), which, while high, is still below the average number of recorded teleports. This indicates that there were a number of false positives, which would negatively impact accuracy. The average number of voice commands recorded by the observer was 37.42 (SD=8.1), which is much higher than the number of teleportations. This indicates that there were a large number of instances in which the participant spoke the command word, but was not detected by the speech recognition and did not teleport (a "false negative"), forcing them to try again; this would negatively impact the participant's time. Though in preliminary studies both stomp and voice seemed to work with a good accuracy, some participants in our study were observed to struggle activating them reliably which led to the issues described.

There is a significant discrepancy between qualitative and quantitative results. For efficiency, participants deemed both controller and blink teleportation to be faster than voice, dwell, and stomp, which had no significant differences in rating when compared to each other. Quantitative results differed, as every method was significantly faster than voice, and the only other significant difference found was between controller and stomp teleportation. This may be because of the nature of the teleportation methods; blink and controller teleportation both had high accuracy and are effectively instantaneous, which means participants can effectively teleport as they please which may make the methods feel efficient. Dwell, on the other hand, is a more passive technique and requires the user to wait, which may make the technique feel like it's taking longer than it really is. Stomping can be very quick, but the low accuracy and false positives may have made it appear to take longer. Regarding accuracy, controller and blink were found to be more accurate than stomp which is consistent with the quantitative results on number of teleports issued and selection accuracy.

Stomp showed a significantly lower selection accuracy than other methods. This was largely because participants ended up stomping quite vigorously to improve stomp detection, but this then unintentionally shifted the location they focused their gaze on and they would often end up next to the waypoint, ass opposed to teleporting inside it. As a consequence, participants had to issue far more teleports –nearly twice as much as other methods– to successfully get to the waypoints. Another problem is that participants issued 45.0 stomps (SD=7.9) while 50.19 (SD=9.8) teleports were detected, which indicates a high ratio of false positives. This was because some participants selected a very low threshold for stomp detection; the result that unintentional head movements, or large head movements when scanning the virtual world for the next waypoint would lead to a false detection of a stomp. This problem could be avoided by using a higher threshold setting for foot stomp detections. Qualitative results also confirmed that stomp had the lowest learnability and accuracy.

For voice teleportation, participants on average issued 37.42 (SD=8.1) voice commands, while 24.56 (SD=.6) teleports were issued, which indicates quite a high ratio of false negatives, where the command was not detected properly. The high standard deviation indicates this was not a problem for all participants (minimum: 27, maximum: 54). One possible explanation could be that several participants were not native English speakers and they they might have had trouble correctly pronouncing the word "Okra". On the other hand, speech recognition on a mobile devices are not known for having a 100% accuracy. Because the smartphone was enclosed in the VR adapter, it may not have allowed for the full utilization of noise cancellation, which could have improved speech recognition accuracy using the smartphone's second speaker. It may also have simply muffled the sound enough to prevent recognition if participants were not speaking loudly.

Our study did not find a significant difference in performance between dwell and controller teleportation. This result contradicts prior results [36] that found that eye gaze selection is slower than controller selection for objects that were far away. Besides a difference in techniques (head versus eye gaze) this prior study used dwell but didn't specify a dwell time used, which makes a comparison difficult. The performance of dwell could be improved by using a lower dwell time but this will exacerbate the Midas touch problem. We only showed a teleportation cursor when the user's gaze raycast intersected with the ground plane, and was within the 15m teleportation range. This strategy helped to mitigate the Midas touch problem, since there were almost no false positives (e.g., the average number of teleports issued 24.69 was close to the number required: 24). Given that dwell doesn't require any instrumentation and can already be widely implemented on both mobile VR and PC VR; it seems to be a feasible alternative to using a controller, despite user preference for blink and controller teleportation. This method may be particularly valuable for users with physical impairments. Controller and stomp teleportation may be difficult or impossible for people who have lost limbs or have severe motor impairment; voice teleportation has the issue of speech recognition and false negatives, and it may be difficult to implement on PC headsets if they do not have microphones attached; and blink detection is not yet widely available. Dwell does not have these problems and can thus be used by people with disabilities preventing the use of a controller.

Blink performed well in terms of performance, accuracy, learnability, and was –together with controller input– liked the most by participants. Because no significant differences in performance, selection accuracy and total teleports were found, blink is the best viable alternative to a controller from all of the hands-free methods we evaluated. The value we used for eye blink detection (170ms) seemed to work well for most participants, as this value was long enough to prevent normal blinking from causing teleportation in most instances, but short enough that participants did not have to greatly exaggerate their blinking. While there were some false positives according to the participant surveys, there were very few and most of them did not appear to occur from regular blinking but were instead the result of errors in the Fove's eye detection. It is worth noting that in testing the blink implementation before the study, one individual was getting false positives quite frequently even while holding his eyes open; the effectiveness of the eye-/blink tracker can have a significant impact on the performance of the technique regardless of the blink threshold. Blink offered a slightly faster performance (not significant) than dwell since it lets users teleport without a dwell time. From a practical point of view, Blink can already be offered on VR headsets that feature an integrated eye tracker such as the Fove [34]. It is possible to implement blink detection for mobile VR [39], so this should be a viable technique for mobile as well, but dwell may still be preferable because it is much simpler to implement.

A limitation of this study is that we only used a navigation task. Many VR applications, such as games, also use a controller for shooting enemies and/or interaction with objects. It has been suggested that overloading the hands with navigation functionality may increase cognitive load and decrease efficiency [29]. Another limitation is that this study focused on teleportation at close distances; differences in accuracy may become more pronounced at longer distances. Similarly, there may be performance changes in environments with different elevations, while this study used only a flat plane. Hardware was also a minor limitation, making implementation of each technique on a single device excessively difficult with currently available devices.

5.2 Future Work

Future studies will assess the benefits of hand-free navigation in addition to performing other tasks with a controller which would most likely show benefits in terms of performance and cognitive load while using hands-free methods over using a controller. Our study was unable to get blink detection to work reliably on the Google Daydream platform, so in a future study we would like to have all of the conditions on the same device. We may integrate a low-cost lights sensor into a VR headset with controller compatibility and place this close to the eye for detecting eye blinks; this might resolve the illumination issues we experienced from trying to detect a change in light reflection from the eye. This would allow us to test dwell, blink, and controller teleportation on the same device, which are the methods of greatest interest based on our results. Other possible future studies could investigate how these hands-free methods compare with controller teleportation at longer distances, or in environments that are not purely flat.

Though hand-free teleportation could reduce arm fatigue, our study focused on performance and accuracy of teleportation techniques that can be implemented on mobile VR platforms due to their large potential to bring VR to the masses. In future studies, we aim to investigate hands-free teleportation on PC VR platforms which typically feature heavier controllers that can be more precisely tracked, which would provide a better insight into how prolonged use of mid-air gestures could cause gorilla arm syndrome and how hands-free methods could mitigate this. Because VR can offer immersive out-of-body experiences for those with limited mobility [43], future work will also evaluate hands-free teleportation methods with users who have severe motor impairments (e.g., individuals who are quadriplegic or who have cerebral palsy) to identify whether this would enable them to access VR.

CHAPTER 6
CONCLUSION

Teleportation is a popular form of locomotion in VR, but currently most teleportation methods rely on a controller, which may not be available on mobile VR platforms. Use of a controller for navigation could also lead to arm fatigue, especially if done regularly for a long duration. This paper evaluates four hands-free teleportation methods, where users select a location to teleport to using head gaze and activate teleportation by dwelling their gaze, stomping a foot, uttering a voice command or blinking. An empirical study with 16 participants collected data on the performance and accuracy of these methods and compared them to using a controller. There were no significant differences in efficiency and accuracy between blink, dwell and using a controller, suggesting that the blink and dwell techniques may be viable hands-free alternatives. Voice performed worst for efficiency, and stomp performed worse for selection accuracy; both methods also had high rates of false determinations: false positives in the case of stomp, and false negatives (where a command was issued but not detected) for voice. Participants liked using blink and a controller the most with no difference between them. Our study shows that blink and dwell are feasible hands-free alternatives to using a controller for teleportation.

BIBLIOGRAPHY

[1] Martin Usoh, Kevin Arthur, Mary C Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P Brooks Jr. 1999. Walking> walking-in-place> flying, in virtual environments. In *Proceedings of the 26th Conference on Computer Graphics and Interactive Techniques*, pages 359–364.

[2] Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. 2004. *3D user interfaces: theory & practice.* Addison-Wesley.

[3] Eelke Folmer, Jiwan Bhandari, Paul MacNeilage. 2018. Teleportation without Spatial Disorientation using Optical Flow Cues. In *Proceedings of Graphics Interface 2018 (GI'18).*

[4] Kyt Dotson. 2015. Silicon Angle: Oculus co-founder says mobile VR will dominate before PC VR goes wireless, `http://siliconangle.com/blog/2015/11/02/oculus-co-founder-says-mobile-vr-will-dominatebefore-pc-vr-goes-wireless/`. Last accessed November, 2015.

[5] Sam Tregillus and Eelke Folmer. 2016. VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, pages 1250–1255.

[6] Majed Al Zayer, Sam Tregillus, and Eelke Folmer. 2016. PAWdio: Hand Input for Mobile VR Using Acoustic Sensing. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '16)*. ACM, New York, NY, USA, pages 154–158. DOI: `http://dx.doi.org/10.1145/2967934.2968079`

[7] Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009 Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*. ACM, pages 161–168.

[8] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems.*. ACM, pages 1063–1072.

[9] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling Cumulative Arm Fatigue in Mid-Air Interaction based on Perceived Exertion and Kinetics of Arm Motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.*. ACM, pages 3328–3339.

[10] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle arms-down mid-air interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology.*. ACM, pages 63–71.

[11] A. Cockburn, P. Quinn, C. Gutwin, G. Ramos, and J. Looser. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *. International Journal of Human-Computer Studies*. 69, 6 2011, pages 401–414. DOI: `http://dx.doi.org/10.1016/j.ijhcs.2011.02.005`

[12] Leap Motion | Mac & PC Motion Controller for Games, Design, Virtual reality `https://www.leapmotion.com/`.

[13] Andreas Linn. 2017. *Gaze Teleportation in Virtual Reality* Master's thesis. HCI Master Thesis CSC KTH.

[14] Ben Lang. 2018. Hands-on: Infinadeck's Latest Prototype is the Most Natural Feeling VR Treadmill Yet, `https://www.roadtovr.com/infinadeck-2018-prototype-hands-on-most-natural-feeling-vr-treadmill-yet/`, Last accessed on Jul 10, 2018.

[15] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. 2001 Redirected walking. In *Proceedings of EUROGRAPHICS*, volume 9, Citeseer, pages 105–106.

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

[17] Jan L Souman, P Robuffo Giordano, M Schwaiger, Ilja Frissen, Thomas Thümmel, Heinz Ulbrich, A De Luca, Heinrich H Bülthoff, and Marc O Ernst. 2011. CyberWalk: Enabling unconstrained omnidirectional walking through virtual environments. *ACM Transactions on Applied Perception (TAP)*, 8, 4 2011, page 25.

[18] Steffi Beckhaus, Kristopher J Blom, and Matthias Haringer. 2007. ChairIO-

the chair-based Interface. *ACM Concepts and technologies for pervasive games: a reader for pervasive gaming research*, 1 2007, pages 231–264.

[19] Hiroo Iwata, Hiroaki Yano, and Hiroshi Tomioka. 2006. Powered shoes. In *ACM SIGGRAPH 2006 Emerging technologies*. ACM, page 28.

[20] Kiran J Fernandes, Vinesh Raja, and Julian Eyre. 2003. Cybersphere: the fully immersive spherical projection system. *Communications of the ACM*, 46, 9 2003, pages 141–146.

[21] Sam Tregillus, Majed Al-Zayer, and Eelke Folmer. 2017. Handsfree Omnidirectional VR Navigation using Head Tilt. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM.

[22] James Templeman, Patricia S. Denbrook, and Linda E. Sibert. 1999. Virtual Locomotion: Walking in Place through Virtual Environments. *Presence*. 8, 6 Dec 1999, pages 598–617. DOI: `http://dx.doi.org/10.1162/105474699566512`

[23] Niels Christian Nilsson, Stefania Serafin, and Rolf Nordahl. 2013. The perceived naturalness of virtual locomotion methods devoid of explicit leg movements. In *Proceedings of Motion on Games*. ACM, pages 155-164.

[24] Richard C Fitzpatrick, Jane E Butler, and Brian L Day. 2006. Resolving head rotation for human bipedalism. *Current Biology*, 16, 15 2006, pages 1509–1514.

[25] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. 2016. Point and teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '16)*. ACM, New York, NY, USA, pages 205–216. DOI: `:http://dx.doi.org/10.1145/2967934.2968105`

[26] Beverly K Jaeger and Ronald R Mourant. 2001. Comparison of simulator sickness using static and dynamic walking simulators. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications, volume 45, pages 1896–1900.

[27] Doug Bowman, David Koller, Larry F Hodges. 1997. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *Virtual Reality Annual International Symposium, 1997., IEEE 1997*, IEEE, pages 45–52.

[28] Niels H Bakker, Peter O Passenier, and Peter J Werkhoven. 2003. Effects of head-slaved navigation and the use of teleports on spatial orientation in virtual environments. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 45, 1 2003, pages 160–169.

[29] Joseph J LaViola Jr, Daniel Acevedo Feliz, Daniel F Keefe, and Robert C Zeleznik. 2001. Hands-free multi-scale navigation in virtual environments. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM, pages 9–15.

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

[31] Steffi Beckhaus, Kristopher J Blom, and Matthias Haringer. 2005. Intuitive, hands-free travel interfaces for virtual environments. In *New Directions in 3D User Interfaces Workshop of IEEE VR*, pages 57–60.

[32] Daniel Zielasko, Sven Horn, Sebastian Freitag, Benjamin Weyers, and Torsten W Kuhlen. 2016. Evaluation of hands-free HMD-based navigation techniques for immersive data analysis. In *3D User Interfaces (3DUI), 2016 IEEE Symposium on*, IEEE, pages 113–119.

[33] Benjamin Bolte, Frank Steinicke, and Gerd Bruder. 2011. The jumper metaphor: an effective navigation technique for immersive display setups. In *Proceedings of Virtual Reality International Conference*.

[34] FOVE Eye Tracking Virtual Reality headset: `https://www.getfove.com`.

[35] Linda E Sibert and Robert JK Jacob. 2000. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems.* ACM, pages 281–288.

[36] Nathan Cournia, John D Smith, and Andrew T Duchowski. 2003. Gaze-vs. hand-based pointing in virtual environments In *CHI'03 extended abstracts on Human factors in computing systems.* ACM, pages 772–773.

[37] Sophie Stellmach and Raimund Dachselt. 2012. Designing gaze-based user interfaces for steering in virtual environments In *Proceedings of the Symposium on Eye Tracking Research and Applications.* ACM, pages 131–138.

[38] 2016. VR-STEP; Unity Plugin that enables walking-in-place on mobile

VR, https://www.assetstore.unity3d.com/en/#!/content/60450. March 2016.

[39] Luca Mefisto. VR blink detection http://mefistofiles.com/vr-blink-detection

[40] *Blink and you miss it!*. 2006. http://www.ucl.ac.uk/media/library/blinking

[41] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems.* ACM, pages 1063–1072.

[42] Eelke Folmer and Jan Bosch. 2004. Architecting for usability; a survey *Journal of Systems and Software*, 70, 1 2004.

[43] Youtube: Walking with Reality, https://www.youtube.com/watch?v=Nxml3p5fbmU