

Improving Walking in Place Methods with Individualization and Deep Networks

Sara Hanson*

University of Southern California, USA

Richard A. Paris[†]

Vanderbilt University, USA

Haley A. Adams[‡]

Vanderbilt University, USA

Bobby Bodenheimer[§]

Vanderbilt University, USA

ABSTRACT

Walking in place is a standard method for moving through large virtual environments when physical space or positional tracking is limited. This technique has become increasingly prominent with the advent of mobile virtual reality in which external tracking may not be present. In this paper, we revisit walking in place algorithms to address some of their technical challenges. Namely, our solutions attend to improving starting, stopping, and speed control for individual users. From a hand-tuned threshold based algorithm, we provide a new, fast method for individualizing the walking in place algorithm based on biomechanic measures of step rate. In addition, we introduce a new walking in place model based on a convolutional neural network trained to differentiate walking and standing. Over two experiments we assess these methods against a traditional threshold based algorithm on two mobile virtual reality platforms. The assessments are based on controllability, scale, and presence. Our results suggest that an adequately trained convolutional neural network can be an effective way of implementing walking in place.

Keywords: Virtual environments, locomotion, walking in place, convolutional neural network, perception

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; J.4 [Computer Applications]: Social and Behavioral Sciences—Psychology

1 INTRODUCTION

Moving through a virtual environment (VE) is one of the most important interactions provided by virtual reality (VR). Although natural methods of locomotion are desirable in immersive VEs [40], in practice challenges such as limits to the size of a physically tracked space [37, 46] often make them infeasible. In this paper, we study locomotion methods that comply to an analogous technical constraint—a complete lack of position tracking. In particular, we focus on mobile VR as presented through inexpensive systems such as the Samsung Gear, Google Daydream, and Oculus Go. This family of devices provides only orientation tracking from the head-mounted display (HMD). As a result, their applications frequently forgo locomotion in VR entirely.

Traditional hand-held controllers can serve as an alternative method for locomotion. However, they have repeatedly been demonstrated to increase the effects of simulator sickness [26, 28], making them a problematic solution for locomotion in VR. Other locomotion techniques that do not involve peripheral devices, like buttons or controllers, exist for mobile VR. Often, these techniques involve gestures such as leaning or proxy walking [9, 13, 14], or they might involve arm swinging [19, 47]. In the current paper we study walking in place (WiP), perhaps the most common of these techniques, as it

does not require position tracking and it has certain key advantages. First, WiP proxy gestures are confined to the lower limbs, leaving the hands free to interact with the VE in other ways (e.g., using the buttons of the Samsung Gear or a paired controller for the Oculus Go). Second, WiP techniques provide similar proprioceptive and kinesthetic feedback to that experienced by real walking [23], which is advantageous in acquiring spatial knowledge over more passive techniques [25, 31].

Yet WiP methods are not widely used. This is largely due to performance issues. Whitton et al. [44] found that their WiP method was the worst of the locomotion methods that they compared; they concluded that their implementation was poor. More recently, Zielasko et al. [50] found that a seated WiP method performed poorly for locomotion, whereas Bozgeyikli [3] found positive but middling performance. Such results suggest that implementing WiP properly is non-trivial. WiP methods typically suffer from three problems: starting and stopping latencies, velocity control, and control of direction [45]. In this paper, we focus on the first two, and do not deal with the third. Direction control is an important consideration. However, given mobile VR’s lack of peripheral devices, we foresee no easy solution outside of a gaze-directed or audio interfaces. We elect to use a gaze-directed direction control similar to those implemented in many other WiP interfaces.

Our goal is to improve WiP algorithms so that they are viable and easy to use locomotion solutions for mobile VR. Latencies are a known issue in WiP algorithms, because it is difficult to detect when a person starts to walk in place and consequently when to initiate motion or optic flow in an HMD. Similarly, it can be difficult to determine when to cease motion in the WiP algorithm when a proxy motion has stopped. Solutions to walking in place that rely on external tracking sensors can mitigate this problem [8, 43]. However, solutions that rely solely on internal inertial measurement units (IMUs) typically have considerable latency, because IMUs are noisy and filtering them appropriately can cause slow system responsiveness.

For velocity control, the challenge is in determining the rate of optical flow during proxy motion. Optic flow should be associated with one’s step rate during walking in place in order to match one’s perception of self-motion. However, the perception of self-motion based on gait is complex, not fully understood, and it varies depending on the locomotion context (e.g., real walking or treadmill [7]). Nilsson et al. [20] have studied the relationship between step rate and optic flow gain for WiP, but taking this into account is complex as it depends on field of view.

In the current work, we propose two novel, individualized WiP algorithms. We then compare these individualized methods to a tuned threshold-based method. Our first algorithm uses an individual’s biomechanical information to customize thresholds for walking detection. The second extrapolates walking in place patterns for walking versus standing across a large data set using a convolutional neural net (CNN). Both solutions seek to compensate for the variability seen across individuals when walking in place. We implement these methods on two commercially available mobile VR platforms, the Samsung Gear and the Oculus Go. We then assess both the methods and platforms for controllability using an evaluation approach inspired by Whitton et al. [44] and a new evaluation technique based

*e-mail:sdhanson@usc.edu

[†]e-mail:richard.a.paris@vanderbilt.edu

[‡]e-mail:haley.a.adams@vanderbilt.edu

[§]e-mail:bobby.bodenheimer@vanderbilt.edu

on distance estimation.

2 BACKGROUND

2.1 Locomotion in Virtual Environments

There are several reviews of locomotion techniques in VR that concentrate on walking in place methods [2, 24, 45]. Our current discussion focuses on mobile VR solutions without external position tracking systems.

HMDs and smartphones track orientation through the use of inertial measurement units (IMUs), which extract linear velocity and acceleration as well as angular velocity and acceleration. IMUs have been used to detect leg movement by attaching them to a user’s shins [8, 12, 41] and to a user’s arms to detect arm swinging motion [18, 47] during WiP. Wendt et al. [42] approximated a head bobbing model from head tracking data. Tregillus and Folmer [38] implemented VR-STEP with a real-time pedometry approach using a smart phone’s inertial sensors. And Paris et al. [25] set thresholds between acceleration values in the Samsung Gear VR to detect forward walking. Our work also uses the IMU incorporated in an HMD alone to detect WiP motion.

With the exception of the system proposed by Wendt et al. [42], prior solutions for WiP have used preprogrammed thresholds to detect movement. In some users, this is an effective method, and it works in real time. However, there is notable variance in performance across individuals. Just as gait characteristics vary, walking in place can express itself in different movement patterns depending on the posture and physical exertion of the individual. Wendt et al [41–43], in contrast, draws from biomechanics literature to model forward velocity as a quadratic function of step rate and height. In particular, their work analyzes differences in gait cycles between real walking and walking in place to develop better WiP solutions. See Figure 1 for an illustration of gait cycles for real walking versus walking in place. Nilsson and colleagues used empirical techniques to determine velocity matches for stepping rates during WiP motions based on numerous factors [20–22]. In our work we wish to individualize these ideas further by tuning the velocity to what each subject might expect from a given step rate.

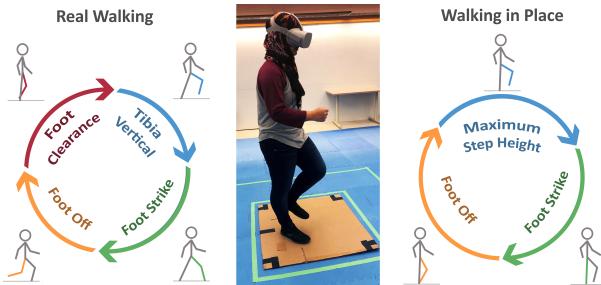


Figure 1: Simplified gait cycles based on Wendt et al. [43] for real walking (left) and walking in place (right) are shown. Center, a participant performs WiP on top of cardboard to prevent drifting.

2.2 Machine Learning for Walking In Place

To the best of our knowledge, a pattern recognition approach has not been attempted in earnest since early walking in place research. In Slater et al.’s [36] original WiP algorithm, a simple feed-forward neural network was implemented to extract patterns from the position tracked information of a head-mounted display. Usoh et al. [40] then used the same neural net to evaluate WiP against real walking and flying in virtual environments. Razzaque et al. [29] then leveraged the same neural network approach to analyze head motion data from a position tracker to determine when the participant was walking in place in their Redirected WiP method.

One form of deep learning, the convolutional neural network (CNN), has performed well for many human activity recognition problems [15, 48, 49]. As demonstrated in Yang et al. [48], CNNs can reliably model local patterns, they are straightforward to train, and they are generalizable since they require little domain specific knowledge about the activities of interest. In particular, we chose CNNs in this paper because of their suitability in classification prediction problems and because they are usually easier to train — our training data will be small by many machine learning standards.

2.3 Distance Estimation

The scale of space and how people perceive distance through HMDs have been issues of long-standing concern in virtual environments research. See Renner et al. [30] and Creem-Regehr et al. [5] for surveys on these issues. Recent work by Buck et al. [4] and Kelly et al. [11] examine how distances are perceived in modern commodity level HMDs. We anticipate that the devices used in this paper—the Samsung Gear VR and Oculus Go—would perform comparably to contemporary devices, such as the Oculus Rift CV1. And, in fact, both devices share hardware and software components with the Oculus Rift CV1. However, to our knowledge, no one has thoroughly examined spatial perception in these devices.

The goal of that body of work is to determine how spatial relations in a virtual environment are perceived by users wearing an HMD; more specifically, the idea is to understand how users estimate absolute egocentric distance. The method used for such estimation is commonly blind-walking, and the idea behind such a task is that users are performing a task in which the action (walking) is dependent on the perceived distance to an object. Judgments of absolute egocentric distance have been shown to be accurate in the real world when assessed with blind walking up to at least 15m [17, 32].

In this paper, we take a perceived distance to an object and then vary the task used to assess the perceived distance. That is, the task will either be blind walking or blind walking in place. If our walking in place algorithm is equivalent to real walking, then our results should match. Thus, we measure the response of our walking in place system by assessing how close the action of blind walking is to the actions of blind walking in place, where we vary the parameters of the walking in place algorithms.

3 RESEARCH DESIGN

Our current work presents three unique approaches to body-based turning methods for WiP in which steering is indicated by head orientation. Each technique infers linear motion in the direction of a subject’s gaze when walking. Thus, turning or rotation in all such mobile VR systems is head or gaze-based. Linear motion is extracted from the inertial measurement unit (IMUs) from the Oculus Go and the Gear VR systems. Using different devices may allow us to discern performance discrepancies between different hardware and software solutions for WiP. The IMU used in the Samsung Gear VR is less sensitive than that of the Go, and it reports signals using different units (g ’s as opposed to m/s^2), resulting in different threshold values between devices.

Our methods use linear acceleration from the bobbing motion of a user’s head when walking in place to impart motion in the direction of their gaze. This is in contrast to the conventional pedometer approach. Pedometers use pattern analysis techniques to recognize repeating motions that are assumed to be steps. The repetition requirement of pedometers leads to either unreliable detection when the number of repetitions is low or severe lag when the number of repetitions is high. Both behaviors are undesirable for comfortable locomotion in immersive virtual environments.

3.1 Threshold Based Motion

The first method is derived from that of Paris et al. [25]. The state machine consists of three states, one which is standing and two

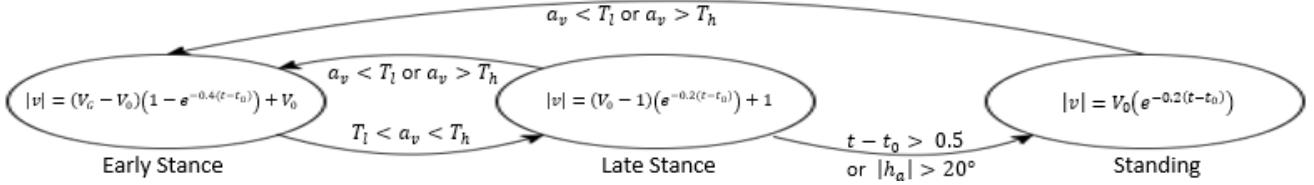


Figure 2: V_G is the target velocity (2.0 for threshold based WiP). T_l and T_h are thresholds for vertical acceleration and set to 0.75 m/s/s. h_a is the head angle of the subject. V_0 and t_0 are set to the current velocity and time on each transition. The state machine above controls the velocity to an average speed of roughly 1.65 m/s.

which constitute walking. Walking is represented by two repeating states: the early stance and late stance states. The state machine and associated parameters were constructed with the conditions displayed in Figure 2. Users are considered to be stepping when the magnitude of the vertical direction of acceleration, a_v is greater than thresholds reported in [25] (0.075g or 0.75m/s²) and not stepping otherwise. These thresholds are denoted T_l and T_h in the figure. In this implementation, $T_l = -T_h$.

We desired an average walking pace of 1.65 m/s (see Nilsson et al. [20] for this rate). Upon initial detection of WiP motion, the early stance state is entered, and the magnitude of the velocity of the user (optic flow rate) is exponentially decayed in an increasing way, (i.e., $1 - e^{-kt}$) to a maximum speed where it remains until this state is exited. The maximum goal speed, V_G , is set to $1.25 \cdot 1.65$. We multiplied our desired average walking speed by 1.25 because numerical simulations indicated that this constant resulted in an overall velocity approximately equal to 1.65 m/s. The early stance stage transitions to the late stance state when the magnitude of the vertical acceleration becomes less than the thresholds (T_l and T_h). In the late stance state, speed is exponentially decayed to 1 m/s. If no vertical acceleration is detected for 0.5s, then the standing state, in which velocity is decayed to zero, is entered. Constants for the exponentials in all states were derived by trial and error in Paris et al. [25] and they remain unchanged in the current work. These values included: 0.4 for the growth rate, 0.2 for the decay rate, and 0.5s for the exit criteria. The other exit condition from the late stance stage can occur if the pitch or roll of the head exceeded 20° (denoted by h_a in Figure 2). We included this condition to prevent strafing, reduce simulator sickness, and stop motion when we detect a subject is visually searching. Since our system only translates in the gaze direction, visually searching while moving is not recommended.

3.2 Biomechanics-Based Motion

Our second approach to walking in place customizes the state machine of Figure 2 to an individual's step rate and real walking rate. It is a three step process that is data-driven based on an individual's measurements. Prior work [42] modeled speed as a quadratic function of step rate. In our method we use virtual step rate as a replacement of real step rate. We define virtual step rate as the average time between successive "step peaks" in our IMU data stream. A step peak occurs when WiP head motion creates a spike in head acceleration followed by a local minimum in head acceleration. To determine the appropriate thresholds for these peaks (step one) we measure a user's head acceleration in the up and down direction for 20 steps at varying WiP speeds (slow, medium, fast). We visually identify the 20 peaks, and we set the upper threshold to the minimum value of the 20 local maxima and lower threshold to the maximum value of the 20 local minima. Extraneous local peaks within 0.25 seconds of a greater local peak are considered noise. Based on this measurement, we determined T_l and T_h within the state machine for an individual for a given HMD.

Next, (step two), we have users step again for 10 seconds at slow, medium, and fast rates, and we detect their step rate using these

thresholds. When a user passes both the high and low thresholds in 0.25 seconds, we say a step has occurred. We determine an average step rate for each of these three step rates, r_s , r_m , and r_f , given a particular headset. Next, we have users walk 10m down a hallway at a slow, medium, and fast pace, and we measure their average walking speed, v_s , v_m , and v_f . In prior work [8, 43] velocity was a quadratic function of stepping rate and height only. This quadratic has the form

$$v(r) = ar^2 + br + c \quad (1)$$

where a , b , and c are coefficients mapping virtual step frequency r to real world speed v . Using our three rate-velocity pairs for an individual, we determine these coefficients for a particular user on a particular headset.

Finally (step three), we take the algorithm as presented in Section 3.1 and the state machine of Figure 2 and modify it for the current WiP method. V_G is then based on the result of Equation 1, and is set to $1.25v(r)$ so that the system will have an average speed that roughly matches $v(r)$. Additionally, T_l and T_h are modified as described earlier in this section. Note that they are no longer symmetric as in the case of the previous model. Also note that the state machine is customized for a particular individual and a particular headset.

3.3 Convolutional Neural Network

The goal of our CNN WiP solution was to account for individual differences in motion across users with a deep learning approach. Given a sequence of triaxial linear acceleration values, our CNN WiP solution determines the probability of the current state of the user in real time. Specifically, we formulate WiP as a classification problem for walking and standing states. The CNN modifies but works with the basic state machine of Figure 2. In the next sections, we first describe the system architecture. Then, we discuss the preprocessing steps taken for data set generation that are specific for our current problem, our training results, and how the CNN WiP solution operates in VR.

3.3.1 Architecture

All CNNs consist of at least one temporal convolution layer, one pooling layer, and one fully connected layer in a neural network. Inspired by the architecture and optimization techniques suggested by Saeed et al. [33], our CNN WiP architecture uses two depthwise convolution layers, which are separated by a max pooling layer for dimensionality reduction. Each convolution layer applies linear convolution to the input layer, adds a constant bias, and then applies an activation function to produce an output. Rectified linear units (ReLUs) were selected for the activation function, given their nonlinearity and their reduced likelihood of vanishing gradient. The pooling layer downsamples along the spatial dimension of the input, reducing the number of parameters. The truncated normal distribution is used to initialize the weights of the neural network as normal initialization has been demonstrated to alleviate the vanishing and exploding gradient problems experienced by artificial

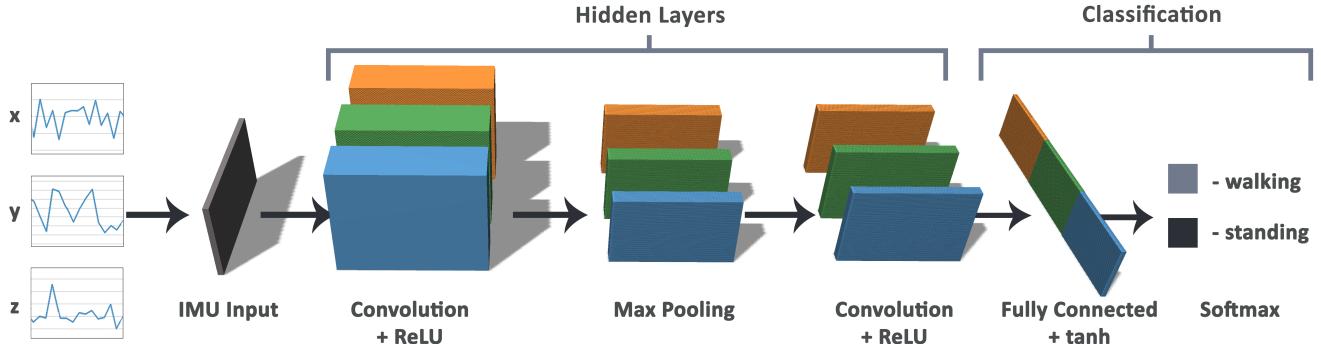


Figure 3: Illustration of the CNN architecture used for walking in place activity detection. Our method uses a sequence of IMU data as input for the network, which is trained to recognize walking and standing actions.

neural networks. In our model, we have used max pooling to apply a maximum filter to subregions of the first convolutional layer output.

The remainder of the architecture is similar to other neural networks. The feature matrices generated by the previous operations are combined and flattened to generate a vector, which is the fully connected layer. And the tanh function is applied to introduce non-linearity. Finally, the softmax function is applied to the final layer to produce the probability distribution of the activity states (i.e., walking or standing). Figure 3 illustrates our entire system pipeline.

3.3.2 Dataset

To create training data for the CNN, seven research assistants were enlisted to perform specific actions while wearing each HMD (Oculus Go and Samsung Gear VR). Five research assistants recorded walking consecutively for 20 minutes in each headset. Two assistants recorded standing for 20 minutes in each headset. The IMU for each device was polled every 0.017 seconds for triaxial data with an application developed in Unity. IMU data was extracted using the Unity API and for the Gear and the OVR API for the Go. Although this data was likely cleaned, we were not privy to the exact techniques employed by the API.

The data was labeled ‘walking’ or ‘standing’ at each time step. 6044.08 total seconds of walking and 2061.78 seconds of standing data were collected for the Go. 5541.37 total seconds of walking and 2035.5 seconds of standing data were collected for the Gear. The x, y, and z linear acceleration values were aggregated at each time step and grouped into time sequences. This generated $N \times 3$ input tensors for the CNN, where N indicates input width (number of seconds times sampling rate).

3.3.3 Training

The Go and Gear systems used different IMUs. As a result, the linear acceleration values between the devices varied. In comparison to the Go, the Gear output created lower and less variable values. To compensate for the differences between the two IMUs, 40 consecutive triaxial datapoints were used for the Oculus Go CNN and 60 consecutive triaxial datapoints were used for the Gear VR CNN to construct input vectors. Each device was trained on its own data.

While the same architecture was used for both the Gear and Go, the hyperparameters for each solution differed in order to obtain comparable performance between devices. The first convolutional layer was applied with a window size of 20 for the Go and 30 for the Gear with a ReLU activation function. The max pooling layer for both solutions used a stride of 2. However, the pooling filter was of size 10 for the Go and size 20 for the Gear. The second convolution filter used a kernel size of 2 for both devices. The output from the second convolution filter was then flattened. The softmax

filter was next applied and calculated the probability of each activity and returned a tensor of dimensions 1 by 2.

The CNN was trained using stochastic gradient descent to minimize negative log-likelihood with batch size of 10 examples and a learning rate of 0.0001. The reduced mean was used to determine accuracy. The Go CNN was trained on an initial input width of 40 and required 5 training epochs to achieve 98.6158% testing accuracy. Initially, with this same network model and hyperparameters, the Gear testing accuracy was 85.16%. This accuracy was not considered high enough, hence the increase in the initial input width to 60 with training increased to 6 epochs to achieve 92.55% testing accuracy for the Gear. The input width could not be lengthened beyond 60 without causing noticeable lag when classifying activity in the WiP method.

3.3.4 VR Integration

Our CNN architecture and optimization were implemented in TensorFlow [1]. And the virtual reality integration was developed in Unity—a multi-platform game engine. For real time application in Unity, the trained CNN classifiers require input IMU data sequences to be in the same format (e.g., polling rate and input width) as used during training. Given this input, the trained CNN classifier predicts whether the user is currently ‘walking’ or ‘standing’ in real time. This prediction is then used to inform forward motion of the user by modifying the state machine described in Section 3.1 (Threshold Based Motion). In this modified version of the state machine, the transitions to and from the standing state are governed by the CNN. When the CNN detects that a user is ‘standing’, their velocity is immediately set to zero and the state machine transitions into the standing state. When the CNN predicts ‘walking’, the state machine transitions into the early stance phase. All other transitions are the same as described in Section 3.1

In comparison to the Go’s CNN WiP solution, the Gear’s CNN WiP solution more frequently misclassifies walking as standing. This is experienced by the user as latency for walking motion when walking in place. We suspect that this discrepancy in performance is due to the Gear’s comparatively noisy IMU data and its subsequent need for a longer input window to obtain comparable classification results. Longer temporal sequences risk capturing multiple activity states, which may compromise a network’s ability to distinguish between activities. To compensate for the delay, a smoothing latch is applied to the Gear’s CNN WiP classifier during runtime. As a result, if the Gear CNN reports the user as walking in the previous 0.25 seconds, the individual is reported as still walking. This introduces some lag, but has the benefit of preventing a number of start and stop errors.

4 EXPERIMENT 1

4.1 Hypotheses

We are interested in assessing the overall controllability of the WiP interfaces. And we believe that a system with high latency will affect controllability of the interface. Inspired by the experiment of Whitton et al. [44], in this experiment we have users traverse corridors with several turns in them, which we call mazes, although there is no possibility of getting lost. At various waypoints (gates) along the maze, users are instructed to stop on a target—as close to the target as possible. We measure the error between the target and their actual stopping location. We hypothesize generally that systems with high latencies and mismatches between stepping rate and optic flow rate will be less controllable in this experiment, and thus users will have worse performance. Specifically, we believe that the biomechanical and CNN models will outperform the threshold based model. We also believe that the IMU of the Samsung Gear VR system is less sensitive than the IMU of the Oculus Go. Therefore, the Gear will have inferior performance.

We measure simulator sickness, presence, and system usability in this experiment, and we hypothesize that less controllable systems will have lower presence and usability scores. Systems that have strong mismatches between optic flow rate and apparent velocity may be susceptible to simulator sickness. We do not believe any of our implementations suffer from this, but we measure it regardless.



Figure 4: In this study we used the Samsung Gear VR with an S8 (left) and the Oculus Go (right).

4.2 Materials

The virtual environment was displayed using either a Samsung Gear VR with a Galaxy S8 or an Oculus Go. The resolutions of the S8 and the Go are 1480x1440 and 1280x1440 per eye, respectively. The fields of view were not measured but both are reported to be near 90°–96°. Each HMD weighed 460g. Subject motion was tracked using the built-in IMUs of the Gear VR system or the Oculus Go. Subjects provided input either through the touchpad of the Gear VR or the motion controller provided with the Go.

4.3 Environment

Since Experiment 1 contains six conditions (2 headsets x 3 techniques), six paths were designed. Each path contained 180 straight segments connected via turns ranging between 90° and 135°. Each path segment consisted of two textured walls and two fence posts, which we call a gate. Within a given path, subjects could see the succeeding gate, but no farther. The layout of each of these mazes and a segment of a path can be seen in Figure 5. One continuous Voronoi textured floor plane spanned the entirety of the maze.

4.4 Participants

For the first experiment we recruited 21 college age students from our institution between the ages of 18 and 25. All participants gave written consent and were compensated \$15 for the experiment, which lasted roughly an hour and a half. Subjects were informed that they would participate in six walking trials for a duration of 5 minutes

each, but the walking methods were not specified. Three subjects were excluded from data analysis due to system malfunctions, and therefore 18 (9 male and 9 female) subjects remained.

4.5 Procedure

A within subjects experiment was conducted as a 2 (headset: Gear, Go) \times 3 (WiP technique: threshold, biomechanical, CNN) design. We blocked on headset to minimize the necessity of both switching HMDs and describing how each HMD worked. Headset and WiP technique were counterbalanced to prevent order effects.

Prior to entering VR, subjects were given brief instructions on how to walk in place. Calibration was required for only the biomechanical motion condition and consisted of three parts. The first phase of calibration requires subjects to walk in place for roughly 20 steps to determine an individualized threshold to detect individual steps. Next, subjects were asked to walk in place at a slow, medium, and then fast pace to determine the step rate detected by the system. Finally to correlate stepping rate to optic flow we ask subjects to actually walk at a slow, medium, and then fast pace. Upon completion of this calibration, subjects filled out a pre-test simulator sickness questionnaire and then entered the first condition. In each condition subjects were given five minutes to walk through the linear maze, stopping at each gate. The layouts of these mazes are shown in Figure 5, the ordering of the presented mazes was fixed. Subjects were instructed to stop at each gate for at least one second but could stop for longer, if necessary. The first three gates were considered training gates and were excluded from any analysis. After five minutes had passed, subjects were instructed to stop and remove the headset and complete three post-condition surveys.

4.6 Results

Experiment 1 includes several measurements of usability. We measure simulator sickness, controllability as measured by stopping distance, presence as measured by a standard Slater-Usoh-Steed (SUS) questionnaire [34, 35], and general usability which we assess using the System Usability Scale [39].

4.6.1 Control

To measure controllability we introduce stopping error. Stopping error is a measure of absolute distance from the center of the gate to the stopping location of the user for each gate. We define stopping location to be the location at which the longest stop in motion occurs. We require that the stop be within 2m of the gate. The average stopping error and standard error of the mean in each condition can be seen in Figure 6. We ran a 2x3 (headset x WiP technique) repeated measures ANOVA on stopping error. All assumptions were checked or corrected for by SPSS and we found an effect of WiP technique ($F(2, 34) = 9.412, p = .001$). Post-hoc t-test comparisons among conditions show that the threshold condition ($\delta = -.071, t = -4.58, p < .001$) and CNN condition ($\delta = -.055, t = -3.11, p = .006$) are significantly better than the biomechanical based condition.

4.6.2 SSQ

We administered a simulator sickness questionnaire before following each condition and a pre-test questionnaire to provide a baseline measurement. We found no effect of either condition or headset on simulator sickness.

4.6.3 Presence

To assess presence we administered post-test SUS questionnaires following each condition [34, 35]. As in Peck et al. [27], we transformed responses into a binary value and considered responses of 5, 6, or 7 to indicate high presence while other responses indicated low presence. The percentage of responses indicating high presence in each headset can be seen in figure 7. A logistic regression with two

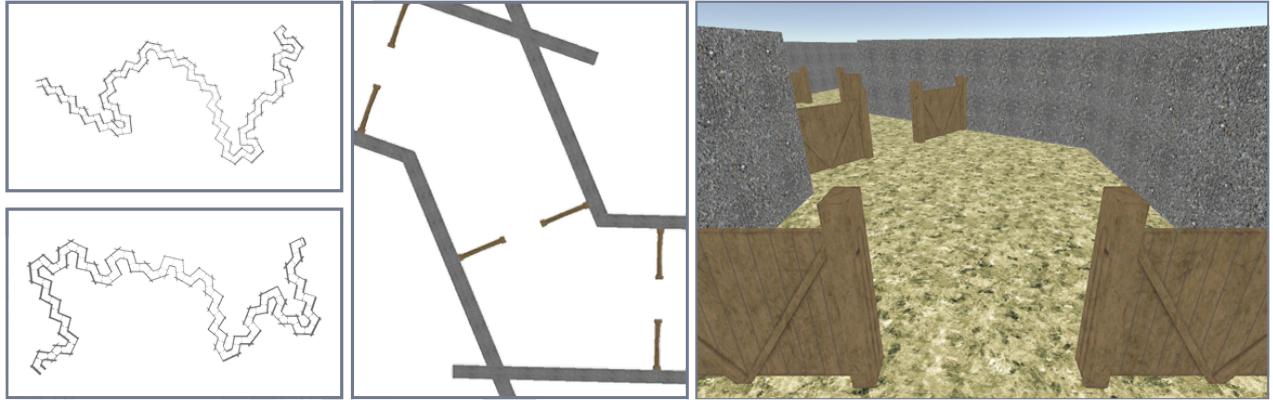


Figure 5: Top-down (right) view of the mazes used in Experiment 1. A zoomed in overhead view is shown in the middle and a first-person perspective view is shown left. Each maze was randomly generated such that there would be no overlap and so that the subsequent gate would be visible before making the turn.

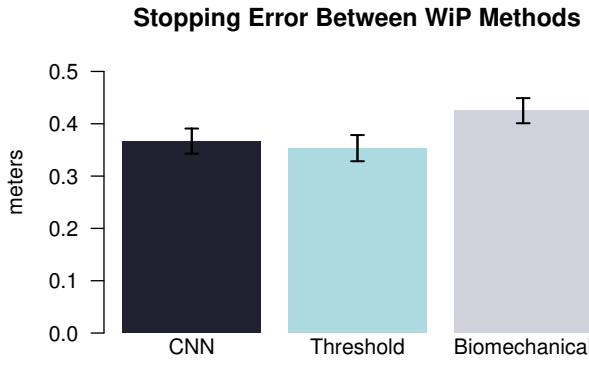


Figure 6: The stopping error measurements for the three WiP techniques. The threshold based and CNN based techniques result in subjects stopping significantly closer to the target than in the biomechanically based technique.

predictor values (headset and technique) revealed a significant difference between headsets in the presence evoked ($\chi^2(1) = 8.6617$, $p = 0.0034$), with the Oculus Go having significantly higher presence. No difference was found between the WiP techniques.

4.6.4 System Usability

Finally, to assess general usability of the conditions presented, we administered a System Usability Survey consisting of ten standard questions that measure ease of use. As with the SUS presence questionnaire, subjects completed this following each condition. We ran a 2×3 (headset x WiP technique) repeated measures ANOVA on the total score and found no significant differences between either factor.

4.7 Discussion

There was not a strong difference in controllability between the HMDs in this experiment, although the Oculus Go evoked significantly higher feelings of presence. The presence result is interesting, because the resolution of the Gear is actually higher than Go. The biomechanical WiP method did not perform as well as either the tuned threshold method or the CNN in controllability. Figure 6 shows that the biomechanical method of detecting steps and controlling velocity results in the worst controllability of the system. This

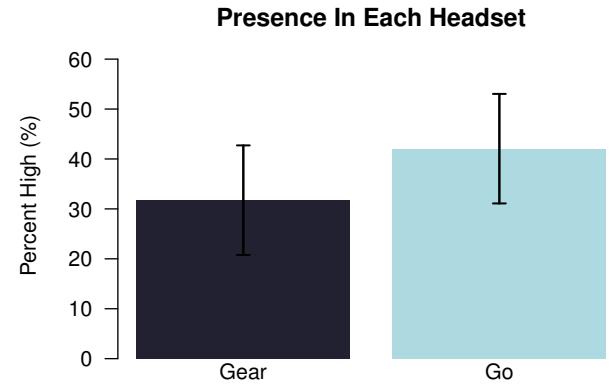


Figure 7: Presence evoked in each headset collapsed across WiP techniques.

method is the only one that allows speeds above 2.0 m/s, and this increased speed is likely the reason for the decreased controllability. Subjects are able to reach much higher speeds by stepping faster. The point is that there is a speed/controllability tradeoff in that moving at a high speed makes attempting to stop at the intended location more difficult. Users were either not willing or not able to regulate this tradeoff adequately with the biomechanical method, whereas the other methods enforced a speed limit. We will address this further in the general discussion, but it is possible that our mapping from stepping rate to optic flow should be better informed by HMD locomotion, as suggested by Durgin et al. [7].

5 EXPERIMENT 2

This experiment examines some of the issues raised by Experiment 1 in more detail by comparing real walking performance in a distance estimation task to WiP performance. That is, we will allow people a period of adjustment to the WiP technique to understand how optic flow correlates with distance, then we will judge how well they are able to use this information in a distance estimation task. Performance is compared to real walking as a metric.

5.1 Hypotheses

In this experiment, we perform egocentric distance estimation by blind walking (in place). During walking or WiP motion, no visual

feedback is provided to the subject. We thus test how robust the WiP technique is by evaluating how well participants are able to use the technique to perform a perception task. And we compare this with their performance on the same task with the same headset using real blind walking. If WiP locomotion is equivalent to real walking, as we would ideally like, then these should be same. Despite the results of Experiment 1, we hypothesize that the CNN and biomechanical techniques will yield answers closer to real walking than the threshold method. However, in view of Experiment 1, we hypothesize that the CNN method will outperform the biomechanical method.

5.2 Materials

The same HMDs and WiP techniques from Experiment 1 were used for Experiment 2.

5.3 Environment

Two environments were used during this experiment. The first is a linear maze as described in Experiment 1 and the second is sparse Voronoi textured ground plane.

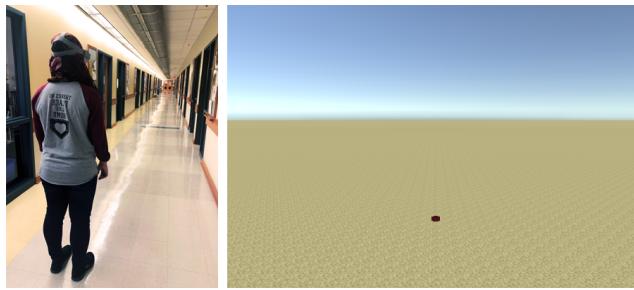


Figure 8: In Experiment 2 subjects stood at one end of a hallway (left) and after judging the distance, walked down the hallway based on that judged distance. Subjects saw a sparse Voronoi textured environment with no visual distal information beyond the puck itself (right). In this image the puck is 7.5m from the user.

5.4 Experimental Design

Experiment 2 is again a within subjects design with the same six conditions as in Experiment 1. Additionally, we add a real walking condition as a baseline, or ground truth, condition. As in Experiment 1, we blocked on headset, and the headset used for the real walking condition was randomized. The real walking condition was randomized and balanced within these conditions such that six subjects performed real walking first, six subjects performed real walking after completing all WiP techniques in one headset, and six subjects completed real walking as their final condition.

Each of the six WiP conditions consisted of two phases. In the first phase, we take a maze as described in Experiment 1 and allow subjects to walk in it for 60 seconds to become familiar with the method, the HMD, and the speed at which they can move. The second phase consists of nine distance estimation trials. In each trial subjects are stationary and are presented with a red hockey puck as a target (Figure 8). They are then asked to estimate the distance to that target. Once subjects indicate that they are familiar with the target, the virtual environment disappears, and subjects use the WiP technique or real walking to walk to the puck. The nine trials consist of three repetitions of three distance (5m, 7.5m, and 10m). Subjects pressed either the touchpad or the trigger on the motion controller to hide the puck and environment and enable tracking which allowed them to move. The system recorded distances during WiP trials and the experimenter used a laser measuring system to measure the distance walked. In the real walking condition, subjects walked

down a 70m hallway and no feedback on the distance traveled was given. Subjects were only alerted if they began to walk too far left or right with a tap on the shoulder. No subject reached the end of the hallway nor did they run into any walls.

5.5 Results

We are primarily interested in comparing the accuracy of how users performed in the WiP conditions with the accuracy of users in the real walking condition, rather than absolute accuracy. The measure we use is the ratio of judged distance to true distance, where by “judged” we mean both perceived and acted upon through the WiP system. Thus,

$$\text{Judged Ratio of True Distance} = \frac{\text{Judged Distance}}{\text{True Distance}}.$$

We average this value across conditions for each subject and trial to find the mean judged ratio in Table 1. Note that both the Gear and Go show signs of distance underestimation of the virtual environment. Subjects underwalked by an average of 26%. This amount of distance compression is different than has been reported for the Oculus Rift in Buck et al. [4]. To compare this with WiP methods, we ran a repeated measures ANOVA with headset and WiP technique as factors. We found a main effect of both headset ($F(1,971) = 9.933, p = .002$) and technique ($F(2,971) = 20.023, p < .001$). There is also a significant interaction between headset and technique ($F(2,971) = 7.964, p < .001$). The interaction reveals that the biomechanical technique in the Go HMD is higher than in the Gear HMD (see Figure 9). Post-hoc Tukey tests revealed a significant difference between each of the three WiP techniques with the CNN based motion having the lowest amount of over walking.

5.6 Discussion

The CNN technique performed significantly better than the other two methods. Although not the main focus of this paper, significant distance underestimation occurs in both HMD platforms. The biomechanical method had significantly different performance in the Go and Gear HMDs, which was surprising since we did not find that in Experiment 1. One possibility is that the Go is more sensitive to noise than the Gear, which makes it harder to control. We do not particularly observe this in data we collect, nor did we find this difference in Experiment 1. In examining the performance of the individual subjects in this experiment, we note that two subjects considerably overestimated the distance in the biomechanical WiP method compared to all other subjects and compared to themselves (judged ratios of 4 compared to ratios of 1.4 for the next highest in the biomechanical WiP condition; 1.4 was also the highest ratio in the Gear). We have no explanation for these ratios. They were not, for example, the first trials that the subjects did in WiP distance estimation, nor were they the first trials with the Go. Eliminating these subjects and re-performing the analysis above does not affect the main effect of WiP technique — the CNN method still outperforms the other methods — but it does eliminate the interaction and the main effect of headset. In contrast to experiment 1, there was no visual feedback with which the subject could adjust their walking mechanics to stop near the goal. We thus attribute the improved performance in the CNN based condition to the system having better recognition of general walking than that of the threshold based condition.

6 GENERAL DISCUSSION

In this paper we compare three WiP methods on two different mobile VR platforms — the Gear VR and Oculus Go — and we assessed both the WiP methods and the mobile VR platforms. We are interested in mobile VR platforms because they do not have external positional tracking, and thus our implementations use only the IMUs

| Method | Total Mean | Gear Mean | Go Mean |
|---------------|------------|-----------|---------|
| CNN | .936 | .873 | .999 |
| Biomechanical | 1.246 | 1.025 | 1.448 |
| Threshold | 1.085 | 1.112 | 1.058 |
| Real | .739 | .704 | .774 |

Table 1: Ratio of over walking collapsed across headsets for each condition.

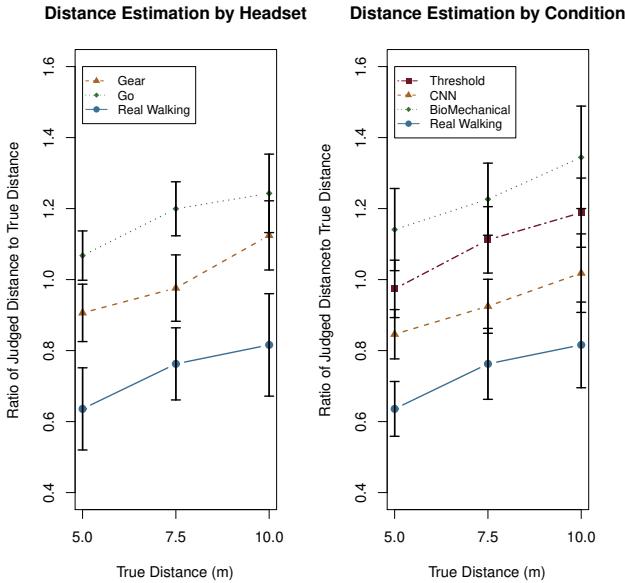


Figure 9: Ratio of the judged distance calculated for each headset (left) and technique (right) compared to real walking in Experiment 2.

of the systems to detect WiP motions. The basic WiP method we started from was a hand-tuned implementation. We modified it first to incorporate a data-driven biomechanical approach based on a model that relates step rate to velocity. We also modified the basic method to incorporate a type of deep network, a convolutional neural network or CNN, to apply an even more data-driven approach in determining how WiP is executed. We assessed these methods and platforms for controllability and performance in two experiments and found, generally, that the CNN performed best in both.

We had originally hypothesized that there would be a difference in the two mobile VR platforms. This hypothesis was based on our experience designing the CNN, where the CNN for the Gear VR was more troublesome to build and never performed as well in initial tests as that for the Go. We did find a difference in presence ratings between the two platforms, with the Go having a higher presence rating than the Gear VR, but we are reluctant to draw the conclusion that trouble with the IMU in the Go (from building the CNN) resulted in higher presence in the Go and no other differences. Rather, we believe the higher presence reflects ergonomic issues with the comfort and fit of the Go, although a more careful investigation of this would be needed. In particular, we note that there is no difference in the weight of the HMDs, the resolution of the Gear is actually superior to that of the Go, and there was no detectable difference between platforms in terms of usability or simulator sickness. However, there was also a difference between the two platforms in Experiment 2 with one of the WiP implementations, where a subset of subjects (two out of eighteen) had significantly worse performance in the Go. We currently have no explanation for this and it is a topic for future investigation.

We also hypothesized that the CNN would perform well, and this

hypothesis was confirmed in both experiments. We offer a conjecture (based on our observations during both experiments) as to why the CNN performs better. WiP gestures in general are usually unnatural, as has been noted elsewhere [23]. Thus, a marching gesture quickly becomes tiresome, and people resort to a more relaxed type of motion. In Experiment 1, the duration that people had to walk was considerable for a marching motion (five minutes). In this experiment, if they faltered in their motion, however, they received immediate visual feedback that they were not moving and had to pick up their step. In Experiment 2, subjects received no visual feedback if the WiP motion was working or not. In either case, we believe that the CNN, because it was data-driven and based on a large training set of people walking in place for a considerable period time, was more robust to variations in the WiP walking gesture. It could thus better interpret when people intended to walk in place and when they intended to stop.

We hypothesized that our biomechanically-based method would perform well, and this hypothesis was not confirmed. This WiP method did not perform as well as we intended. The biomechanical method was not as controllable as the other two methods, as demonstrated by Experiment 1. Perhaps it is not surprising, therefore, that it underperformed in the second experiment. The reasons why it is not as controllable are a topic of ongoing inquiry. It is possible that the gain of the system, even though it is based on an individual's walking speed, is too high. Some evidence for this can be found in the work of Durgin et al. [7].

We note that our experiment found that subjects underestimated distances in both platforms as judged by egocentric blind walking. Both platforms are lightweight and have large field of views, but the amount of distance underestimation is not consistent with other commodity level devices that have recently been tested [4, 6, 11, 16]. Further investigation of this issue seems warranted.

There are many avenues for further research in this area. Expanding the role of the CNN to cover more functionality of the WiP algorithm, such as velocity control, seems a clear path forward. Other machine learning methods, such as recurrent neural networks [10] may offer advantages over CNNs for walking in place, since they might be able to capture walking sequences better. The addition of extra functionality such as velocity to the neural network increases the training complexity of it and different frameworks may give different results. This is also a question for further work.

We did not directly assess starting latencies in this work, and determining how they affect the quality of the WiP interface is itself an open problem. And there are clear directions in addressing how biomechanics and individualization can be better accomplished.

7 CONCLUSION

This research has shown that WiP interfaces can be successfully implemented in a data-driven manner on mobile VR platforms, and that people can use them successfully. In particular, we have shown that a deep network approach to WiP systems can lead to a successful locomotion interface on a commodity-level system without external position tracking.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grants 1116988, 1116636, and 1526448; by the Office of Naval Research under grant N00014-18-1-2964; and by the Department of Defense Contract Number W81XWH-17-C-0252 from the CDMRP Defense Medical Research and Development Program.

REFERENCES

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow,

- A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] C. Boletsis. The new era of virtual reality locomotion: A systematic literature review of techniques and a proposed typology. *Multimodal Technologies and Interaction*, 1(4), 2017. doi: 10.3390/mti1040024
- [3] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. Locomotion in virtual reality for room scale tracked areas. *International Journal of Human-Computer Studies*, 122:38–49, 2019. doi: 10.1016/j.ijhcs.2018.08.002
- [4] L. E. Buck, M. K. Young, and B. Bodenheimer. A comparison of distance estimation in hmd-based virtual environments with different hmd-based conditions. *ACM Trans. Appl. Percept.*, 15(3):21:1–21:15, July 2018. doi: 10.1145/3196885
- [5] S. Creem-Regehr, J. Stefanucci, and W. Thompson. Perceiving absolute scale in virtual environments: How theory and application have mutually informed the role of body-based perception. *Psychology of Learning and Motivation - Advances in Research and Theory*, 62:195–224, 2015. doi: 10.1016/bs.plm.2014.09.006
- [6] S. H. Creem-Regehr, J. K. Stefanucci, W. B. Thompson, N. Nash, and M. McCardell. Egocentric distance perception in the oculus rift (dk2). In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*, SAP '15, pp. 47–50. ACM, New York, NY, USA, 2015. doi: 10.1145/2804408.2804422
- [7] F. H. Durgin, C. Reed, and C. Tigue. Step frequency and perceived self-motion. *ACM Trans. Appl. Percept.*, 4(1), Jan. 2007. doi: 10.1145/1227134.1227139
- [8] J. Feasel, M. C. Whitton, and J. D. Wendt. Llcm-wip: Low-latency, continuous-motion walking-in-place. In *Proceedings of the 2008 IEEE Symposium on 3D User Interfaces*, 3DUI '08, pp. 97–104. IEEE Computer Society, Washington, DC, USA, 2008. doi: 10.1109/3DUI.2008.4476598
- [9] E. Guy, P. Punpongsanon, D. Iwai, K. Sato, and T. Boubekeur. Lazynav: 3d ground navigation with non-critical body parts. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 43–50, March 2015. doi: 10.1109/3DUI.2015.7131725
- [10] N. Y. Hammerla, S. Halloran, and T. Ploetz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *CoRR*, abs/1604.08880, 2016.
- [11] J. W. Kelly, L. A. Cherep, and Z. D. Siegel. Perceived space in the htc vive. *ACM Trans. Appl. Percept.*, 15(1):2:1–2:16, July 2017. doi: 10.1145/3106155
- [12] J.-S. Kim, D. Gracanin, and F. Quek. Sensor-fusion walking-in-place interaction technique using mobile devices. In *Proceedings of the 2012 IEEE Virtual Reality*, VR '12, pp. 39–42. IEEE Computer Society, Washington, DC, USA, 2012. doi: 10.1109/VR.2012.6180876
- [13] A. Kitson, A. M. Hashemian, E. R. Stepanova, E. Kruijff, and B. E. Riecke. Comparing leaning-based motion cueing interfaces for virtual reality locomotion. In *3D User Interfaces (3DUI), 2017 IEEE Symposium on*, pp. 73–82. IEEE, 2017.
- [14] E. Langbehn, T. Eichler, S. Ghose, K. von Luck, G. Bruder, and F. Steinicke. Evaluation of an omnidirectional walking-in-place user interface with virtual locomotion speed scaled by forward leaning angle. In *Proceedings of the GI Workshop on Virtual and Augmented Reality (GI VR/AR)*, pp. 149–160, 2015.
- [15] S.-M. Lee, S. M. Yoon, and H. Cho. Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 131–134, Feb 2017. doi: 10.1109/BIGCOMP.2017.7881728
- [16] B. Li, R. Zhang, A. Nordman, and S. Kuhl. The effects of minification and display field of view on distance judgments in real and hmd-based environments. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '15. ACM, New York, NY, USA, 2015.
- [17] J. M. Loomis. Distal attribution and presence. *Presence: Teleoperators and Virtual Environments*, 1:113–119, 1992.
- [18] M. McCullough, H. Xu, J. Michelson, M. Jackoski, W. Pease, W. Cobb, W. Kalesky, J. Ladd, and B. Williams. Myo arm: swinging to explore a ve. In *Proceedings of the ACM SIGGRAPH Symposium on applied perception*, SAP '15, pp. 107–113. ACM, 2015-09-13.
- [19] N. C. Nilsson, S. Serafin, and R. Nordahl. The perceived naturalness of virtual locomotion methods devoid of explicit leg movements. In *Proceedings of Motion on Games*, MIG '13, pp. 133:155–133:164. ACM, New York, NY, USA, 2013. doi: 10.1145/2522628.2522655
- [20] N. C. Nilsson, S. Serafin, and R. Nordahl. Establishing the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):569–578, April 2014. doi: 10.1109/TVCG.2014.21
- [21] N. C. Nilsson, S. Serafin, and R. Nordahl. The influence of step frequency on the range of perceptually natural visual walking speeds during walking-in-place and treadmill locomotion. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, pp. 187–190. ACM, New York, NY, USA, 2014. doi: 10.1145/2671015.2671113
- [22] N. C. Nilsson, S. Serafin, and R. Nordahl. The effect of visual display properties and gain presentation mode on the perceived naturalness of virtual walking speeds. In *2015 IEEE Virtual Reality (VR)*, pp. 81–88, March 2015. doi: 10.1109/VR.2015.7223328
- [23] N. C. Nilsson, S. Serafin, and R. Nordahl. Walking in place through virtual worlds. In M. Kurosu, ed., *Human-Computer Interaction. Interaction Platforms and Techniques*, pp. 37–48. Springer International Publishing, Cham, 2016.
- [24] N. C. Nilsson, S. Serafin, F. Steinicke, and R. Nordahl. Natural walking in virtual reality: A review. *Comput. Entertain.*, 16(2):8:1–8:22, Apr. 2018. doi: 10.1145/3180658
- [25] R. Paris, M. V. Joshi, Q. He, G. Narasimham, T. P. McNamara, and B. Bodenheimer. Acquisition of survey knowledge using walking in place and resetting methods in immersive virtual environments. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '17. ACM, New York, NY, USA, 2017.
- [26] T. Peck, H. Fuchs, and M. Whitton. The design and evaluation of a large-scale real-walking locomotion interface. *Visualization and Computer Graphics, IEEE Transactions on*, 18(7):1053–1067, 2012. doi: 10.1109/TVCG.2011.289
- [27] T. C. Peck, H. Fuchs, and M. C. Whitton. Evaluation of reorientation techniques and distractors for walking in large virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 15:383–394, 2009. doi: 10.1109/TVCG.2008.191
- [28] P. Peng, B. E. Riecke, B. Williams, T. P. McNamara, and B. Bodenheimer. Navigation modes in virtual environments: walking vs. joystick. Poster at the Symposium on Applied Perception in Graphics and Visualization, Aug. 2008.
- [29] S. Razzaque, D. Swapp, M. Slater, M. C. Whitton, and A. Steed. Redirected walking in place. In *Proceedings of the Workshop on Virtual Environments 2002*, EGVE '02, pp. 123–130. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002.
- [30] R. S. Renner, B. M. Velichkovsky, and J. R. Helmert. The perception of egocentric distances in virtual environments — A review. *ACM Computer Surveys*, 46(2):23:1–23:40, 2013.
- [31] B. E. Riecke, B. Bodenheimer, T. P. McNamara, B. Williams, P. Peng, and D. Feuerissen. Do we need to walk for effective virtual reality navigation? physical rotations alone may suffice. In *Spatial Cognition*, pp. 234–247, 2010.
- [32] J. J. Rieser, D. H. Ashmead, C. R. Taylor, and G. A. Youngquist. Visual perception and the guidance of locomotion without vision to previously seen targets. *Perception*, 19:675–689, 1990.
- [33] A. Saeed, T. Ozcelebi, S. Trajanovski, and J. Lukkien. Learning behavioral context recognition with multi-stream temporal convolutional networks, 2018.
- [34] M. Slater and M. Usoh. Presence in immersive virtual environments. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pp. 90–96, Sep 1993. doi: 10.1109/VRAIS.1993.380793
- [35] M. Slater, M. Usoh, and A. Steed. Depth of presence in virtual environments. *Presence: Teleoperators and Virtual Environments*, 3(2):15, 1994.
- [36] M. Slater, M. Usoh, and A. Steed. Taking steps: The influence of a

- walking technique on presence in virtual reality. *ACM Transactions on Human Computer Interaction*, 2(3):201–219, 1995.
- [37] E. A. Suma, G. Bruder, F. Steinicke, D. M. Krum, and M. Bolas. A taxonomy for deploying redirection techniques in immersive virtual environments. In *2012 IEEE Virtual Reality Workshops (VRW)*, pp. 43–46. IEEE, 2012.
- [38] S. Tregillus and E. Folmer. 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, CHI ’16*, pp. 1250–1255. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858084
- [39] Usability.Gov. System usability scale (sus). internet, 2018. Accessed: 2018-6-21.
- [40] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks. Walking > walking-in-place > flying, in virtual environments. In *SIGGRAPH 99*, pp. 359–364, Aug. 1999.
- [41] J. D. Wendt. *Real Walking Models Improve Walking-In-Place Systems*. PhD thesis, University of North Carolina at Chapel Hill, 2010.
- [42] J. D. Wendt, M. C. Whitton, D. Adalsteinsson, and F. P. Brooks. Reliable forward walking parameters from head-track data alone. In *2012 IEEE Virtual Reality Workshops (VRW)*, pp. 81–82, March 2012. doi: 10.1109/VR.2012.6180892
- [43] J. D. Wendt, M. C. Whitton, and F. P. Brooks. Gud wip: Gait-understanding-driven walking-in-place. In *2010 IEEE Virtual Reality Conference (VR)*, pp. 51–58, March 2010. doi: 10.1109/VR.2010.5444812
- [44] M. C. Whitton, J. V. Cohn, J. Feasel, P. Zimmons, S. Razzaque, S. J. Poultney, B. McLeod, and J. Frederick P. Brooks. Comparing ve locomotion interfaces. *Virtual Reality Conference, IEEE*, 0:123–130, 2005. doi: 10.1109/VR.2005.25
- [45] M. C. Whitton and T. C. Peck. *Stepping-Driven Locomotion Interfaces*, pp. 241–262. Springer New York, New York, NY, 2013. doi: 10.1007/978-1-4419-8432-6_11
- [46] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. J. Rieser, and B. Bodenheimer. Exploring large virtual environments with an hmd when physical space is limited. In *Symposium on Applied Perception in Graphics and Visualization*, pp. 41–48. Tübingen, Germany, July 2007.
- [47] P. T. Wilson, W. Kalescky, A. MacLaughlin, and B. Williams. Vr locomotion: walking, walking in place, arm swinging. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*, pp. 243–249. ACM, 2016.
- [48] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI’15*, pp. 3995–4001. AAAI Press, 2015.
- [49] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pp. 197–205, Nov 2014. doi: 10.4108/icst.mobicas.2014.257786
- [50] D. Zielasko, S. Horn, S. Freitag, B. Weyers, and T. W. Kuhlen. Evaluation of hands-free hmd-based navigation techniques for immersive data analysis. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 113–119, March 2016. doi: 10.1109/3DUI.2016.7460040