VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments

Conference Paper · May 2016

DOI: 10.1145/2858036.2858084

VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments

Sam Tregillus
tregillus@cse.unr.edu

Eelke Folmer
efolmer@cse.unr.edu

Human+ Lab - Computer Science - University of Nevada

ABSTRACT
Low-cost smartphone adapters can bring virtual reality to the masses, but input is typically limited to using head tracking, which makes it difficult to perform complex tasks like navigation. Walking-in-place (WIP) offers a natural and immersive form of virtual locomotion that can reduce simulation sickness. WIP, however, is difficult to implement in mobile contexts as it typically relies on bulky controllers or an external camera. We present VR-STEP; a WIP implementation that uses real-time pedometry to implement virtual locomotion. VR-STEP requires no additional instrumentation outside of a smartphone's inertial sensors. A user study with 18 users compares VR-STEP with a commonly used auto-walk navigation method and finds no significant difference in performance or reliability, though VR-STEP was found to be more immersive and intuitive.

Author Keywords
Virtual reality; pedometry; walking-in-place; virtual locomotion; head-mounted display; motion-sickness; games;

ACM Classification Keywords
I.3.7 Graphics: 3D Graphics and Realism–Virtual Reality;

INTRODUCTION
Virtual reality (VR) has reached potential for mainstream adoption due to the introduction of low-cost smartphone adapters, such as Google Cardboard, that can transform any smartphone into a head-mounted VR display. Navigation is an essential 3D interaction task [7], but a challenge with head-mounted displays is that their input options are limited [10]. Because the smartphone is inside the adapter, touchscreen input is not possible. A controller cannot be used with Google Cardboard adapters because its users are required to hold the adapter with both hands. Doing so limits head rotation speed to the torso, which may minimize simulation sickness [1]. Due to these constraints, input typically relies on head-tracking using the smartphone's inertial sensors. Several VR apps feature an auto-walk button that is rendered at the user's feet. Users toggle the auto-walk by looking down at the button briefly, which then starts moving the user in the direction of their gaze. Besides being limited to a fixed locomotion speed, this method requires users to interrupt their focus on the virtual environment to look down at the button, which may be detrimental to immersion and undesirable in gaming contexts.

CHI'16, May 07-12, 2016, San Jose, CA, USA.
©2016 ACM. ISBN 978-1-4503-3362-7/16/05 $15.00
DOI: http://dx.doi.org/10.1145/2858036.2858084

Walking-in-place (WIP) allows users to perform handsfree virtual locomotion using step-like movements while remaining stationary [26]. WIP resembles walking and offers a natural, immersive form of input [25]. When compared to using a controller, WIP allows for better spatial orientation [15] while approximating the performance of real walking [23].

Simulation sickness is a major concern to the mass adoption of VR [19] and is caused by –among other factors– a sensory mismatch between visual and vestibular stimuli [29]. Because WIP generates proprioceptive feedback, users are less likely to experience simulation sickness than when using a conventional controller [13]. Unfortunately, current WIP implementations aren't feasible in mobile contexts, as they rely either on external cameras [24], which may require calibration, or expensive, bulky treadmill controllers [33].

This paper addresses a need for a low-cost virtual locomotion technique [33] by presenting VR-STEP: a novel WIP implementation that requires no instrumentation beyond the sensors already present in smartphones. VR-STEP is immersive, has low latency, and is easy to learn. We show a quantitative and qualitative comparison of our system with another handsfree navigation technique that is found in several VR apps. We provide a detailed description of our virtual locomotion implementation. Our results highlight the importance of smooth, natural movement and low starting/stopping latency in VR locomotion.

BACKGROUND
Exploring natural, immersive types of input for virtual locomotion is an active field of research (see [26] for an survey). Using a joystick for navigation is a simple and common approach. Gaze based navigation with joystick control of locomotion speed was found to be superior to navigation using a joystick alone [23]. Real walking, where the user actually moves in real space, is the most natural and effective method for virtual locomotion [23, 30]. Real walking can be implemented using an optical tracking system [31, 30], but this approach doesn't scale very well since the tracked space and the virtual space need to be the same size. Omnidirectional treadmills [9] circumvent physical space limitations, but these are often expensive and bulky, which restricts their use in mobile contexts.

Walking-in-place [24] has many advantages over other methods for virtual locomotion. WIP can be used to control locomotion speed just as effectively as using a joystick [11], but WIP has also been found to be much more immersive [25]. Maintaining spatial knowledge of the virtual world is difficult in VR environments [21], but WIP was found to allow for better spatial orientation than when using a joystick [15]. WIP has also been found to be almost as efficient as real walking [23] but is not subject to space constraints and is therefore more cost effective to implement.

A challenge with implementing WIP is starting and stopping latency [26], i.e., how quickly a step is detected and translated into virtual motion, or how long it takes the virtual avatar to stop after the user stops taking any steps. High stopping and starting latency makes precise navigation challenging [11]. Latency can be a result of the step detection algorithm implementation, noise reduction, or damping based on which part of the body is being tracked. In general, step detection is most accurate when tracking motions closest to the feet [12]. The first WIP [24] implementation used an optical motion tracking system to track head movements to infer steps. However, due to "false steps" being detected, they did not start actual motion until 4 consecutive steps had been taken, which resulted in a large delay between walking and actually moving. Other implementations used optical tracking of knee or shin motions [26, 32], and had a latency of a "half-step", or about 400ms at a normal walking pace. Another method by Feasel [11] used a magnetic tracking system for detecting heel motions, with a starting latency of 138ms and stopping latency of 96ms.

The optical & magnetic tracking systems used in the above studies are typically expensive [11, 33], and their weight and size makes them difficult to employ in mobile contexts. Low-cost implementations have also been explored. To enable desktop VR, a basic webcam has been used to detect head oscillations, which are then interpreted for virtual locomotion [27]. A user study found this approach to be faster and more immersive than keyboard input when seated. A commercially-available pressure-sensitive game controller (Wii Balance Board) was explored for step detection [33]. A study found no significant difference in latency when compared with joystick and real walking input, but limitations included difficulty controlling small motions and a risk of users falling off the balance board. Other notable WIP implementations include using a Kinect [34] and a CAVE system [14, 22] but these implementations aren't feasible in mobile contexts.

IMPLEMENTATION OF VR-STEP

Inertial measurement units (IMU) have become ubiquitous in smartphones and typically consist of a 3-axis accelerometer and 3-axis gyroscope. Various studies have investigated the accuracy of pedometry for different smartphone/sensor placements on the body [8, 16] and found that steps can be detected most accurately for a sensor worn closest to the feet[12]. Rather than using an external sensor, VR-STEP uses the smartphone's accelerometer for step detection. Since the smartphone sits in the head-mounted adapter, latency is a major concern for effective virtual locomotion [11]. A challenge with our approach is that a smartphone worn on the head may dampen the acceleration signal (due to the larger distance from the feet), causing problems for accurate step detection. However, recent work [5] evaluated the accuracy with which pedometry can be achieved on a head mounted display (Google Glass) and found no significant difference in accuracy on the head when compared with a smartphone worn on in the pocket or held in the hand, which supports the feasibility of our approach.

Though various step detection algorithms exist [8], we use a real-time algorithm proposed by Zhao [35]. This algorithm has little computational overhead and ensures the smartphone can maintain a high frame rate, which is important for VR simulations [17]. This algorithm averages every 5 samples to smooth the acceleration signal, which minimizes noise while still yielding a near real-time response of between 100-200ms. A step is detected if the accelerometer values pass a dynamic threshold level with a large enough negative slope. The dynamic threshold changes every 50 samples from the accelerometer to account for different step intensities. The slope acts as a sensitivity setting, which we can tune to make sure that small steps that the user uses to turn left and right are not translated into forward movement.

Once steps are detected, they need to be translated into virtual locomotion, in the direction of the user's gaze. A limitation of existing WIP implementations is that most provide few details on how virtual locomotion is implemented [11]. For our implementation, we wanted users to feel like they have precise control over their velocity, to be able to move reasonable distances quickly, but also be able to make small changes in position if needed. Because users walk in place, there is no way to detect the length of their stride to determine how far users might want to move. A discrete approach where every detected step is translated into moving a fixed distance forward leads to jarring and unnatural movement [11]. A high stopping latency can make users experience a frustrating sensation of gliding [11].

We query the accelerometer and gyroscope values to determine acceleration along the gravity axis, which we feed into the Zhao's algorithm described above. A sensitivity value, S, can be used to tweak how sensitive the algorithm is. When a step is detected, an event is fired, which the step handler listens for. The step handler measures the time between steps, (t_{step}) to determine the virtual velocity (v). A large t_{step} indicates that the user is walking slowly, so we set v to a low value. A small t_{step} value indicates faster steps, so we set v to a higher value.

We establish a bounds on $t_{step} \in [I_{min}, I_{max}]$, and a maximum and minimum velocity, $[V_{min}, V_{max}]$. If $t_{step} > I_{max}$, it means that a step has not been detected in some time (the user has been standing still before they took this step), so we set v to V_{min}. $t_{step} < I_{min}$ is not possible, as it is filtered out by the step detector, but if $t_{step} < I_{min}$, velocity is set to V_{max}. If $t_{step} \in [I_{min}, I_{max}]$, we linearly interpolate v between $[V_{min}, V_{max}]$ according to:

$$v = \frac{t_{step} - I_{min}}{I_{max} - I_{min}} * (V_{max} - V_{min}) + V_{min} \tag{1}$$

By modifying their WIP speed, users have precise control over their virtual speed. We also adjust the velocity of the avatar between steps by applying friction. This slows down the user between steps, giving a natural feeling of walking, and it reduces stopping latency, since the user begins slowing down immediately after taking their last step. If the user continues walking, the velocity is maintained.

EVALUATION

A number of studies have already compared WIP with joystick based virtual locomotion [23, 11, 25, 15, 33]. We also felt that comparisons to other WIP methods that involve extensive instrumentation weren't as useful, as users of mobile VR would not typically have access to such instrumentation. Because VR-STEP is hands free and requires no instrumentation, it is more meaningful to compare its performance with another hands free navigation method, "look down to move" (LDTM), which is used in several VR apps, such as the popular Oculus "Tuscany" demo [4]. Users toggle a button at their feet by briefly looking down at it, then back up. When activated the user will move with a fixed horizontal velocity in the direction of their gaze. Similar to other WIP evaluations [23, 11, 25, 15, 33, 6], we compare VR-STEP to LDTM by having users perform a number of navigation tasks.

Instrumentation

For our smartphone mount, we used a Zeiss VR One. The VR One features Zeiss precision lenses with a 100 degree field of view. The VR One features a strap and weighs 590 grams. We used an Android Nexus 5 smartphone with a Qualcomm Snapdragon 800 CPU (2.3 GHz quad-core) and an Adreno 330 GPU which can render 3D simulations with a high frame rate. The Nexus 5 features a InvenSense MPU-6515 six-axis (gyro + accelerometer) with a 50Hz sample rate. The user study was implemented using the Google Cardboard for Unity SDK and the Unity 5 engine.

Because LDTM uses a fixed velocity (V_{LDTM}), a precise quantitative comparison with VR-STEP is challenging, as VR-STEP allows variable locomotion speeds. Even if we limit V_{max} to be the same as V_{LDTM}, the average speed for VR-STEP will be smaller than LDTM unless the participant consistently runs at the highest speed. To allow for a fair comparison, we modified VR-STEP such that when $t_{step} < I_{max}$, we set v to V_{LDTM}. If no step has been taken for I_{max} seconds, we immediately set v to 0. This means that velocity is the same for both methods at all times when moving, but it does make stopping latency for VR-STEP equal to I_{max}, which causes a minor gliding issue, where after not taking any steps, you continue to move forward for I_{max} seconds. For the user study, we set I_{max} = .7s, and V_{LDTM} to 6 m/s.

Participants

We recruited 18 participants (9 female, average age 28.83, SD=7.93) to participate in our user study. None of the subjects self-reported any non-correctable impairments in perception or limitations in mobility. Individuals who had previously experienced simulator sickness were excluded from participation as they were at a high risk of not completing the study. User studies were approved by an IRB. On average users had 4.97 years experience (SD=8.05) navigating 3D environments and 11 subjects had prior experience using a VR headset, such as the Oculus Rift.

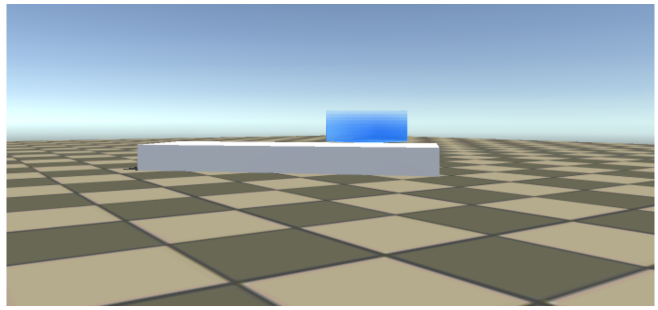

Figure 1. A view of the navigation task in the virtual environment. The blue area is the target zone, and the white rectangle is an obstacle.

Procedure

A criticism of existing WIP studies is that they have mostly included navigation tasks that use a straight trajectory [28], which isn't very realistic for many VR applications. Our study had participants perform two different navigation tasks: an unobstructed navigation task with a straight trajectory, and an obstructed one requiring them to circumnavigate an obstacle. We designed our navigation task to be similar to one described by Bowman [6]. Our virtual environment consists of a flat plane featuring a checkerboard pattern. Participants must find a target zone (a blue transparent column), navigate to it, and remain inside the area for 2 seconds before the next target zone becomes visible. This column changes color when participants enter it, and auditory feedback indicates success. The locations of the target zone were initially determined at random, with each new location a minimum distance of 10 meters away from the prior location. Obstructed navigation tasks feature a rectangular box (Size: 1x3 meters) halfway to the target zone. To decouple the visual search task from the navigation task, we started measuring time and distance traveled as soon as participants started moving. Each participant tests both navigation methods, but which method they use first is randomly assigned (half of the participants used VR-STEP first, then LDTM). Participants performed 20 navigation tasks for each navigation method with 10 tasks containing an obstacle. The specific locations of the target zones and the obstacle ordering was the same for all users and both methods. Before the data collection, participants were fitted with the VR One headset and we ensured the HMD was firmly attached to their head using the straps. Some participants with eyeglasses took them off but others kept them on. A built-in tutorial explained how each navigation method worked and let participants practice three navigation tasks. After the trials, basic demographic and qualitative feedback was collected using a questionnaire. The entire session, including training and questionnaires, took about 10 minutes.

Results

Table 1 lists the results for all users. For our analysis, we used a two-way repeated measures ANOVA where we evaluate efficiency of each navigation method using time and distance

Method-trajectory	Time (s)	SD	Distance (m)	SD
LDTM-straight	59.12	25.0	225.31	127.1
VR-STEP-straight	61.88	8.7	198.18	13.3
LDTM-curved	65.31	19.2	243.25	65.4
VR-STEP-curved	77.45	23.3	224.87	32.3

Table 1. Total time and distance traveled averaged per user.

traveled. There was no statistically significant two-way interaction between navigation type and trajectory for navigation time ($\boldsymbol{F}_{1,17} = 1.680$, $p = .212$), or distance ($\boldsymbol{F}_{1,17} = .107$, $p = .784$). Pairwise comparisons using a Bonferroni correction found a statistical significant difference between navigation methods for time for a curved trajectory ($p = .012$) but not for distance or straight trajectories.

Qualitative results. After the trial, we let participants rank each navigation method based on a number of criteria which included: efficiency, reliability, learnability, immersion and overall preference. If participants had no preference, they could also select both as a 3^{rd} option. The summarized results are shown in Figure 2. A Chi square test found the rankings for learnability ($p = .0023$) and immersion ($p = .0045$) to be statistically significantly different. Non-directed interviews with open-ended questions were used to collect general experiences and identify areas for improvement. For LDTM, ten participants stated that having to look down to activate the button was awkward or strenuous for their neck. Six participants suggested using a larger button or placing the button at eye level. For VR-STEP, eight participants said they found it difficult to stop precisely in the target zone due to gliding. Two participants wanted control over their velocity as they felt their avatar was moving too fast. Three participants wanted better step detection.

DISCUSSION

Overall, we demonstrated the feasibility of VR-STEP as a low-cost virtual locomotion technique for mobile VR. Participants found VR-STEP more immersive and easy to learn than LDTM. Though LDTM was perceived to be more reliable and efficient, quantitative results only showed LDTM to be faster for curved trajectories. An analysis of paths revealed that some participants overshot the target zone, requiring them to turn around. Overshooting would occur with LDTM as well, but not as frequently as with VR-STEP. If users failed to stop, the switchback they took back to the target zone significantly increased both time and distance, leading to much larger standard deviation values for these metrics.

Ironically, the main user complaints about VR-STEP, high stopping latency (gliding) and no control over speed, are issues that are absent when variable locomotion speed is used. Though we wanted to be able to compare quantitative values, such as navigation speed, between the two methods, the changes we made to equate velocities ended up hurting user perception of efficiency and reliability. However, these results do highlight how stopping latency impacts user perception. Overshooting is very frustrating to the user, and enabling them to stop intuitively where they would like to is very important. LDTM can have large stopping latency, as users must stop looking at the world to look down at a button.

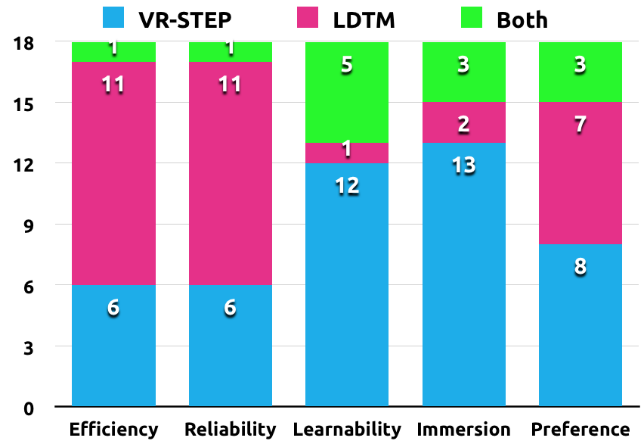

Figure 2. Ranking of navigation methods based on usability criteria.

Some participants found a way around this by looking forward and down while navigating, but if your goal is to create an immersive world, this is obviously not what you want your users to be doing. The trade off we made to allow for a fair comparison did clearly demonstrate participant's qualitative preferences for VR-STEP.

For VR-STEP, we selected a step detection algorithm [35] with a low computational overhead to assure a high frame rate. Some participants initially expressed that our step detection wasn't sensitive enough, but when we instructed them to "jog in place" so their head bounced, they were able to use VR-STEP quite efficiently. If a user is already moving, it may be challenging –based on visual feedback alone– to know whether the system has detected a step. When we added audio feedback, e.g., the sound of a footstep, users were able to better anticipate how the system works and modify their stepping input accordingly.

LIMITATIONS & FUTURE WORK

VR-STEP requires users to stand and provide continuous stepping input. This may not be desirable when using mobile VR for longer periods of time. However, popular VR headset manufacturers recommend that users take regular breaks from using VR [3], so standing may not be a major issue. VR-step moves the user in the direction of their gaze, but this is rather limited compared to how humans navigate in real life. For future work, we will explore the use of head tilt to indicate a direction for navigation which may allow users to walk other directions. For step detection feedback, we will explore haptic feedback, since audio feedback may be obfuscated in gaming contexts. Besides detecting steps for walking, other feet related actions, such as jumps or stomps may be detected using acceleration. We were able to implement jumps in VR-STEP using a threshold for vertical acceleration. A promising application area of VR-STEP is exercise games [18]. Jogging-in-place has already been found to get users into moderate-to-vigorous levels of exercise [20]. In addition, bringing the immersion of VR to exergames could engage players for longer, which may stimulate larger health benefits. This may address some of the current criticisms of exergames that they don't provide sufficient health benefits [2].

REFERENCES

1. Works with Google Cardboard Guidelines and Best Practices, http://static.googleusercontent.com/media/www.google.com/en//get/cardboard/downloads/wwgc_best_practices.pdf. (September 2015).
2. 2012. New York Times, 'Exergames' Don't Make Children Fitter, http://parenting.blogs.nytimes.com/2012/06/25/exergames-dont-make-children-more-fit/. (June 2012).
3. 2014. Oculus Best Practices: Health and Safety Warnings, https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_app_health/. (January 2014).
4. 2015. Tuscany Demo, http://play.google.com/store/apps/details?id=com.FabulousPixel.TuscanyDive. (August 2015).
5. Ilias Apostolopoulos, Daniel S. Coming, and Eelke Folmer. 2015. Accuracy of Pedometry on a Head-mounted Display. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2153–2156. DOI: http://dx.doi.org/10.1145/2702123.2702143
6. Doug Bowman, David Koller, Larry F Hodges, and others. 1997. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *Virtual Reality Annual International Symposium, 1997., IEEE 1997*. IEEE, 45–52.
7. Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. 2004. *3D user interfaces: theory and practice*. Addison-Wesley.
8. Agata Brajdic and Robert Harle. 2013. Walk Detection and Step Counting on Unconstrained Smartphones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 225–234. DOI: http://dx.doi.org/10.1145/2493432.2493449
9. Rudolph P Darken, William R Cockayne, and David Carmein. 1997. The omni-directional treadmill: a locomotion device for virtual worlds. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. ACM, 213–221.
10. David Dobbelstein, Philipp Hock, and Enrico Rukzio. 2015. Belt: An Unobtrusive Touch Input Device for Head-worn Displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2135–2138.
11. Jeff Feasel, Mary C. Whitton, and Jeremy D. Wendt. 2008. LLCM-WIP: Low-Latency, Continuous-Motion Walking-in-Place. In *Proceedings of the 2008 IEEE Symposium on 3D User Interfaces (3DUI '08)*. IEEE Computer Society, Washington, DC, USA, 97–104. DOI: http://dx.doi.org/10.1109/3DUI.2008.4476598
12. Randal C Foster, Lorraine M Lanningham-Foster, Chinmay Manohar, and et al. 2005. Precision and accuracy of an ankle-worn accelerometer-based pedometer in step counting and energy expenditure. *Prev Med* 41, 3-4 (2005), 778–83. DOI: http://dx.doi.org/10.1016/j.ypmed.2005.07.006
13. 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*, Vol. 45. SAGE Publications, 1896–1900.
14. Ji-Sun Kim, Denis Gračanin, and Francis Quek. 2012. Sensor-fusion walking-in-place interaction technique using mobile devices. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*. IEEE, 39–42.
15. William B Lathrop and Mary K Kaiser. 2002. Perceived orientation in physical and virtual environments: changes in perceived orientation as a function of idiothetic information available. *Presence* 11, 1 (2002), 19–32.
16. Catherine G Ling and Sheila O Smith. 2010. Alternative placements for women wearing pedometers. *J Am Acad Nurse Pract* 22, 5 (May 2010), 264–9. DOI: http://dx.doi.org/10.1111/j.1745-7599.2010.00502.x
17. Michael Meehan, Sharif Razzaque, Mary C Whitton, and Frederick P Brooks Jr. 2003. Effect of latency on presence in stressful virtual environments. In *Virtual Reality, 2003. Proceedings. IEEE*. IEEE, 141–148.
18. Florian 'Floyd' Mueller, Martin R. Gibbs, and Frank Vetere. 2008. Taxonomy of exertion games. In *OZCHI '08: Proceedings of the 20th Australasian Conference on Computer-Human Interaction*. ACM, New York, NY, USA, 263–266. DOI: http://dx.doi.org/10.1145/1517744.1517772
19. Noah Nelson. 2015. NPR - All Tech Considered - Virtual Reality's Next Hurdle: Overcoming 'Sim Sickness' http://www.npr.org/sections/alltechconsidered/2014/08/05/338015854/virtual-realitys-next-hurdle-overcoming-sim-sickness. (August 2015).
20. Nicolaos Papazoglou, Panayotis Loussidis, Andreas Giannoulopoulos, and et al. 1991. Comparison of respiratory response of jogging in place and Bruce treadmill exercise test. *Clinical cardiology* 14, 11 (1991), 913–916.
21. Patrick Péruch, Loïc Belingard, and Catherine Thinus-Blanc. 2000. Transfer of spatial knowledge from virtual to real environments. In *Spatial Cognition II*. Springer, 253–264.
22. Sharif Razzaque, David Swapp, Mel Slater, and et al. 2002. Redirected walking in place. In *ACM International Conference Proceeding Series*, Vol. 23. 123–130.

23. Bernhard E. Riecke, Bobby Bodenheimer, Timothy P. McNamara, andet al. 2010. Do We Need to Walk for Effective Virtual Reality Navigation? Physical Rotations Alone May Suffice. In *Proceedings of the 7th International Conference on Spatial Cognition (SC'10)*. Springer-Verlag, Berlin, Heidelberg, 234–247. http://dl.acm.org/citation.cfm?id=1887093.1887121
24. Mel Slater, Martin Usoh, and Anthony Steed. 1994. Steps and ladders in virtual reality. In *Proceedings of the ACM Conference on Virtual Reality Software and Technology*. World Scientific, 45–54.
25. Mel Slater, Martin Usoh, and Anthony Steed. 1995. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)* 2, 3 (1995), 201–219.
26. James Templeman, Patricia S. Denbrook, and Linda E. Sibert. 1999. Virtual Locomotion: Walking in Place through Virtual Environments. *Presence* 8, 6 (Dec 1999), 598–617. DOI: http://dx.doi.org/10.1162/105474699566512
27. Léo Terziman, Maud Marchal, Mathieu Emily, and et al. 2010. Shake-your-head: Revisiting walking-in-place for desktop virtual reality. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*. ACM, 27–34.
28. Léo Terziman, Maud Marchal, Franck Multon, and Anatole Lécuyer. 2011. Comparing virtual trajectories made in slalom using walking-in-place and joystick techniques. In *EuroVR/EGVE Joint Virtual Reality Conference*. Eurographics.
29. Michel Treisman. 1977. Motion sickness: an evolutionary hypothesis. *Science* 197, 4302 (1977), 493–495.
30. Martin Usoh, Kevin Arthur, Mary C Whitton, andet al. 1999 Walking > walking-in-place > flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 359–364.
31. Mark Ward, Ronald Azuma, Robert Bennett, and et al. 1992. A demonstrated optical tracker with scalable work area for head-mounted display systems. In *Proceedings of the 1992 symposium on Interactive 3D graphics*. ACM, 43–52.
32. Jeremy D Wendt, Mary C Whitton, and Frederick P Brooks Jr. 2010. Gud wip: Gait-understanding-driven walking-in-place. In *Virtual Reality Conference (VR), 2010 IEEE*. IEEE, 51–58.
33. Betsy Williams, Stephen Bailey, Gayathri Narasimham, andet al. 2011. Evaluation of Walking in Place on a Wii Balance Board to Explore a Virtual Environment. *ACM Trans. Appl. Percept.* 8, 3, Article 19 (Aug. 2011), 14 pages. DOI: http://dx.doi.org/10.1145/2010325.2010329
34. Preston Tunnell Wilson, Kevin Nguyen, Alyssa Harris, and Betsy Williams. 2014. Walking in place using the Microsoft Kinect to explore a large VE. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM, 27–33.
35. Neil Zhao. 2010. Full-featured pedometer design realized with 3-Axis digital accelerometer. *Analog Dialogue* 44, 06 (2010).