



Hip-directed walking-in-place using a single depth camera

Luís Bruno^{a,*}, Maurício Sousa^b, Alfredo Ferreira^b, João Madeiras Pereira^b, Joaquim Jorge^b

^a Instituto Politécnico de Beja, Rua Pedro Soares, Apartado 6155, 7800-295 Beja, Portugal

^b INESC-ID/IST/University of Lisboa, R Alves Redol, 9, 1000-021 Lisboa, Portugal

ARTICLE INFO

Keywords:

Walking in place
Virtual locomotion
Navigation control
Virtual environments
Interaction design

ABSTRACT

Walking-in-place (WIP) is a locomotion technique that allows users to travel in virtual environments (VEs) without significantly changing their physical position on the floor. Hip-directed steering (HDS) is a novel physical technique for controlling direction changes in virtual travel using hip movements. We present an WIP-based navigation approach for controlling locomotion in VEs that combines the speed and direction in a scenario similar to a domestic setup in which people interact with a flat screen. Their physical motion data are captured by one depth camera properly aligned with the screen and oriented toward the user.

We approach the characteristically noisy data generated by depth cameras via a user study to determine both the range of values and their robustness from the motion data associated with the joints relevant to WIP speed (knee, ankle and foot) and HDS (spine, hip and shoulder) to derive a reliable technique. Our WIP speed method is supported in a simple vocabulary of five different footstep types. Experimental results show that both the knee and hip provide the most robust data.

We evaluated our techniques via usability tests exercising common locomotion tasks. The results show that users liked both the speed control and comfort afforded by our speed method. Regarding HDS, users reported that the angular-based method allowed them to travel faster and was both more controllable and easier to learn than the time-based method. Our work shows that a single depth camera can be used to combine locomotion and direction control in a simple and affordable setup.

1. Introduction

Controlling navigation in virtual environments can rely on indirect interactions supported by physical devices and graphical user interface controllers. Alternatively, it can be driven by direct interactions based on gestures expressed by different body parts. In particular, walking-in-place (WIP) enables people to control movements in a VE by mimicking real walking movements. In contrast to unrestricted real walking, WIP allows users to operate in small areas. In current approaches, WIP gestures only control the linear movement of a virtual camera. To change direction, different techniques are required. The joystick is probably the most common physical device used for this purpose. However, it renders the hands unavailable for more appropriate tasks, such as selecting and manipulating objects and burdens them with too many control actions. In our approach we use a single depth camera to track different body parts. This low-cost device has considerable potential to provide additional degrees of freedom to control avatar navigation. However, it poses specific problems. Due to camera limitations, a single sensor may not provide reliable skeleton

data due to occlusions.

Display modalities also affect navigation. Most WIP approaches use head-mounted displays (HMD). A few methods operate in a CAVE (Settgast et al., 2014; Razzaque et al., 2002; Zielinski and Brady, 2011) or use a flat screen (Terziman et al., 2010). These devices require distinct techniques to control movement direction changes. Furthermore, different devices present diverse field of Regard (FOR) constraints. When using an HMD, it is possible for users to rotate physically 360° without losing visual feedback. In contrast, inside a CAVE, the allowed body rotation depends on the number of screen walls surrounding the person. Eg in a CAVE with three walls, users cannot rotate by more than 90° without losing the FOR, which may hinder interaction. Finally, when using a single depth sensor its placement can pose problems. To avoid interfering with visualization, sensors should be placed above the display, which increases the likelihood of lower-limb occlusions.

Moreover, in front of a flat screen, body rotation becomes more constrained because user may lose her FOR. Its allowable extent depends on both the screen size and user distance to the display.

* Corresponding author.

E-mail addresses: lbruno@ipbeja.pt (L. Bruno), antonio.sousa@tecnico.ulisboa.pt (M. Sousa), alfredo.ferreira@inesc-id.pt (A. Ferreira), jap@inesc-id.pt (J.M. Pereira), jaj@inesc-id.pt (J. Jorge).

<http://dx.doi.org/10.1016/j.ijhcs.2017.03.006>

Received 18 May 2016; Received in revised form 27 February 2017; Accepted 13 March 2017

Available online 18 March 2017

1071-5819/ © 2017 Elsevier Ltd. All rights reserved.

Indeed, previous studies (Settgast et al., 2014; Williams et al., 2013; Zheng et al., 2012; Langbehn et al., 2015) reported problems using a single depth camera. Although camera placement can be crucial to both data accuracy and noise, our approach demonstrates that proper filtering and the judicious choice of tracked joints can greatly contribute to reliable, controlled motions. This is in contrast to previous methods that relied solely on either foot or torso data collected separately.

In this paper, we present a simple (but not simplistic) approach to reliably control navigation using a single depth camera in front of a large screen display. Our technique tries to ensure the maximum of data accuracy coming from specific joints related to torso-leg movements. Furthermore, through a comprehensive user study, we devised algorithms especially adapted to both the visualization and single-camera constraints.

Our results show that it is possible to control locomotion and direction using a simple affordable setup, which can be amenable to many different scenarios.

The remainder of this paper is organized as follows. In Section 2, we discuss the main related works on controlling speed and direction travel using WIP and other comparative techniques, as well on the data accuracy problems when body joints are captured by one low-cost depth camera. Section 3 details our setup and presents a user study conducted to collect the parameters to implement the proposed algorithms and to choose appropriate skeleton joints. Section 4 describes the design of our speed WIP and HDS approaches. Section 5 describes the usability study and its results for assessing both the performance and user experience of the proposed methods using common locomotion tasks. Finally, we present the main conclusions of our work in Section 6 and identify paths for future research.

2. Related work

Our work builds on related research on walking in place techniques and sensing people's movements using commodity depth cameras.

2.1. Walking in place

Early research on WIP-based navigation techniques for large VEs explored footstep detection algorithms (Slater et al., 1995; Templeman et al., 1999; Feasel et al., 2008; Wendt et al., 2010). Controlling virtual speed is another topic that has been investigated, and various methods have been proposed for this purpose. Some approaches rely on step frequency (Wendt et al., 2010; Nilsson et al., 2014), whereas others rely on scaling of real walking speeds (Williams et al., 2006; Interrante et al., 2007), on foot speed (Feasel et al., 2008) or even on the steps' height and speed combination (Bruno et al., 2013). Some work was focused to improve the user experience, like to understand what speed users find the most natural while walking in place (Nilsson et al., 2014a, 2014b).

The WIP model proposed by Bruno et al. (2013) based its virtual speed formula on the footstep height and speed metrics. A comparison study against the WIP model proposed by Wendt et al. (2010) supported the speed equation mainly in the footstep frequency, showed that the user is more effective, efficient and precise using the first model. This work follows this Bruno et al. (2013) model using those input metrics, but introducing a different approach for capturing lower limb motion, change travel direction, and using a single depth camera rather than a sophisticated IR tracking system. Indeed, precision IR tracking systems using multiple cameras, are expensive and are not available in common home settings.

However, when using a single depth camera, lower-limb occlusions can occur while stepping in place. In addition to the issues related to user performance, other technical aspects of WIP user experience have been addressed in previous studies. Various studies (Williams et al., 2011; Templeman et al., 2006) compared the joystick against gestural

interfaces for spatial orientation tasks. An usability study conducted by Slater et al. (1995) reported that user experience has a higher subjective sense of presence when traveling by walking-in-place (virtual walking) than when they push-button-fly (along the floor plane). Another study by Feasel et al. (2008), in which participants controlled motion in the VE via a chest orientation sensor, found that their WIP performed better compared to joystick-navigation but significantly worse than real walking. Williams et al. (2013) studied the user spatial orientation in an immersive environment and reported that her WIP interface is better than joystick navigation but similar to normal walking.

Different inputs, outputs and control laws of virtual displacement or simulated movements have been explored for developing effective WIP methods. Additionally, different technologies have been used to detect steps in place to WIP techniques: magnetic trackers (Slater et al., 1995; Feasel et al., 2008), force sensors placed on shoe insoles (Templeman et al., 1999) and optical camera trackers (Wendt et al., 2010; Bruno et al., 2013). However, all of these approaches utilize expensive tracking systems to track the position of the knees (Templeman et al., 1999), feet (Bruno et al., 2013), head (Slater et al., 1995) or shins (Feasel et al., 2008). Other techniques relied on physical interfaces, such as custom-made platforms (Bouguila et al., 2003, 2005). The recent survey by Velloso et al. (2015) provides a comprehensive overview of devices and interactions that focus on lower limbs. In order for WIP systems to be used in living room-sized spaces in common house settings, affordable and versatile tracking systems are required to cope with both the dimensions and use of such spaces. This requires researching new WIP techniques that are adapted to the limitations and characteristics of these tracking systems.

In this context, several WIP approaches explored Nintendo™ Wiimote devices with inertial measurement units (Shiratori and Hodgins, 2008; Williams et al. 2011). Shiratori and Hodgins (2008) used Wiimote devices attached to each of the user's legs, and the step motion data were collected by the devices own internal accelerometer. In contrast, Williams et al. (2011) collected the steps input data using the pressure of the feet on a Wii Balance Board. Both approaches require users to be connected to physical devices, which seriously impacts the naturalness of the user experience due to the intrusive nature of these devices, and both works reported footstep lag problems.

2.2. Using depth cameras for navigation control

The depth cameras offer an affordable price range, do not require wearing additional contraptions, and provides a sensing feature suitable for common living rooms, thus making them suitable for domestic settings. However, occlusion by lower limbs becomes an issue and can lead to accuracy problems when capturing motion.

Recent studies adopted skeleton information captured from depth cameras, such as Microsoft Kinect sensors, to control travel in VEs. One of the earliest works using this input device used body posture (not WIP gestures) as a metaphorical joystick to navigate in VEs. The method developed by Khan et al. (2012) tracks the user's head position in the physical room to calculate the distance from the user to the center of the CAVE to predict virtual speed. In the method proposed by Dam et al. (2013), users just point their feet toward the desired direction of movement, which is then calculated based on the direction of the users' feet relative to their waist. However, it is not clear where the depth cameras are positioned inside the physical rooms, and the authors did not report on accuracy problems when capturing the user's skeleton.

Another study (Settgast et al., 2014) described navigation control in a four-sided CAVE using the Microsoft Kinect to capture both the stepping-in-place gestures and user body posture. These authors found that placing the depth camera on top of the front wall yielded considerably better skeleton recognition results than at the ceiling or

at the back opening of the CAVE. Additionally, users needed to stand approximately 1.5 m away from the front wall to be completely visible due to the sensor specifications. Informal user studies related to their WIP gestures suggest that ankle joint data were too noisy and that knee data were apparently more reliable. This result is logical given the employed camera placement. Indeed, the authors reported that the Kinect can only recognize the skeleton if the legs and arms are not occluded and if the maximum allowed body rotation angle is less than 45 degrees. In our work, we attempt to validate and minimize these limitations to capture data from the legs and the torso movement by only one depth sensor.

A preliminary work (Zheng et al., 2012) confirmed that the skeletal data provided by the Kinect from users walking in place are noisy. To better detect steps-in-place, those authors argued that the most robust data could be obtained from the knees. Also, they found the steps are correctly detected when the knee reaches a threshold value of 0.08 m above the standing height. Refuting this conclusion for WIP gestures detection, Williams et al. (2013) found that the most robust data were obtained from the feet (not knees) and that foot occlusion was effectively a problem. This ambiguity motivates our work to identify more accurate data from lower limb joints related to different footstep types. To overcome this limitation, Williams et al. proposed developing a WIP method supported by two depth sensors that are exactly 90° apart and each 2.5 m away from the intended user. One of the two directional methods allows users wearing an HMD to change WIP direction using the torso orientation reported from the Kinect sensor, and the other method relied on gaze direction.

Langbehn et al. (2015) recently introduced a WIP approach with omnidirectional tracking using four cameras placed in a circle around the subject. They used torso leaning angles to scale virtual locomotion speed. Their preliminary user evaluation showed that the setup can be viable. However, leaning angles can be impractical for locomotion at higher speeds.

Our work is based on a powerwall visualization setup that restricts the user rotations to avoid losing her visual feedback. We hypothesize that under these conditions, it will be possible to develop a robust method that processes reliable skeleton data from the torso and lower limbs through only one depth camera. An earlier work (Wilson et al., 2014) used a similar approach to skeletal tracking data using two Kinect sensors placed 90° apart and approximately 3 m away from the user that accurately track the participant and reduce occlusion problems. The authors described a WIP method to overcome some of the limitations of previous works (Williams et al., 2013). To this end, they measured the angle between the hip, knee, and ankle of the participant's legs to detect steps when that angle would reach a certain threshold. However, both approaches do not allow users to change virtual locomotion speed.

The next studies adopted Kinect sensors for tasks other than to control virtual travel direction and speed. Torres et al. (2012) used motion data from the arm and torso together with OpenNI software in a human-robot interaction system. They applied a low-pass filter to smooth out noisy data provided by torso rotation by averaging the two shoulder rotation angles. Another recent work on human posture identification (Monir et al., 2012) not related to WIP also used Kinect sensors to track users' skeletons. Their results based on Kinect SDK show an average accuracy above 94.9 percent in posture classification (standing, sitting, moving forward and bending) from skeletal features using data collected at user-sensor distances ranging from 1.3 m to 3.5 m. The best results were obtained at distances between 2.0 m and 2.5 m. However, these data were not collected from common leg/foot movements. Indeed, most reported studies do not provide details of the position or orientation of the Kinect sensor. Our research aims to determine these parameters to avoid body occlusions in legs and torso movements. A work by Aitpayev and Gaber (2012) not focused in virtual locomotion techniques reported that a single depth camera is not sufficient to provide accurate body data using the Kinect

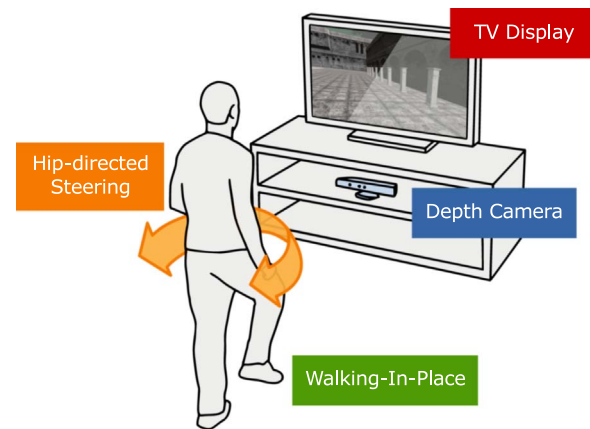


Fig. 1. Our vision for future applications of this work in a domestic setup.

SDK 2 for Windows. They found accuracy problems when measuring joint skeleton data, particularly when users rotated 90° off the camera direction. Previous works on WIP methods supported by Kinect sensors are not clear on the accuracy of data from lower limb joints (foot or knee). Our work contributes to clarifying these points and thus fills the gap through empirical studies to rigorously derive the accuracy from different footstep types and torso rotation. In the next section we describe these studies, before describing the full technique.

3. Estimating motion parameters

In order to implement the speed and HDS methods, we developed a user study to determine which lower limb and torso joints provide the most accurate data when using only one Kinect sensing camera, as well as to estimate the parameters required to generate the equations to control the travel motion. This section describes the system setup and the user study to select the more appropriate joints to those methods fig. 1.

3.1. Setup

Our WIP system operates in a laboratory with a large wall-mounted display and a single depth camera, as depicted in Fig. 2. The wall display is comprised of a flexible screen of 4x2.25 square meters in a

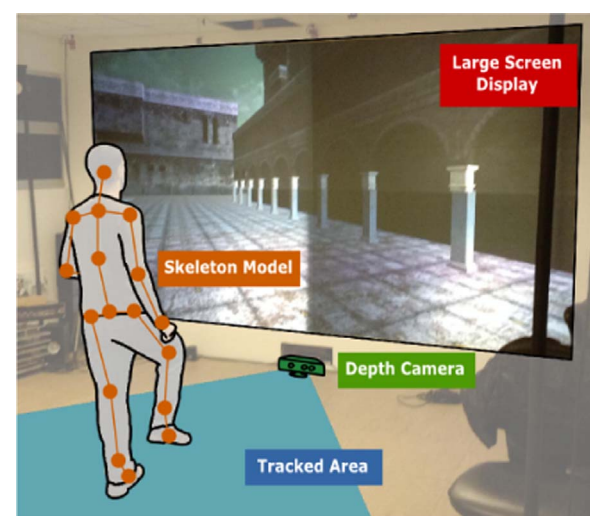


Fig. 2. Setup of the locomotion system where the user stands about 3.0 m from the screen.

2x2 tiled system of projectors in rear-projection mode. The adopted camera was the Microsoft Kinect V1 at 30 Hz sampling rate. It includes a visualization module to display a virtual world and a Kinect-based user motion tracker to identify body gestures and navigate the virtual environment. The large display is a 2x2 High Definition tiled screen in rear-projection mode at 8.6 mega-pixel resolution. The display dimensions create spatial constraints for positioning the Kinect camera, restricting the range of optimal user detection between 1.2 and 4.0 m.

Despite the spatial restrictions of our setup and the Microsoft Kinect product recommendations, user tracking can be successfully achieved with the depth sensor on the floor at a tilt angle that fits the entire user in the camera's field-of-view. The sensor is placed on the floor tilted 10.0° upward. Participants stand approximately 3.0 m from the camera and are free to move within a 2x2 meter area in front of the display, as depicted in Fig. 2. The Kinect-based user motion tracker software module is responsible for recognizing people and follows their actions by means of accessing and controlling the depth camera. It relies on using Microsoft's Kinect for Windows SDK *Kinect for Windows SDK, 2016-05-01* software which estimates the skeleton model as depicted in Fig. 2. With such data, in every new frame of the camera, we can obtain the position of the joints and segments that comprise the calculated user skeleton. In the Kinect skeleton model, each lower limb is divided into segments that contain joint data for the foot, ankle and knee, which we use to compute the navigation speed within the virtual environment. The SDK also provides us individual joint orientation relative to the camera position. Our steering technique relies on the orientation of the user's torso. We get these data through three joints that comprise the upper body of the model: shoulder center (referred below as shoulder), spine and hip center (referred below as hip). The use of just one single depth sensor can limit the tracking efficacy when some joints are occluded by other body parts. A slight rotation of the user's body can occlude the joints of at least one leg. A higher elevation of the knee can cause miscalculations of the torso joints.

In most cases, the SDK is able to infer the position of an occluded joint through a predicting model. However, this feature returns noisy and inaccurate values that may cause erroneous calculations by the WIP algorithm and negatively impact the fluidity of the navigation. Fortunately, the SDK returns a parameter for each joint that represents its tracking confidence, ranging from Tracked to Inferred values. Not all the joint positions labeled as inferred are accurate. Our approach strongly depends on accurate measures of user height and the sensor's ability to detect steps. The user height is computed in real time by adding the lengths of each segment that comprise the upper body, head and one of the legs. We add 0.1 m to the result to account for the distance between the sensor-detected head joint to the top of the user's head. To detect steps, our user tracker needs to differentiate between the lower limb's rest state (on the floor) and the raised state during gait. Rather than estimating the ground plane, our approach detects the onset of a step via height differences between the joints of each lower limb.

3.2. Collecting data from users

We conducted a preliminary study to assess the accuracy of data from three lower limb joints (ankle, foot and knee) and from three torso joints (hip, shoulder and spine) sensed by only one depth camera, as shown in Fig. 3. This study allowed us to estimate specific kinematic parameters from five different footstep types and torso rotation actions. The metrics analyzed are the footstep height and speed computed from these joint data, referred to below as height and speed metrics. The footstep types describe different gestures, which are combinations from those metric features: (i) short and slow (SS), (ii) short and fast (SF), (iii) medium and moderate (MM), (iv) long and slow (LS), and long and fast (LF). The data was collected in a controlled experiment using the setup and apparatus described previously, and participants were asked to execute the requested tasks.

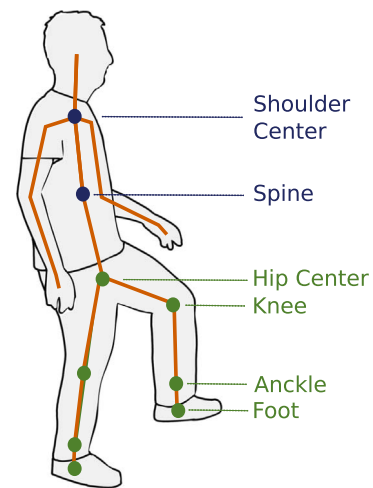


Fig. 3. Skeleton model with torso joints (blue), used for steering, and lower limb joints (green), used for speed control, sensed by camera. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2.1. Tasks

The participants were asked to execute two different task types: (i) stepping-in-place without rotating their torso (linear tasks) and (ii) stepping in place while rotating their torso clockwise, from 0° to 90°, and counterclockwise, from 0° to -90° (nonlinear tasks).

The linear tasks simulates the user body movements to travel forward. All these tasks had no visual feedback from their actions in a virtual environment. However, the user body and head should be perpendicularly oriented towards the wall screen while receiving the appropriate instructions to 5 different tasks, each one corresponding to one footstep type (SS, SF, MM, LS and LF). For each of these tasks, the participants were asked to step-in-place with the specified footstep gestures lasting 10 s.

The non-linear tasks simulates the body movements to change the travel direction and were divided into 10 different sub-tasks (5 different footstep types (SS, SF, MM, LS and LF) X 2 opposite body rotations). Each task took 10 s. Initially, the users' bodies were oriented toward a screen, then they had to rotate clockwise or counterclockwise, from 0° to 90°, and finally, they had to rotate to the initial body orientation. In these actions user should orient her head to large display.

Each task was repeated in two similar sessions to avoid data loss or other problems that could occur during the experiment. One interval of 5 seconds was used between two consecutive sessions. At the beginning of each task session, instructions related to the next task were presented to the user in a large screen, as shown in Fig. 4. This allows the user to focus on their tasks and avoids misunderstandings. The moderator of the experience was available to help the user when questions arose.

3.2.2. Participants

Twenty participants, thirteen males and seven females, ranging in age from 22 to 30 years (mean=24.1 years) participated in the user study and were recruited from our local university campus. Their height ranged from 1.55 m to 1.95 m (mean=1.73 m). The Shapiro-Wilkinson normality test revealed that height data sample presents a normal distribution ($p=0.498$). No participant had motor or physical impairments.

3.2.3. Experimental procedure

At the beginning of the experiment, the participants were informed of the study purpose and the script was explained. To introduce the stepping-in-place technique, the participants experimented for at least 10 minutes with a WIP prototype to control the virtual locomotion in

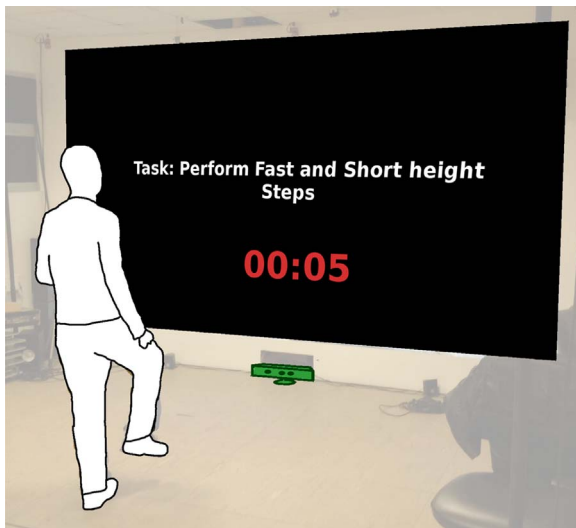


Fig. 4. User stepping-in-place following the task instructions on a large screen.

one VE. Thereafter, the experiment moderator demonstrated the tracking confidence state values obtained from skeleton tracker data structure. The height and speed are the metrics that describe WIP gestures. The maximum rotation angle is the metric that describes the torso orientation threshold.

3.2.4. Metrics

In this work, the data accuracy metric is measured through the tracking confidence state values obtained from skeleton tracker data structure. The height and speed are the metrics that describe WIP gestures. The maximum rotation angle is the metric that describes the torso orientation threshold.

3.2.5. Design and data analysis

The experiment design was within-subjects, where each participant performed the five linear and ten non-linear tasks. Therefore, the original data consisted of 20 participants x 5 linear tasks x 10 nonlinear tasks (1000 samples). The “height”, “speed”, and “maximum rotation for non-linear tasks” data metrics presented normality in their data samples, and then two-way repeated-measures ANOVA (tasks x joints) was applied. Greenhouse-Geisser's sphericity corrections were applied whenever Mauchly's test of sphericity showed a significant effect. The post-hoc test of pairwise multiple comparisons for each factor was adjusted by Bonferroni test. The tracking confidence state and the maximum rotation for linear tasks samples did not present a normal distribution; thus, the non-parametric Friedman test was used to determine the main effect of the factors step type and joint. The Wilcoxon test was used to determine the pairwise differences among factor levels. The level of significance used for all of the hypothesis tests was $\alpha = 0.05$.

3.2.6. Results and discussion

The quantitative data collected from the trials were statically analyzed to understand the comparative effects of the footstep type and joint on the mentioned metrics. All the trials data was 100% kept.

Height metric The maximum height was achieved for each lower limb joint and for each footstep type was recorded to compute the respectively average values. As shown in Table 1, the average maximum height naturally increases from the short steps (slow to fast) to the long steps (slow to fast). The two-way repeated-measures ANOVA revealed

Table 1

Average/St. Error of height and speed, and average of confidence tracking state for footstep types.

Step Metric	[SS]	[SF]	[MM]	[LS]	[LF]
Height (m)	0.07/0.09	0.07/0.07	0.15/0.01	0.32/0.02	0.36/0.02
Speed (m/s)	0.39/0.04	0.51/0.03	0.69/0.05	0.90/0.05	1.45/0.075
Tracking Confidence State Ratio	99.72	99.75	99.86	99.23	99.27

the significant effect of the step type ($F_{2,154,28,006} = 119.102$, $p < .0001$) on the maximum height metric but did not show a significant effect of the joint ($F_{1,301,16,916} = 3.488$, $p = .070$). No interactions between the step type and the joint were found ($F_{3,756,48,827} = 2.282$, $p = .078$). Regarding the footstep type, the post-hoc test for pairwise multiple comparisons revealed differences between all the pairs ($p < .0001$) except for the two short type steps ($p = 1.000$) and for the two long type steps ($p = .348$). These results suggest a consistent relationship between the step height and the footstep types and can be safely used to support a WIP design with different footstep gestures.

The average height values obtained for each joint are very similar: foot (0.19 m), ankle (0.19 m) and knee (0.20 m). The ANOVA reveals no significant differences among these three levels, indicating that this metric is invariant to the joint type, and thus, this feature does not contribute to the selection of the most appropriate joint for the WIP system design.

Speed Metric The footstep average speed was calculated for each lower limb joint and for each footstep type from the relation between the distances (ascendant and descendant) performed by each joint with the time spent for this action. As shown in Table 1, the average speed increases from the shorter steps (short/slow and short/fast) to the medium height steps, and finally to the longer steps (long/slow and long/fast). The two-way repeated-measures ANOVA reveals the significant effect of the step type ($F_{4,52} = 103.902$, $p < .0001$) as well as the effect of the joint ($F_{1,379,17,921} = 8.840$, $p = .005$) on the speed metric. Interactions between step type and joint were found ($F_{3,114,40,477} = 3.126$, $p = .035$). The post-hoc test supported by pairwise multiple comparisons reveals significant differences for all the footstep types on the speed metric ($p < .05$). These results suggest that these different WIP footstep gestures can be classified through the speed metric and could be used to design an perceptible method to user. Regarding the joint metric, the foot presented the fastest speed (0.83 m/s), followed by the ankle (0.80 m/s), and finally by the knee (0.73 m/s). The post-hoc test shows differences between the foot and the knee ($p = 0.004$). No significant differences were found between the foot and the ankle ($p = 0.062$) or between the ankle and the knee (0.187). If one of the joints had a proven higher speed value among the others then it can provide a wider speeds range to control the virtual speed. But, since that was not the case, then it is not possible to choose one joint over the others to our WIP method design.

Data accuracy from lower limb joints The accuracy level of the skeleton joint input data is critical for the usability of the WIP technique. From each frame transmitted by Kinect, the tracking confidence state (tracked or inferred) is defined for each joint position. For each footstep, the ratio between the frames labeled with the tracked state by the total number of frames captured was calculated. This is called the average confidence state ratio. Table 1 shows the average ratio for each footstep type. The two faster steps (LS and LF) present on average the lowest confidence values, and the moderate steps (MM) present the best values. The slower and shorter steps (SS and SF) present better confidence than the faster steps. The Friedman test shows that the step type has a significant effect ($p = 0.005$) on this metric. The post-hoc Wilcoxon test confirms significant differences

between all the pairs, except for the two slower steps ($p=0.120$), the two faster steps ($p=0.151$), and MM against SS ($p=0.070$) and SF ($p=0.122$). This result indicates that the tracking data obtained from the moderate footsteps are more robust than those from the other footstep types. Regarding the joint analysis, the average confidence ratio indicated that the knee (99.94%) presents the best value, followed by the foot (99.71%), and finally by the ankle (99.05%). The Friedman test reveals the significant effect ($p=0.001$) of the joint on this metric. The post-hoc Wilcoxon test reveals significant differences between the foot/ankle ($p=0.041$), the ankle/knee ($p=0.005$) and foot/knee ($p=0.290$). As the knee presents the best ratio, the stats results confirms that this joint ensures the most robust data tracking from Kinect.

Maximum rotation angle from torso joints for linear tasks To determine the linear threshold angle in our HDS methods, the maximum rotation angle metric for the footstep types and for joints (hip, spine and shoulder) was processed. The mean amplitude values for each joint are as follows: hip (7.85°), spine (3.75°) and shoulder (3.85°). The mean values for each footstep type are as follows: SS (3.65°), SF (2.65°), MM (5.70°), LS (8.20°) and LF (7.50°). The Friedman test reveals the significant effect ($p < .0001$) of the joint on this metric. The post-hoc Wilcoxon test reveals significant differences between the hip/spine ($p < .0001$) and the hip/shoulder ($p < .0001$) but not the spine/shoulder ($p=.882$). These results reveal that the torso rotation amplitude for linear tasks presents values below 10° , independent of the joint source. On average, the hip joint provides a higher amplitude than the spine and shoulder joints. Supported by the Friedman test, a significant effect of this metric on the step type is found ($p < .0001$). All the step type comparison pairs are different, except the two shortest SF/SS ($p=.228$) and the two longest LF/LS ($p=.225$). The step type that produces on average the highest amplitude is the LS (8.20°).

Maximum rotation angle from torso for non-linear tasks The maximum rotation angle metric for non-linear tasks was processed to infer significant differences for the footstep types and for joint (hip, spine and shoulder) conditions. The mean values for each joint are as follows: hip (65.0°), spine (36.0°) and shoulder (35.0°). The mean values for each footstep type are as follows: SS (42.6°), SF (44.4°), MM (46.4°), LS (46.1°) and LF (48.0°). The two-way repeated-measures ANOVA reveals the significant effect of the joint ($F_{2,38} = 89.181$, $p < .0001$) on this metric, but no effects were detected for the step type ($F_{4,76} = 1.408$, $p = 0.239$), and no interactions between step type and joint were found ($F_{3,996,75,920} = 1.643$, $p = 0.172$). The post-hoc test for pairwise multiple comparisons reveals differences between the hip, spine and shoulder joints ($p < .0001$ for both pairs) and did not reveal differences between the spine and shoulder joints ($p=1.000$ for both pairs). These results reveal that the hip joint provides the highest torso angle amplitude for the same tracking confidence state ratio, ensuring a wider amplitude range for the direction control algorithm.

From the previous results we selected the knee joint to input our speed method due the robustness data tracking from Kinect to our scenario. The hip joint was selected as input to methods that allow user to control directional changes using body rotations in front a large screen. Next section explains the design and implementation of the speed and HDS methods proposed in this work.

4. Our approach

This section describes the design of the speed model and the two alternative HDS models taking into account the user study results, described in Section 3. The proposed interaction rules and algorithms should assure the lowest possible number of body occlusions to one camera. It allows us to obtain the maximum data accuracy from the selected skeleton joints to provide a fluid experience for user. Based on the previous study (in Section 3), we state the knee joint meets these requirements to our speed method and hip joint to our proposed HDS methods.

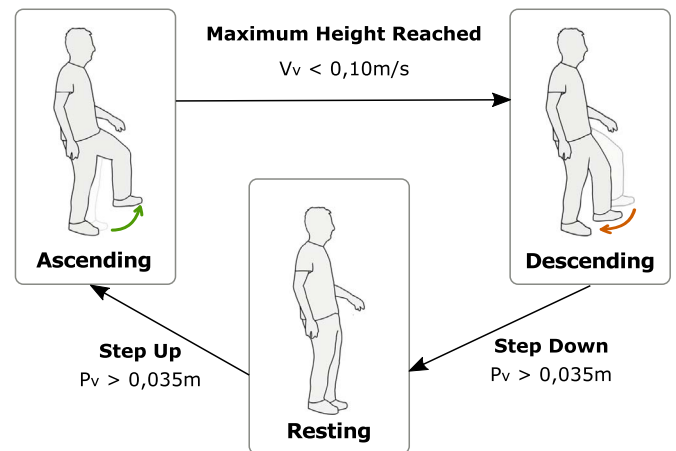


Fig. 5. Foot states and conditions that trigger WIP events.

4.1. WIP speed algorithm model

The main guidelines to design the speed model consisted of defining suitable interaction rules responsible for managing input data from knee and deriving the speed control law that describes the avatar movement.

4.1.1. Interaction rules

Our proposed speed model is supported by three state machines for each left and right knee, and for virtual travel. The model of these states machines are similar to those reported in other research works in this area (Wendt et al., 2010; Bruno et al., 2013; Francese et al., 2012). As depicted in Fig. 5, the state machine for the predefined knee joint is composed of three states: ascending, descending and resting. The events that trigger the transitions are the step up, maximum height reached and step down. The ascending state occurs when the knee joint vertical position (P_v) exceeds a minimum height threshold ($P_v > 0.035 m$) from its resting position. The descending state is triggered when the knee maximum height is detected which occurs when the vertical velocity decreases to a predefined threshold ($V_v < 0.10 m/s$). The resting state occurs when the knee height is lower than the minimum height threshold ($P_v > 0.035 m$). Those threshold parameters were empirically calculated through different experimental tests with some voluntary colleagues (male and female with different heights) that contribute to get a robust footstep detection model from the knee data using only one depth camera. The computed value of the virtual speed is updated in two phases of the step cycle. The first one occurs when the user knee reaches the highest point, and the second one occurs when the resting state is detected. The computed speed value from our algorithm is divided by the imaging frame rate to get the step distance performed for each frame. Stopping travel occurs when the user stops the swing motion of the legs for a predefined period. The detection of this event requires defining a timeout that allows the system to predict whether a user intends to stop or not. The timeout value should avoid not only stopping latency but also identifying false stop events. The data collected from user tests identify the relationship between the speed value of the descending phase and the joint time to rest. The algorithm for predicting the stop timeout is supported by a linear interpolation between the descending speed and the resting time on the floor.

4.1.2. Speed control equation

Our speed algorithm intends to relate the virtual speed to the footstep height and speed. Higher and faster steps should be correlated with higher virtual speeds (Bruno et al., 2013). However, because users have different heights, as depicted in Fig. 6, the maximum virtual speed

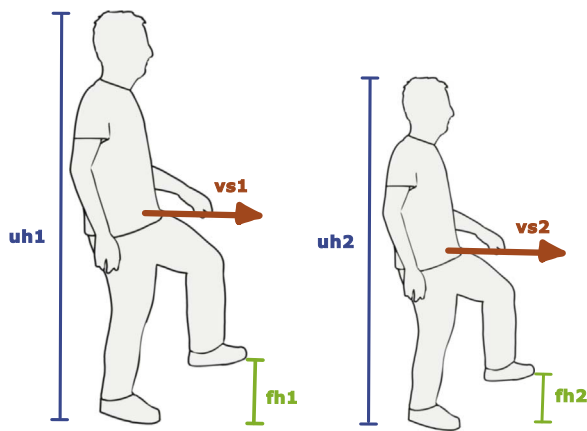


Fig. 6. Different user heights cause different footstep heights.

achieved by different users will differ. Therefore, to overcome this limitation, the designed algorithm should incorporate a scale factor that attenuates these differences. This is very useful when adults and children are exploring the same VE and want to experience similar locomotion levels. The formula (1) was developed empirically to obtain the dependent variable vs (virtual speed) that has the following independent variables: footstep height (fh) defines the maximum vertical distance performed by the knee in each step; footstep speed (fs) defines the vertical mean speed during the ascending and descending phases of each step; user height (uh) is automatically captured from the Kinect software through the analysis of the skeleton data; and the speed scale constant (ss) allows the equation to adapt the speed according to the needs for each virtual environment and locomotion requirements.

$$vs = ss * (fh/0.36) * (1.95/uh) + (fs/1.45) \quad (1)$$

In our hypothesis, the virtual speed is more predictable for the user if it is supported in linear relationships. The $(fh/0.36)$ and $(fs/1.45)$ describes ratios where the denominators represent respectively the maximum mean values of the footstep heights and speeds, as shown in Table 1. The (fh) variable value is adjusted by the ratio $(1.95/uh)$. The chosen height threshold (1.95) matches to the tallest user in the previous user study (in Section 3) that is also representative of the highest people that can use the system. This solution allows shorter people to compensate their shorter vertical footsteps, assuring that they have similar speed values relative to taller people, as depicted in Fig. 6. The user height (uh) is automatically captured from the Kinect software when the system starts through the analysis of the skeleton data.

By simulating this formula for a mean values, as one user with a common height of 1.75 m , a mean footstep height (fh) equal to 0.18 m , and a moderate speed (fs) equal to 0.73 m/s , and a scale factor (ss) equal to 1, then the virtual speed obtained is 1.06 m/s . This value corresponds to a mean speed in real walking at a moderate gait. The relationship between those input may serve as reference for the user calibrate the desired speed.

Stopping travel occurs when the user stops the swing motion of the two legs for a predefined period. To trigger this event, it is necessary to define a timeout that allows the system to detect whether a user intends to stop. The timeout value should avoid false stops and minimize the stopping latency. From the data collected in the previous study, we identified a relationship between the speed value in the descending phase and the time period where the user supports their feet on the floor. From this relation, one linear interpolation was established, as shown in Eq. (2).

$$timeout = -0.29 * abs(ds) + 0.84 \quad (2)$$

The interpolation is supported in the speed range from 0.14 m/s to

1.7 m/s , and in the time interval from 0.8 s to 0.35 s . The speed value (ds) of the descending phase is the independent metric for predicting the timeout. When the user supports their feet beyond the timeout value, the virtual motion is stopped. Experimental tests with voluntary users showed that this formula is suitable and does not cause false positives.

The following section describes the designed models for controlling travel directional changes of the WIP.

4.2. HDS algorithm model

Our HDS model design considered the visualization restrictions on a single large flat screen, which significantly decreases the freedom of user movements. In this context, the proposed design was adapted to the hip joint that was selected from the previous user study described in section 3. This joint assures the more accurate data and the widest amplitude range when user rotate her body while stepping in place. This model should provide the capability of controlling the virtual travel either in a fixed direction (linear path) or changing the direction (non-linear path) in omnidirectional way.

4.2.1. Interaction rules

Our design defined two yaw threshold angles for the hip joint rotation that were derived from the previous user study (in section 3). The first one, called linear threshold, refers to the maximum hip amplitude (defined by α) that allows user to move in a fixed linear path; above this amplitude, the user can change the travel direction induced by torso rotations, as depicted in Fig. 7. The second threshold angle, called non-linear threshold (defined by β), represents the maximum hip rotation that is processed by the system to avoid body occlusions for the sensing camera and prevents users from losing the visual output on the large flat screen.

The α value was supported in statistical analysis of the maximum torso amplitude while the user is stepping-in-place without rotate voluntarily her body. On average, the hip presents an amplitude of 7.85° for all the footstep types and the highest amplitude is 8.20° for LS steps. Based on this value plus a safety margin (about 2.00°), we selected α to 10° . This value was tested by some informal users and performs well. The non-linear threshold was based on the visualization restrictions of the flat screen and the sensing camera occlusions. Therefore, if the torso rotates by more than 45° , body occlusions and gazing outside the screen can occur more frequently. Based on this value, we choose β value to 25° as the non-linear threshold. The sum of the α plus β thresholds is equal to 35° , which provides a safety margin relative to the limit of 45° .

4.2.2. Control methods

From the empirical analysis of users rotating their torsos to control travel directions, we proposed two different methods. The first is primarily dependent on the hip angle amplitude around the yaw axis, and we called it the “angular method”. The second is supported on the time metric during hip rotation and is called the “time method”. For both methods user has to turn her whole body beyond the linear

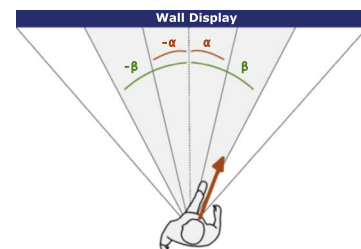


Fig. 7. Torso orientation within the linear (alpha) and non-linear (beta) thresholds that defines the travel direction.

threshold to start changing travel direction. When user get the desired path direction then he can recenter her body to the linear amplitude. In the angular method, the directional change is directly proportional to the torso angle amplitude. In the “time method”, the directional change depends on the time spent by the user while rotating her torso. The angular method has the advantage of being more natural because the user perceives a relation between torso physical movement and directional effect on VE. However, it has the disadvantage of a restricted torso rotation amplitude to avoid the user looking outside the flat screen and camera to obtain noisy data. The time method has the advantage of not requiring the user to perform expressive rotations amplitudes, keeping him more focused on the flat screen and avoiding leaving the frustum of the camera sensing. The angular method offers also to the user a proprioceptive cue that is more tangible and objective than the time perception. Due to the impossibility of designing a 1:1 mapping between the physical and virtual rotation, it is necessary to use a scale factor for the virtual camera rotation to accommodate the omnidirectional travel requirement. The option to apply this scale was based on the angular speed concept for the virtual camera. The angular speed range determined for the two HDS methods was defined from 6.5°/s (this small value is very little perceptible to user) to 150°/s (for very fast directional changes), which were obtained from several empirical tests with users. The angular speed supports the omnidirectional requirement for this approach and its values range allow user to control direction in a perceptible and smooth way. The two designed methods compute the angle offset of the virtual camera supported by a linear interpolation between the range for the respective metrics and angular speed range. The linear interpolation was chosen rather than an exponential interpolation because the mapping between the input values and the resulting directional change is more predictable to users when based on a linear curve according tests with informal users.

Angular Method: When users rotate their hip joint beyond the linear threshold amplitude (10.0°), the virtual angular offset (ao) is computed according the following equation (3). This design was supported by a linear interpolation between the angular speed range (6.5°/s to 150°/s) defined before and the hip yaw angle range (10° to 35°). The value 10° corresponds to the linear threshold that activates the travel directional changes and 35° to the non-linear threshold as defined before. To generate the equation all of those parameters were converted to the respective percentage values.

$$ao = 57.4 * abs(ha) - 5.09 \quad (3)$$

The independent variable ha is a percentage value that defines the current hip amplitude. The obtained ao value defines the angle offset applied sequentially to the virtual camera when people rotate their torso within the non-linear amplitude. This offset should be divided by the frame rate of the imaging device.

Time Method: This method uses the time as the main independent variable to calculate the angle offset of the virtual camera rotation. When users rotate the hip yaw angle beyond the linear threshold, the angular offset (ao) is computed according to the following equation (4). This design was supported by a linear interpolation between the angular speed range (6.5°/s to 150°/s) and time interval (0ms to 2000ms). The angular speed range is the same as the angular method, and the speed range varies from 0ms to 2000ms. This maximum value was obtained from several empirical tests with users.

$$ao = 0.65 * abs(t) + 0.01 \quad (4)$$

The t variable defines the time elapsed in milliseconds since the user began rotating their torso beyond the non-linear threshold. Simulating the equation for feet resting more than 2000ms, then the angle offset (ao) computed is percent value of 150°/s (the maximum value allowed).

Transition between linear and non-linear modes One problem encountered in an exploratory prototype was the lack of smoothness in the visual feedback caused by the abrupt activation of the travel

direction when the linear threshold is crossed.

To attenuate this jitter, one function was applied to produce continuously small directional changes, below the linear threshold, that are not clearly perceptible to user. That was selected an exponential curve, as shown in Eq. 5. This was obtained from an exponential interpolation between the hip yaw angle amplitude (5° to 35°) and the angular speed interval (5°/s to 30°/s). The equation parameters were converted to the respective percentage values which were obtained from empirical studies with users. The result of this equation is only used while the hip angle is lower than 10°.

$$ao = Exp(Log(0.05) + (Log(30) - Log(0.05)) * \frac{ha}{0.25}) \quad (5)$$

The ha variable represents the current hip amplitude. Simulating the equation, the camera rotation offset (ao) is equal to 6.5° percent for the linear threshold (10°). This value causes small changes in the travel direction that are not perceptible to user according reports from informal user tests. These changes are also balanced by the alternative rhythmic movement of the two legs; thus, the travel direction changes are counterbalanced by a complete footstep cycle.

The next section describes the evaluation of our approach comparing the two HDS methods and the performance of WIP speed method through an usability test.

5. Usability testing

We conducted a usability experiment in a controlled environment using the same setup described in Section 3. The main purpose of this experiment was to compare our two directional control methods and check the proficiency level of the developed WIP speed method.

For this experiment, we developed a simple scenario consisting of a virtual environment that combines open spaces with a few obstacles. The overall layout is a wide courtyard enclosed by buildings. The requested task was to walk along a path outlined by several checkpoints (red spheres), as depicted in Fig. 8. At all times, the user was able to visualize at least two of the next checkpoints. The arrays on the ground help user to follow her own path in order to get in range of each sphere. The task ends when the user goes through the last sphere.

5.1. Participants

A total of 12 participants (2 female) were asked to perform our designed experiment. The age of the participants ranged between 21 and 32 ($mean=24$). Their height ranged between 1.55 m and 1.88 m

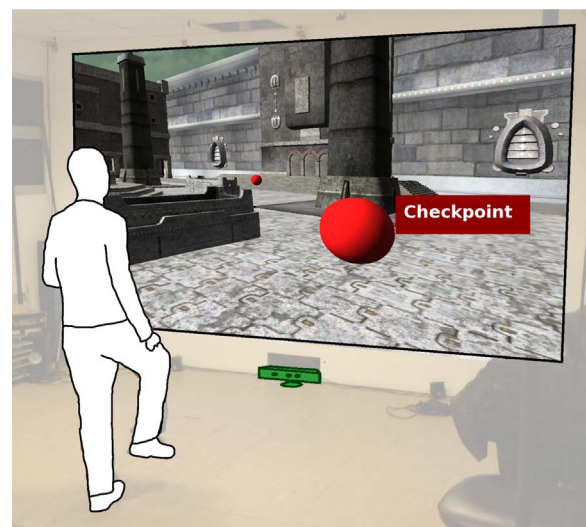


Fig. 8. User walking-in-place to control locomotion in a VE.

(mean = 1.76 m). Additionally, 9 of the 12 participants had previously experienced a virtual navigation interface based on the WIP technique.

5.2. Procedures

Each participant could navigate the predefined path two times (trials) using each directional control method, corresponding to a total of four laps. Half of the users started with the angular method, and the other half started with the time method to counterbalance the bias effect. Each test session began with a brief explanation of the purpose of the experiment, and the navigation control actions associated with each method were explained. Prior to starting the trials, each user was given a minimum of ten minutes to train with the methods by freely navigating in a neutral virtual environment. Then, they executed the trials, in which they traveled through a virtual predefined path while trying to cross the checkpoints. After the trials, each participant completed a questionnaire containing questions about demographic information and previous experience with locomotion control in VE systems and with these navigation methods.

5.3. Metrics

Two key metrics were used: the time to complete each task and the errors that occurred. The time metric represented the elapsed time seconds from the start until the arrival at the last checkpoint. An error occurred when a user did not get in range of each sphere.

5.4. Design and data analysis

We used a within-subjects design in which each participant experiment consisted of two directional methods x two trials. Therefore, the original data consisted of 12 participants x 4 trials. The time metric samples did not exhibit a normal distribution contrary to the angular metric samples; therefore, the Friedman test was used to detect the differences between the the four samples. Regarding the error metric, the data samples did not show a normal distribution, then it was applied the Friedman test too. The pairwise comparisons were supported by the Wilcoxon test. The subjective questionnaire data were analyzed using the Friedman non-parametric test on the different criteria and the Wilcoxon post-hoc test. The level of significance used for all of the hypothesis tests was $\alpha = 0.05$ that was adjusted by Bonferroni method.

5.5. Results and discussion

All trials data were fully used for further statistical analysis. The users navigated the predefined paths performing two trials (T1 and T2) for each HDS control method (AM-angular method and TM-time method). On average, the time spent with the angular method - $AM(T1) = 57.31 s$ ($SD=18.49$) and $AM(T2) = 44.99 s$ ($SD=6.66$) - is lower than using the time method trials - $TM(T1) = 64.26 s$ ($SD=22.78$) and $TM(T2) = 60.28 s$ ($SD=15.01$) - as illustrated in Fig. 9. As far as it concerns to the error metric, users performed on average fewer errors using the angular method - $AM(T1) = 0.084$ ($SD=0.29$) and $AM(T2) = 0.08$ ($SD=0.29$) - than the time method - $TM(T1) = 0.25$ ($SD=0.45$) and $TM(T2) = 0.17$ ($SD=0.39$) - as shown in Table 2. On the time metric side, for the trial T1 on both methods, the Friedman test didn't reveal statistical differences between the two interfaces ($p=.527$). However, when analysing the data from the trial T2, the Friedman test reveals statistical differences between them ($p=.011$) corroborated by Wilcoxon test ($p=.007$). This seemed to indicate that the angular method would provide faster navigation when the user gains more practice on usage. To confirm that, the Friedman test reveals significant differences between trials T1 and T2 for the angular method ($p=.011$), confirmed by post-hoc Wilcoxon test ($p=.007$), but no differences were detected for the time method

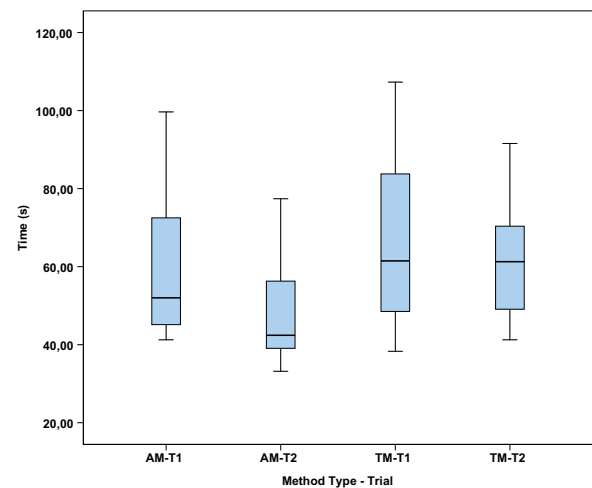


Fig. 9. Time completion average.

Table 2

Performed mean errors and standard deviation from the usability test.

	AM-T1	AM-T2	TM-T1	TM-T2
Mean (St. Error)	0.03 (0)	0.08 (0)	0.25 (0.13)	0.17 (0.11)

($p=1.000$). This result suggests that the angular method allows users to navigate faster and it is easier to learn than the time method.

The mean errors performed by users when crossing the targets were very few for the two methods/trials. The Wilcoxon test does not present significant differences among the four pairs - $AM(T1)/TM(T1)$ ($p=.157$), $AM(T2)/TM(T2)$ ($p=.564$), $AM(T1)/AM(T2)$ ($p=1.000$), and $TM(T1)/TM(T2)$ ($p=.564$). This means that both directional methods provide a similar precision ability to cross path targets and control the travel direction. As referred above, the angular method accomplished better results regarding time metric, which may indicate that users felt more confident when using the hip physical orientation to control travel direction than to manage the time metric. It suggests that physical movements are better perceived as a direct mapping to virtual actions than to intangible metrics such as time.

The post-test questionnaire aims to reflect the opinion of the users about their experience by using the speed method and the two HDS methods. The charts in Fig. 10 presents the mean results from the user ranks on a scale of 1 (the least agreement) to 7 (the most agreement). The first two questions (chart in the left side) are related to the speed method and the others (chart in the right side) to the HDS methods. Question 1 is about the level of control offered by this WIP interface, and question 2 is related to the level of comfort provided. The mean ranks (5.42 and 5.08, respectively) show that users positively appreciate this WIP method for these criteria. The results about control reveals user has a positive perception of the system latency between the amount of time that the users takes on step and the time that she observe one step movement in the VE. Questions 3 and 4 are about the level of control provided by the angular and time methods, respectively. The angular method (5.92) presents a better rating than the time method (4.42). The Wilcoxon test reveals a significant difference ($p=.004$) between the two methods. This result suggests that the users perceive the angular method as yielding more control than the time method. Questions 5 and 6 ask the users how each technique can harm the visualization experience - to loose the screen visual perception. As the angular method was ranked at 5.50, and time one was ranked at 5.83. However, the Wilcoxon test does not reveal significant differences ($p=.279$) between both methods. This suggests that the visual experi-

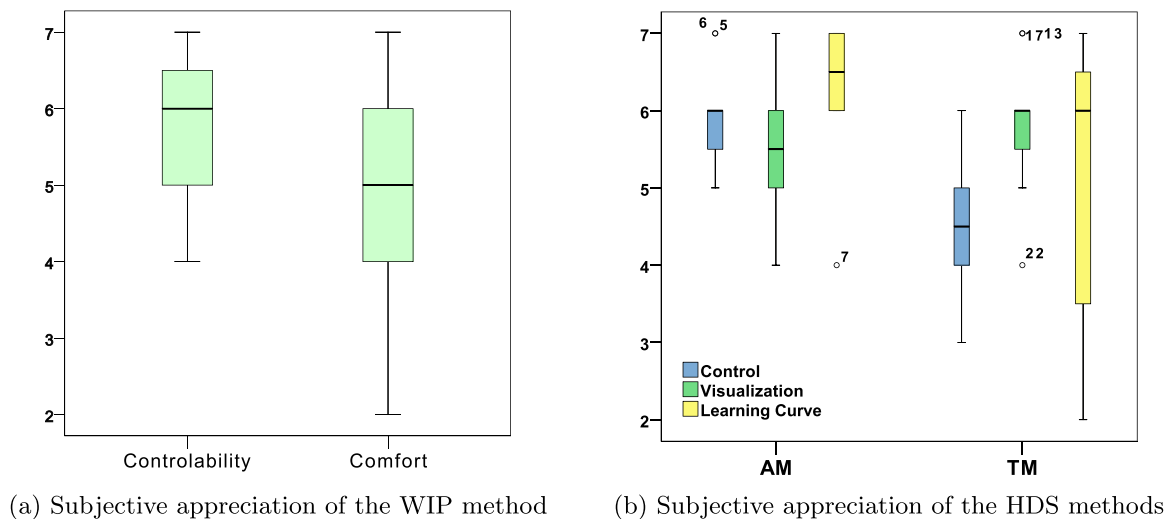


Fig. 10. User Responses.

ence is independent of the method used. Questions 7 and 8 are related to the learning curve felt by users when conducting two consecutive trials for both angular and time methods, respectively. The angular method (6.33) was better ranked than the time method (5.17). The Wilcoxon test confirms it revealing significant differences ($p=0.48$) between both methods too, we concluded that users learned the angular method easier than the time one. This result is consistent with the performance results obtained for the time metric and shows that users better manage the body physical cues than the elapsed time perception. To summarize, taking into account our setting and the VE scenario, the angular method, when compared to the time one, yielded faster navigation, better control and easier to learn.

6. Conclusion and future work

This paper presented three different contributions to control virtual travel using only one inexpensive depth camera while standing in front of a large screen. A within-subjects study yielded both robustness and a range of values from skeleton joints to control speed and direction. These allowed us to provide a WIP approach based on five different footsteps to control different speeds. Our results show that using the knee to describe footsteps and the hip relative to torso yaw are the most robust data source. Furthermore, we were able to adopt the footstep height and speed metrics obtained in a previous study to design a WIP speed equation. This proposed method has the advantage of providing an accessible technique regardless of user height. We also found that people found hip yaw angle to provide a faster, more comfortable and easier to learn means for controlling direction than time-based methods. This suggests that users better manage body physical cues than using relative timeouts. This shows that a single depth camera can be used to combine locomotion and direction control in a simple and affordable setup. In the future, we intend to validate these speed and HDS methods against other setups like HMD visualization to check the effect of different footstep types in immersive virtual environments and understand the sickness induced by the body rotations. An upcoming study using a metronome can also validate our first user study. Also, one intends to personalize the WIP algorithm to the individual user calibrating the respective individual parameters. Other movements such as walking backwards, walking sideways or jumping should be studied to provide a more natural user experience, e.g., in action games.

Acknowledgements

This work was partially supported by the Portuguese Foundation for Science and Technology (FCT) through the projects IT-MEDEX (PTDC/EEISII/6038/2014), TECTON-3D (PTDC/EEL-SII/3154/2012), CEDAR (PTDC/EIA-EIA/116070/2009), A-MOP (UTAP-EXPL/QEQ-COM/0019/2014), and by national funds through FCT with reference UID/CEC/50021/2013.

References

- Aitpayev, K., Gaber, J. 2012. Collision avatar (ca): Adding collision objects for human body in augmented reality using kinect. In: Proceedings of 6th International Conference on Application of Information and Communication Technologies (AICT), IEEE, pp. 1–4.
- Bouguila, L., Hirsbrunner, B., Sato, M., Iwashita, M. 2003. Virtual locomotion interface with ground surface simulation. in: ICAT.
- Bouguila, L., Evequoz, F., Courant, M., Hirsbrunner, B. 2004. Walking-pad: a step-in-place locomotion interface for virtual environments. In: Proceedings of the 6th International Conference on Multimodal Interfaces, ACM, pp. 77–81.
- Bouguila, L., Florian, E., Courant, M., Hirsbrunner, B. 2005. Active walking interface for human-scale virtual environment. In: Proceedings of 11th International Conference on Human-Computer Interaction, HCII, vol. 5, pp. 22–27.
- Bruno, L., Pereira, J., Jorge, J. 2013. A new approach to walking in place. In: Proceedings of Human-Computer Interaction-INTERACT 2013, Springer, pp. 370–387.
- Dam, P., Braz, P., Raposo, A. 2013. A study of navigation and selection techniques in virtual environments using microsoft kinect®. In: Proceedings of Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments, Springer, pp. 139–148.
- Feasel, J., Whitton, M.C., Wendt, J.D. 2008. Llcm-wip: Low-latency continuous-motion walking-in-place. In: Proceedings of IEEE Symposium on 3D User Interfaces, 2008. 3DUI 2008, IEEE, pp. 97–104.
- Francese, R., Passero, I., Tortora, G. 2012. Wiimote and kinect: gestural user interfaces add a natural third dimension to HCI. In: Proceedings of Proceedings of the International Working Conference on Advanced Visual Interfaces, ACM, pp. 116–123.
- Interrante, V., Ries, B., Anderson, L. 2007. Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In: Proceedings of IEEE Symposium on 3D User Interfaces, 2007. 3DUI'07.
- Khan, V., Pekelharing, M., Desle, N. 2012. Efficient navigation in virtual environments: A comparative study of two interaction techniques: The magic wand vs. the human joystick. In: Proceedings of 4th International Conference on Intelligent Human Computer Interaction (IHCI), pp. 1–5. (<http://dx.doi.org/10.1109/IHCI.2012.6481795>).
- Kinect for Windows SDK kinect for windows sdk 1.8. (<https://msdn.microsoft.com/library/dn799271.aspx>).
- Langbehn, E., Eichler, T., Ghose, S., von Luck, K., Bruder, G., Steinicke, F. 2015. 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. (<http://basilic.informatik.uni-hamburg.de/Publications/2015/LEGVBS15>).
- Monir, S., Rubya, S., Ferdous, H.S., 2012. Rotation and scale invariant posture

- recognition using microsoft kinect skeletal tracking feature. In: Proceedings of 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 404–409.
- Nilsson, N.C., Serafin, S., Nordahl, R. 2014. Establishing the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion. *IEEE Transactions on Visualization and Computer Graphics* 20 (4) pp. 569–578 (<http://dx.doi.org/10.1109/VR.2011.5759456>).
- Nilsson, N.C., Serafin, S., Nordahl, R. 2014. 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, ACM, pp. 187–190.
- Razzaque, S., Swapp, D., Slater, M., Whitton, M.C., Steed, A. 2002. Redirected walking in place. In: Proceedings of the Workshop on Virtual Environments 2002, EGVE '02, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 123–130. (<http://dl.acm.org/citation.cfm?id=509709.509729>).
- Settgast, V., Lancelle, M., Bauer, D., Piff, M., Fellner, D., Computing, V. 2014. Hands-free navigation in immersive environments for the evaluation of the effectiveness of indoor navigation systems. *Journal of Virtual Reality and Broadcasting*, 11 (4).
- Shiratori, T., Hodgins, J.K. 2008. Accelerometer-based user interfaces for the control of a physically simulated character. In: Proceedings of ACM Transactions on Graphics (TOG), Vol. 27, ACM, 2008, p. 123.
- Slater, M., Usoh, M., Steed, A., 1995. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Trans. Comput.-Hum. Interact. (TOCHI)* 2 (3), 201–219.
- Templeman, J., Denbrook, P., Sibert, L., 1999. Virtual locomotion: walking in place through virtual environments. *Presence* 8 (6), 598–617.
- Templeman, J.N., Sibert, L.E., Denbrook, P.S. 2006. Immersive simulation to train urban infantry combat. in: In Virtual Media for Military Applications.
- Terziman, L., Marchal, M., Emily, M., Multon, F., Arnaldi, B., Lécuyer, A. 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, VRST '10, ACM, New York, NY, USA, pp. 27–34. (<http://doi.acm.org/10.1145/1889863.1889867>).
- Torres, N.A., Clark, N., Ranatunga, I., Popa, D. 2012. Implementation of interactive arm playback behaviors of social robot zeno for autism spectrum disorder therapy. In: Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, ACM, p. 21.
- Velloso, E., Schmidt, D., Alexander, J., Gellersen, H., Bulling, A. 2015. The feet in human-computer interaction: A survey of foot-based interaction. *ACM Comput. Surv.* 48 (2) 21:1–21:35. (<http://dx.doi.org/10.1145/2816455>).
- Wendt, J.D., Whitton, M.C., Brooks Jr, F.P. 2010. Gud wip: Gait-understanding-driven walking-in-place. In: Proceedings of Virtual Reality Conference (VR), 2010 IEEE, pp. 51–58.
- Williams, B., Narasimham, G., McNamara, T.P., Carr, T.H., Rieser, J.J., Bodenheimer, B. 2006. Updating orientation in large virtual environments using scaled translational gain. In: Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization, ACM, pp. 21–28.
- Williams, B., Bailey, S., Narasimham, G., Li, M., Bodenheimer, B., 2011. Evaluation of walking in place on a wii balance board to explore a virtual environment. *ACM Trans. Appl. Percep. (TAP)* 8 (3), 19.
- Williams, B., McCaleb, M., Strachan, C., Zheng, Y. 2013. Torso versus gaze direction to navigate a ve by walking in place. In: Proceedings of the ACM Symposium on Applied Perception, ACM, pp. 67–70.
- Wilson, P.T., Nguyen, K., Harris, A., Williams, B. 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, pp. 27–33.
- Zheng, Y., McCaleb, M., Strachan, C., Williams, B. 2012. Exploring a virtual environment by walking in place using the microsoft kinect. In: Proceedings of the ACM Symposium on Applied Perception, SAP'12, ACM, New York, NY, USA, 2012, pp. 131–131. (<http://dx.doi.org/10.1145/2338676.2338713>).
- Zheng, Y., McCaleb, M., Strachan, C., Williams, B. 2012. Exploring a virtual environment by walking in place using the microsoft kinect. In: Proceedings of the ACM Symposium on Applied Perception, ACM, pp. 131–131.
- Zielinski, R.P.M.D.J., Brady, R.B. 2011. Shadow walking: An unencumbered locomotion technique for systems with under-floor projection. *IEEE Virtual Reality (VR) 00* (undefined) pp. 167–170. (<http://dx.doi.org/10.1109/VR.2011.5759456>).