

# Shake-Your-Head: Revisiting Walking-In-Place for Desktop Virtual Reality

Léo Terziman\*  
INSA / INRIA / DGA

Maud Marchal\*  
INSA / INRIA

Mathieu Emily†  
Université Rennes 2 / IRMAR

Franck Multon†  
M2S / Université Rennes2 / INRIA

Bruno Arnaldi\*  
INSA / INRIA

Anatole Lécuyer\*  
INRIA

## Abstract

The Walking-In-Place interaction technique was introduced to navigate infinitely in 3D virtual worlds by walking in place in the real world. The technique has been initially developed for users standing in immersive setups and was built upon sophisticated visual displays and tracking equipments.

In this paper, we propose to revisit the whole pipeline of the Walking-In-Place technique to match a larger set of configurations and apply it notably to the context of desktop Virtual Reality. With our novel "Shake-Your-Head" technique, the user is left with the possibility to sit down, and to use small screens and standard input devices such as a basic webcam for tracking. The locomotion simulation can compute various motions such as turning, jumping and crawling, using as sole input the head movements of the user. We also introduce the use of additional visual feedback based on camera motions to enhance the walking sensations.

An experiment was conducted to compare our technique with classical input devices used for navigating in desktop VR. Interestingly, the results showed that our technique could even allow faster navigations when sitting, after a short learning. Our technique was also perceived as more fun and increasing presence, and was generally more appreciated for VR navigation.

**CR Categories:** H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input Devices and Strategies, Interaction Styles, User-centered Design; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality

**Keywords:** Walking, Walking-In-Place, Camera Motions, Head Tracking, First-Person-Navigation, Desktop Virtual Reality

## 1 Introduction

Navigation is one of the fundamental tasks needed for 3D interaction with Virtual Environments (VE) [Bowman et al. 2005]. The possibility to walk inside the VE is necessary in many applications of Virtual Reality (VR) such as for virtual visits (review of architectural and urban projects) or virtual training (technical procedures, military infantry).

In order to provide VR users with realistic sensations of walking while keeping them in their limited workspace, numerous types of VR interfaces have been proposed so far [Bowman et al. 2005]. Locomotion interfaces, such as unidirectional or omnidirectional

treadmills [Stanney et al. 2002][Iwata and Yoshida 1999], are hardware solutions that enable the user to walk infinitely. But they are often costly, complex and cumbersome. Thus, many software interaction techniques have been developed based on classical 3D input devices such as joysticks or gamepads [Bowman et al. 2005]. But they induce very limited physical movements, and sometimes generate a sensation of "flying" in the virtual scene rather than walking [Lécuyer et al. 2006].

The Walking-In-Place (WIP) technique has been introduced by Slater et al. [Slater et al. 1995] to enable a real physical walking movement and an efficient navigation technique in 3D virtual environments. The user has to consciously walk in place while motions of his body are tracked and analyzed. The tracked Walking-In-Place motion is used as input for the locomotion simulation inside the VE. First implementations of WIP were all based on the processing of head positions using a neural network. More recent models track the positions of the heels or knees of the user to compute the resulting virtual locomotion [Feasel et al. 2008][Wendt et al. 2010]. However, all existing WIP techniques require the user to stand up, and they focus on immersive VR applications based on sophisticated tracking devices and head-mounted-displays or CAVE setups.

In this paper, we propose to revisit the whole pipeline of the WIP technique. Our intention is to extend it to match a larger set of configurations, by notably applying it to desktop setups. As a result, we propose to improve the WIP technique on three main aspects, corresponding to the three main innovations claimed in our paper:

- **Novel user interface capacities.** We introduce (1) the possibility for the user to stand or sit (using classical interaction paradigms of desktop VR with mouse), and (2) the possibility to use screens (with limited field of view) and low-cost tracking (webcam) by using the use of head movements as main input of WIP control laws.
- **Novel locomotion simulation.** We extend the range of possible motions with WIP to: walking, turning, jumping and crawling, using heuristics and control laws based exclusively on users' head movements.
- **Novel visual feedback of walking.** We introduce the possibility to combine our approach with more realistic visual feedback of walking, i.e. camera motions, that are known to improve the sensation of walking [Lécuyer et al. 2006].

## 2 Related Work

Navigating in a virtual environment can be achieved in numerous ways. The most straightforward solution probably consists in walking naturally in the real world, using for instance Head Mounted Display (HMD) together with a tracking system which measures head position for updating viewpoint (or virtual camera position and orientation) in the virtual environment. However, this solution can rarely be used due to the limited physical workspace in the real world, as well as limitations of the tracking devices.

To overcome this problem, numerous "locomotion interfaces" have been designed. The locomotion interfaces keep the user within

\*e-mail: {lterzima|mmarchal|barnaldi|alecuyer}@irisa.fr

†e-mail: {Mathieu.Emily|Franck.Multon}@uhb.fr

Copyright © 2010 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail [permissions@acm.org](mailto:permissions@acm.org).

VRST 2010, Hong Kong, November 22 – 24, 2010.  
© 2010 ACM 978-1-4503-0441-2/10/0010 \$10.00

References	Input/Output Interfaces		Interaction Technique	
	Output: Display	Input: Tracking	Control Law	Available Motions
[Slater et al. 1995]	HMD	Head	Neural network	FW, Turn
[Usoh et al. 1999]	HMD	Head	Neural network	FW, Turn
[Templeman et al. 1999]	HMD	Knees, waist, hands, head	Knee pattern	FW, BW, Diagonal, Sidestep, Turn
[Razzaque et al. 2002]	CAVE	Head	Neural network	FW, Turn
[Yan et al. 2004]	CAVE	Legs, waist, head	Leg speed relation	FW, Turn
[Feasel et al. 2008]	HMD	Shins	Filtered heels speed	FW (various speeds), Turn
[Wendt et al. 2010]	HMD	Shins	Biomechanical state machine	FW (various speeds), Turn

**Table 1:** Overview of existing Walking-In-Place techniques.

the limits of the physical workspace while allowing to navigate infinitely in the virtual world. A classical hardware approach consists in using a treadmill which can be either unidirectional (such as the Sarcos Treadport [Stanney et al. 2002]) or omni-directional (such as the Torus Treadmill [Iwata and Yoshida 1999]). Other locomotion interfaces can be based on feet-platforms [Iwata et al. 2001], walking inside a rotating sphere [Fernandes et al. 2003], or a set of moving tiles [Iwata et al. 2005]. However, the current locomotion interfaces are often too expensive, cumbersome, or complex for being integrated easily in both VR or desktop applications.

Thus, numerous interaction techniques have been developed to provide software-based navigation capabilities in VR. For instance, the World In Miniature (WIM) technique [Pausch et al. 1995] is based on a miniature graphical representation of the VE which can be used to select destinations in the VE. The virtual camera “flies” automatically to the new position selected in the WIM. The Magic Barrier Tape [Cirio et al. 2009] is another interaction metaphor which enables to navigate infinitely when immersed in a virtual scene, when located in a restricted walking workspace. The physical boundaries of the workspace are displayed in the virtual environment using a barrier tape that must not be crossed by the user but that can be moved in the VE with a joystick.

Besides, classical input devices such as mice, keyboards and joysticks can also be used within interaction metaphors to navigate from one virtual place to another [Bowman et al. 2005]. An efficient solution for navigating in 3D VE with a desktop configuration is the interaction paradigm proposed in most First-Person Shooter video games (FPS). It relies on mouse control (for orienting the viewpoint and the direction of motion), and keyboard keys (for forward, backward and lateral motions). However, such interaction techniques, relying exclusively on manipulated input devices, do not provide any proprioceptive or vestibular feedback of walking, and lead to a poor sensation of walking in the VE.

An efficient solution to enhance the sensation of walking in virtual worlds consists in playing with visual feedback and adding artificial camera motions [Lécuyer et al. 2006]. Such camera motions can simulate the visual flow corresponding to natural oscillations of the body and head during the walk. This approach also simulates the ocular reflexes present when walking [Hillaire et al. 2008]. Camera motions were shown to improve sensation of walking and perception of traveled distances in VE [Terziman et al. 2009].

Interestingly, novel commercial systems for optical tracking of body movements have appeared such as the EyeToy for Playstation 2 or, more recently, the Microsoft Kinect. Both systems are focused on interaction with multimedia content and video games but, for the moment, these systems have not been used for direct walking in 3D virtual environments such as when “walking in place”.

Walking-In-Place (WIP) is an interaction paradigm designed for both an improved sensation of walking and an efficient navigation in virtual environments [Slater et al. 1995]. The WIP technique

consists in consciously walking in place in the real world. Previous papers on WIP are presented in Table 1. The first implementation proposed for WIP was based on a HMD. Head tracking was used to predict and detect the steps of the user using a neural network [Slater et al. 1995]. When user’s steps were detected, the viewpoint automatically moved forward in the VE. The advance direction was simply related to the head’s direction. In the next implementations of WIP, the processing of head movements with a neural network was abandoned. The next systems could track the motions of the user’s knees or heels. Then, using pattern recognition [Templeman et al. 1999], frequency analysis [Feasel et al. 2008] or biomechanically inspired state automaton [Wendt et al. 2010] the user’s steps could be detected and translated into a variable advance speed depending on the steps frequency.

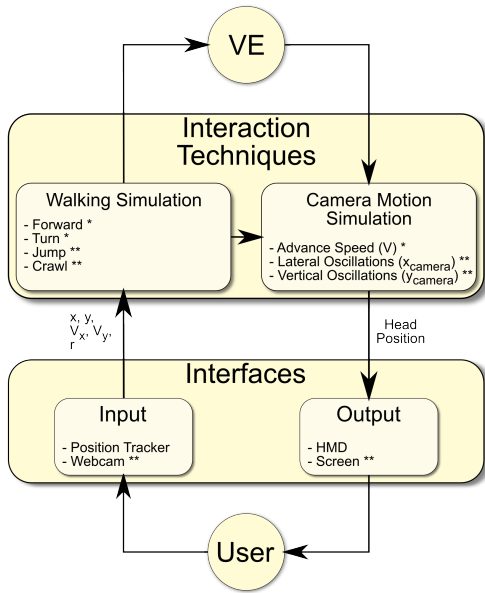
When considering Table 1, it seems that all the WIP techniques described in the literature so far are meant for highly immersive virtual environments and use either HMD or CAVE for visual display [Razzaque et al. 2002]. These setups use position/orientation tracking of body parts to detect footsteps and compute an advance speed [Feasel et al. 2008]. The few systems based on head movements are all using neural network as a processing scheme [Slater et al. 1995]. Turning in the virtual world is often achieved using the orientation of one of the tracked points (e.g., head, waist, chest) [Templeman et al. 1999].

Unfortunately, very few evaluations of WIP have been carried out up to now to compare it with other existing navigation techniques. Moreover, these rare studies have mostly used subjective questionnaires, excepted in [Whitton et al. 2005]. Considering the data available in the literature, it seems that WIP can increase presence compared to classical joystick-based interaction [Razzaque et al. 2002][Usoh et al. 1999]. WIP is also perceived as a more natural technique [Yan et al. 2004]. However, WIP might also result in an increased cybersickness [Usoh et al. 1999]. WIP seems also less efficient in term of usability than a real walk [Feasel et al. 2008].

### 3 Shake-Your-Head: Revisiting WIP for Desktop VR

We propose to revisit the whole pipeline of the Walking-In-Place technique to match a larger set of configurations and apply it to the context of desktop Virtual Reality. Our approach is schematized in Figure 1. The Figure highlights the main differences between our approach and the classical and existing WIP techniques.

With the Shake-Your-Head technique, the user can be standing or sitting (such as in traditional video games or desktop VR configurations). The user interacts with the system by means of head movements. These head movements can be captured using different tracking interfaces, but we insist on the use of low-cost optical tracking with standard webcams. The locomotion simulation proposes not only the computation of a virtual walking motion but also



**Figure 1:** Overview of our novel approach for Walking-In-Place (one star stresses improvement of existing component, two stars stress additional components).

turning, jumping, and crawling possibilities. As a result, the user can perceive the locomotion in the virtual world by means of integrated virtual camera motions on the three axes of motion, to further enhance the sensation of walking.

In the following section, we will successively describe the different parts of our approach, namely: (1) the 3D user interface input/output, (2) the interaction techniques developed for the computation of the virtual locomotion, and (3) the visual feedback relying on camera motions.

### 3.1 Input/Output Interfaces

Our method proposes new features in terms of interfaces in order to extend the set of configurations where WIP can be applied, especially for Desktop VR. Thus, we propose to incorporate new devices for both input and output user interfaces in the VE.

#### 3.1.1 Input: Tracking Based on Head Motions

The input interface of our method is only based on head motions. In our implementation, the input interface is reduced to a webcam, allowing the use of our method for Desktop configuration without any additional peripheral. However, our method can also be implemented with other classical VR tracking systems.

**Use of Head Movements.** The main concept of our method is to exploit the head oscillations as a transposition of the one observed during natural walking. While walking, the head of the user oscillates along the lateral, vertical and forward axes [Lécuyer et al. 2006]. The oscillations are strongly correlated to gait events and foot steps. Moreover, these oscillations also occur while walking in place and can be measured.

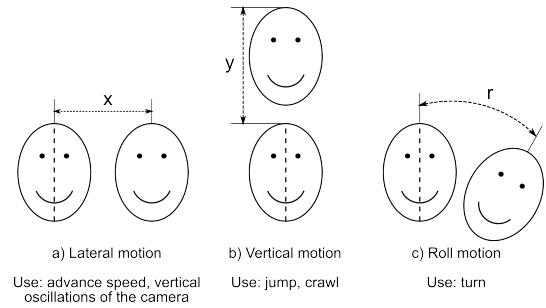
The head motions are classically retrieved in the existing WIP techniques thanks to the use of regular position trackers [Slater et al. 1995]. More generally, any tracking device can be used, as long as its accuracy is within the range of 1cm. Moreover, the acquisition

process of the required position does not have any influence, as long as the real time constraint is maintained. In our method, we propose the use of a video camera system to handle the tracking of the user head. Thus, our interaction technique can be deployed on a large scale at low cost for training purpose or video games for example.

**Extracted Data.** In our method, we propose to use 3 Degrees of Freedom (DOF) that can be easily accessed in the image frame provided by the webcam:

- The lateral position  $x$  (and the computed speed  $V_x$ );
- The vertical position  $y$  (and the computed speed  $V_y$ );
- The rotation of the head in the frontal plan  $r$ .

These three head motions are illustrated in Figure 2.



**Figure 2:** Extracted head motions: (a) lateral motion, (b) vertical motion, (c) roll motion.

**Implementation.** The use of the webcam to track the 3D position of the user head without using markers requires the implementation of real time constraints for the algorithms, i.e. more than 25 frames per seconds. Our implementation is based on the Camshift (Continuously Adaptive Mean Shift) algorithm [Bradski 1998] implemented in the OpenCV library. This algorithm is based on color tracking and is well-suited for real-time tracking of features of a given color, such as the face of the user.

While the user is standing in front of the webcam, our algorithm recognizes him as an ellipsoid. The position of the head  $(x, y)$  can be deduced from the center of the ellipse and the orientation angle of the head  $r$  is given by the angle with the vertical of the ellipse. However,  $x$  and  $y$  depend on the resolution of the used webcam. Thus, we compute a normalized position  $(x_n, y_n)$  on  $[-1; 1]$  on both axes. From the normalized positions of the head  $(x_n, y_n)$ , we compute the instantaneous speeds  $V_x$  and  $V_y$ , and we use Kalman filters on all values to reduce the noise produced by the algorithm.

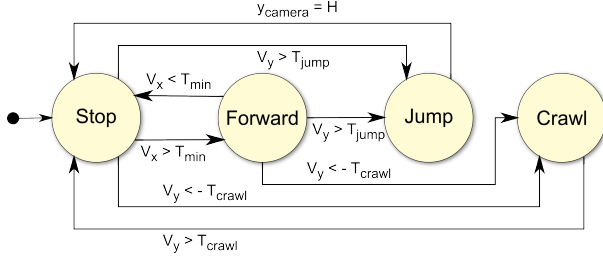
#### 3.1.2 Output: Immersive and Desktop Visual Displays

Our method can be used with both immersive and regular screens as output interface with the VE. A requirement of desktop VR applications was to propose a technique that is usable with a limited Field of View (FOV). In regular WIP techniques, the provided FOV is always  $360^\circ$ , except in [Razaque et al. 2002] where they used the redirected walking technique to simulate a  $360^\circ$  FOV in a 4-walls CAVE. In this paper, our technique is evaluated with both LCD laptop screens to fulfill the Desktop VR pre-requisite or video projected output. Our method can also be used with the classical output used with WIP, i.e. with HMDs or CAVEs.

## 3.2 Walking Simulation

### 3.2.1 Walking States

The main goal of our interaction technique is to translate the inputs of the user, i.e. head motions, into virtual motions in the VE. The user should be able to perform various motions while navigating in the VE. We implemented different locomotion states: walking, turning, jumping and crawling. To manage these different states, we added a state automaton to our algorithm (Figure 3). The state transitions are governed by the user head motions and the main inputs are the lateral velocity  $V_x$  and the vertical velocity  $V_y$ .



**Figure 3:** Walking state automaton: the main inputs are the velocities  $V_x$  and  $V_y$ .  $V_x$  allows the transition from the Stop state to the Forward motion;  $V_y$  allows the transition to the Jump or Crawl states. The Turn motion is not represented as it can be activated in any state, depending on the head orientation.

### 3.2.2 Forward State

The forward movements in the VE are governed by the lateral oscillations as main input. Our technique is designed to emphasize the idea of walking with a varying advance speed depending on the user head motions. The advance velocity  $V_a$  oscillates regularly, accordingly to the lateral head motions. One head oscillation period corresponds to one step. The footstep events are simulated by a null advance speed and correspond to a modification of the lateral velocity sign of the user's head. Thus, when the user's head reaches the maximal amplitude of the oscillations, the oscillating speed is null, as well as the advance speed, simulating a foot step.

For more realistic movements, we introduced two thresholds  $T_{min}$  and  $T_{max}$ .  $T_{min}$  allows to stop the forward movements when the lateral head motions are too small.  $T_{max}$  allows to avoid unrealistic high speed walks. The advance velocity  $V_a$  is computed in two steps in order to test these thresholds. The equations of  $V_a$  are:

$$\begin{aligned} V_{n1} &= \frac{\min(\text{abs}(V_x), T_{max})}{T_{max}} \\ V_{n2} &= \begin{cases} 0 & \text{if } V_{n1} < T_{min} \\ V_{n1} & \text{otherwise} \end{cases} \\ V_a &= V_{n2} * V_{max} \end{aligned} \quad (1)$$

Finally, the advance speed  $V$  of the camera inside the VE is adapted depending on the current locomotion state and is given by:

$$V = \begin{cases} V_a & \text{if state} = \text{walk} \\ 0.4 * V_a & \text{if state} = \text{crawl} \\ V_{max} & \text{if state} = \text{jump} \end{cases} \quad (2)$$

We chose normalized thresholds with the following values:  $T_{min} = 0.05$  and  $T_{max} = 0.5$ . We also set  $V_{max}$  to 3.5 meters per second, corresponding to the maximal speed that can be achieved.

### 3.2.3 Jump and Crawl States

Comparing to existing WIP techniques, we chose to add 2 new states to the navigation possibilities in the VE: Jump and Crawl motions. The jump and crawl states are governed by the vertical oscillations of the user's head. If the vertical velocity exceeds normalized thresholds  $T_{jump}$  in upward direction and  $T_{crawl}$  in downward direction, the user can jump and crawl respectively in the VE. In practice, it means that the user will need to slightly jump or bend forward if he is seated, or jump or crouch down if he is standing. The user has to stand-up to stop crawling.

When a jump is detected, the vertical position of the camera is set to follow a classical parabolic trajectory defined by:

$$y_{camera} = \frac{1}{2} * g * t^2 + V * t \quad (3)$$

with  $g$  the gravity acceleration and  $t$  the time. The jumping state is left automatically while landing, i.e. when the camera reaches again its normal height  $H$  (known as the reference state when the algorithm starts). After preliminary testings, we set  $T_{jump} = 0.3$  and  $T_{crawl} = 0.4$ . While the crawling state is activated, the vertical position of the virtual camera  $y_{camera}$  is lowered by 1 m.

### 3.2.4 Turn State

In parallel to Forward, Jump and Crawl states, the user has the possibility to turn inside the VE in order to modify his navigation direction. During a turn in a normal walk, the human body leans slightly in direction of the center of the turn to compensate the centrifugal force [Courtine and Schieppati 2003]. This phenomenon is often reproduced by video games players which can tend to lean in the direction of the turn even if it does not have any influence on their in-game trajectory. Thus, we choose to use this property to implement turns in our system as a control law based on the head orientation on the roll axis. To turn in the VE, the user has to lean his head in the left or right side respectively to turn left or right in the VE. The rotation speed  $V_r$  of the virtual camera is given by:

$$V_r = \begin{cases} V_{r_{max}} & \text{if } r > r_{max} \\ -V_{r_{max}} & \text{if } r < -r_{max} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where  $r_{max}$  is the minimum angle of head inclination to start the rotation and  $V_{r_{max}}$  is the maximal angular speed of the rotations. In our experiment, we set  $r_{max} = 15^\circ$  and  $V_{r_{max}} = 45^\circ/s$ .

## 3.3 Visual Feedback Based on Camera Motions

To further emphasize the perception of walking in the VE, we extended the visual rendering of the WIP using camera motions driven by the user's head oscillations. There are existing models in the literature that make the virtual camera oscillating along the three axes ([Lécuyer et al. 2006] for example). However, the oscillations are totally independent from the user interactions.

We introduce a new model of camera motions adapted to the user's head motions. The camera oscillations along the different axes must follow the user in real time to maintain the coherency of the system. Thus we have implemented a novel visual feedback with camera motions along the vertical, lateral and advance axes.

### 3.3.1 Advance Oscillations

The advance speed  $V$  of the view point already oscillates. The camera motions are indeed intrinsically linked to the advance velocity

of the control law presented in paragraph 3.2.2. As a result, extra camera motion is not necessary along this axis and the advance camera velocity corresponds exactly to  $V$ .

### 3.3.2 Lateral Oscillations

In order to move in the VE, the user has to make his head oscillating from left to right. Thus, as the user moves in front of the screen, his view point of the scene is modified to follow the head oscillations.

The lateral oscillations of the camera are computed as a function of the user's position. If  $d$  is the distance of the user to the screen and  $\alpha$  and  $\beta$  the opening angles of the webcam, the real world position of the user in front of the screen depends on the normalized coordinates  $x_n$  and  $y_n$ . The real world position of the user's head is given by the following coordinates:

$$\begin{cases} x_{real} &= x_n * d * \tan(\alpha/2) \\ y_{real} &= y_n * d * \tan(\beta/2) \end{cases} \quad (5)$$

Finally, the virtual camera is moved along the lateral axis by a distance  $x_{camera}$  equals to:  $x_{camera} = A_x * x_{real}$ . We set the scale factor  $A_x = 1$  to match the user's head displacement and thus generate the illusion that the screen is a window through which the user can observe directly the VE. However, other values can be used to amplify the camera motions for example. The webcam used during the experiment was such as  $\alpha = 60^\circ$  and  $\beta = 45^\circ$ .

### 3.3.3 Vertical Oscillations

The vertical oscillations of the camera can not be computed with the same algorithm as for lateral oscillations. In a desktop VR context the user can be seated and not be able to produce high vertical oscillations.

In our method, we propose to generate pseudo-sinusoidal vertical camera oscillations based on the current phase of the virtual gait cycle. Similarly to advance speed control law, the vertical amplitude  $y_{camera}$  of the camera oscillations is given by:

$$y_{camera} = V_{n2} * y_{camera}^{max} \quad (6)$$

where  $y_{camera}^{max}$  is the amplitude of the vertical oscillations for velocities greater or equal to the  $T_{max}$  threshold. For smaller speeds, the amplitude of the oscillations is proportional to this maximum, thus increasing the perception of the variations in advance speeds. Using the same factor between the camera motions and the advance velocity forces the synchronization, resulting in a smooth final visual rendering. In our implementation, we set  $y_{camera}^{max} = 15cm$ .

## 3.4 Discussion

To summarize, our approach is composed of (1) an input interface based on the sole user's head movements, (2) a locomotion simulation in the VE composed of various possibilities such as jumping, crawling, turning, and (3) a visual feedback of walking relying on oscillating camera motions. The head motions are tracked along 3 Degrees of Freedom (DOF): lateral, vertical and roll axis (Figure 2). These different physical motions are transposed in virtual movements thanks to a locomotion automaton (Figure 3). We then added oscillating camera motions (Equations 2, 5 and 6) to the visual feedback to enhance the walking sensation. The different control laws were parametrized after preliminary testings. But of course some parameters can be modified in order to amplify/decrease some effects during the locomotion simulation. Besides, other movement possibilities could also be envisaged and added to our automaton such as running state or backward movement.

## 4 Evaluation

The evaluation of the proposed technique was performed using a comparison with classical techniques in Desktop VR. We chose keyboard and joystick peripherals for seated and standing positions respectively as they are often used in Desktop VR context. The experiments were conducted using 3D VE displayed on a screen and we investigated the effectiveness of our technique to travel complex paths composed of different gates placed in the VE.

In this paper, we choose to not compare our technique to existing WIP techniques, and instead used more common interfaces that followed our low cost requirement. We conducted the evaluation of the proposed technique in both immersive Standing Up (SU) position and Desktop Sitting Down (SD) position. The keyboard and joystick were chosen respectively as the control conditions in the SD position and the SU position. Both keyboard and joystick peripherals are referred as Control techniques (Ctrl) in the following paragraphs. Our technique is referred with the "WIP" suffix.

### 4.1 Method

#### 4.1.1 Virtual Environment

The evaluation was performed within a 3D virtual environment without any contextual cues. The only landmarks were the gates that the user had to navigate through. A fog effect was added to mask the distant gates, allowing to perceive only the 2 or 3 closest gates. A texture on the ground provided useful visual flow information during the navigation. The scene was normally illuminated, and no shadows were drawn as illustrated in Figure 4.



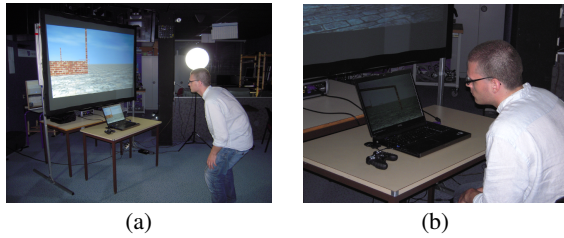
**Figure 4:** Virtual environment with different gates to represent the path that the user has to navigate through.

#### 4.1.2 Population

Twelve participants (10 males and 2 females) aged from 22 to 35 (mean=25.4, standard deviation=3.35) performed the experiment. Two participants were left-handed and none of them had known perception disorders. All participants were used to VEs but were naïve with respect to the proposed technique, as well as to the experimental setup and purpose of the experiment.

#### 4.1.3 Experimental Conditions

The experiment was carried out in a room with controlled luminosity (using two projectors). There were two visual conditions corresponding to the two different configurations (SD and SU). The two configurations are illustrated in Figure 5. For the SU configuration, participants were at a distance of 1.5 m in front of a 1.72 m large and 1.24 m height back-projected screen (physical field of view of  $60^\circ$  horizontally and  $45^\circ$  vertically). During the SD configuration, they sat in front of a 17 inches widescreen laptop screen (physical field of view of  $30^\circ$  horizontally and  $18.5^\circ$  vertically). In both cases, the resulting image had a resolution of  $1600 \times 1200$  pixels. We used monoscopic rendering, with a frame rate of 60 Hz. The projector used was a DepthQ Stereoscopic.



**Figure 5:** Description of the experimental setup for (a) the Stand-Up (SU) configuration, and (b) the Sit-Down (SD) configuration.

#### 4.1.4 Experimental Apparatus

In our experiment, the goal was to compare our technique to classical interface devices (keyboard and joystick). During the experiment, the participants had to navigate in the VE through 3 different paths composed of 8 gates each. The 2 first paths, called Normal, were composed of 3 meters by 3 meters wide gates regularly disposed in order to form a slalom. The third path, called Steeple, was composed of 1.5 meters high tunnels and 80 cm high fences (4 of each). For this path, the users had to jump and crawl to pass the gates, while the 2 other paths only required a simple navigation.

The classical interface devices were implemented as follow:

- Keyboard: Forward and turn motions were triggered using the forward, left and right arrow keys. The jumping was triggered by the left Shift key and the crawling by the left Control key.
- Joystick: Forward and turn motions were triggered using the left joystick of a gamepad. Jumping and crawling were triggered with two of the right buttons.

To provide a fair comparison between WIP and Ctrl conditions, the oscillating camera motions were also implemented in the two Ctrl conditions. To do so, we used the best implementation described in [Lécuyer et al. 2006]. Moreover, the Ctrl condition advance speed was set to match the WIP condition average advance speed, which had been measured in a preliminary phase.

The participants had to press the "Enter" key at the beginning of each block of conditions to launch the experiment. At any time during the experiment they had the possibility to make a break by pressing the "Space" key. After each trial, the participants were automatically teleported to the beginning of the next trial. A black screen displayed 2 s notified the beginning of the new trial.

#### 4.1.5 Experimental Plan

The participants were exposed to 4 blocks of 18 trials each: one block for each of the possible combinations of the experimental conditions. The combinations were the following: (1) Sitting Down, using the keyboard (SD-Ctrl); (2) Sitting Down, using the our technique (SD-WIP); (3) Standing Up, using the joystick (SU-Ctrl); (4) Standing Up, using the proposed technique (SU-WIP).

The participants were split in 4 groups equally composed of 3 people each. Two groups started with the SD configuration and the 2 other with the SU configuration. We counter-balanced the conditions between them, meaning that the group starting with Ctrl condition in the SD configuration starts with WIP condition in the SU configuration. For each of the 18 trials of a block, the participants had to navigate through the 3 paths. The 12 first trials were composed only of the normal paths (6 for each, in a random order), and the last 6 trials corresponded to the steeple path.

## 4.2 Collected Data

The WIP techniques are generally evaluated only with subjective questionnaires to evaluate presence or cybersickness. In our evaluation, we added new criteria based on the task performed by the participants. Thus, we measured for each participant the task completion time for each trial and the percentage of success for the different gates. A subjective questionnaire was also proposed.

## 4.3 Results

### 4.3.1 Task Completion Time

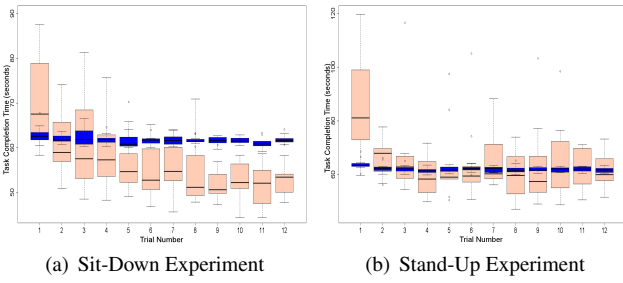
For each participant, the task completion time of each trial was measured for the different experimental conditions. An exploratory analysis was first performed. A Principle Component Analysis revealed the presence of one spurious individual who has been taken out from the analysis. A specific analysis was developed to study the learning effect of the two conditions (Joystick/Keyboard and WIP techniques). A linear model where all conditions are mixed was fitted to explain the relation between the task completion times and the trial number. It revealed that the slope of the linear regression was significantly lower than zero ( $p$ -value  $< 0.000001$ ), reflecting a significant decrease in the task completion time as the number of trials increases. The same analysis, where the first trial was removed, showed that the slope was not significantly different from zero anymore. In the following paragraph, the first trial was removed from the analysis as it corresponds to a learning effect.

A two-way ANOVA was performed on the 2 different conditions (Joystick/Keyboard and WIP techniques) and the 2 different positions (Sit-Down and Stand-Up). A post-hoc analysis using Tukey's procedure was then performed. The ANOVA was achieved separately for the two different types of paths (normal/steeple).

Concerning the normal path, the two-way ANOVA accounting for the conditions and the positions revealed a significant dependency between the position and the task completion time ( $F(1, 11) = 31.9981$ ,  $p < 0.0001$ ) and between the condition and the task completion time ( $F(1, 11) = 6.0449$ ,  $p = 0.0143$ ). Interaction between condition and position was also considered as a significant factor to discriminate task completion time ( $F(1, 11) = 27.1921$ ,  $p < 0.0001$ ). Post-hoc analysis showed that the task completion time in the SD-WIP configuration ( $M = 55.67s$ ) was significantly lower than in the SD-Ctrl configuration ( $M = 61.72s$ ), adjusted  $p$ -value  $< 0.0001$ , in the SU-Ctrl configuration ( $M = 62.06s$ ), adjusted  $p$ -value  $< 0.0001$ , and in the SU-WIP configuration ( $M = 64.23s$ ), adjusted  $p$ -value  $< 0.0001$ . The other pairs of effects did not give any significant adjusted  $p$ -values.

The results concerning the different conditions are represented in Figure 6 for Sit-Down and Stand-Up experiments respectively, for the normal path only. The results are ordered in function of the trials. The first trial was kept to illustrate the learning effect.

For the steeple path, the pre-analysis suggested the presence of a spurious individual and the existence of one learning trial. The two-way ANOVA accounting for the conditions and positions revealed a significant dependency between the position and the task completion time ( $F(1, 11) = 8.5665$ ,  $p < 0.005$ ), and the condition and the task completion time ( $F(1, 11) = 11.8925$ ,  $p < 0.001$ ). Interaction between condition and position was also considered as a significant factor to discriminate task completion time ( $F(1, 11) = 8.7647$ ,  $p < 0.005$ ). Post-hoc analysis showed that the task completion time in the SU-WIP configuration ( $M = 86.07s$ ) was significantly higher than in the SD-Ctrl configuration ( $M = 74.78s$ ), adjusted  $p$ -value  $< 0.0001$ , in the SU-Ctrl configuration ( $M =$



**Figure 6:** Task completion time for the two different techniques on normal paths for (a) SD experiments and (b) SU experiments. The blue and red light colors correspond to the Ctrl and WIP conditions respectively. Each boxplot is delimited by the quartile (25% quantile and 75% quantile) of the distribution of the condition over the individuals. The median is also represented for each condition.

74.72s), adjusted p-values  $< 0.0001$  and in the SD-WIP configuration ( $M = 75.64s$ ), adjusted p-values  $< 0.0001$ . The other pairs of effects did not give any significant adjusted p-values.

### 4.3.2 Accuracy

For each participant and for each trial, the percentage of errors for the different paths was measured. The two types of path (Normal and Steeple) are separated. The resulting percentages are for the normal path: 0.61% of error for SD-WIP configuration, 0% of error for SD-Ctrl configuration, 1.39% of error for SU-WIP configuration, 0.09% of error for SU-Ctrl configuration.

The resulting percentages are for the steeple path: 20.49% of error for SD-WIP configuration, 7.81% of error for SD-Ctrl configuration, 27.28% of error for SU-WIP configuration, 13.19% of error for SU-Ctrl configuration. We found a significant effect between Ctrl and WIP techniques for the steeple path.

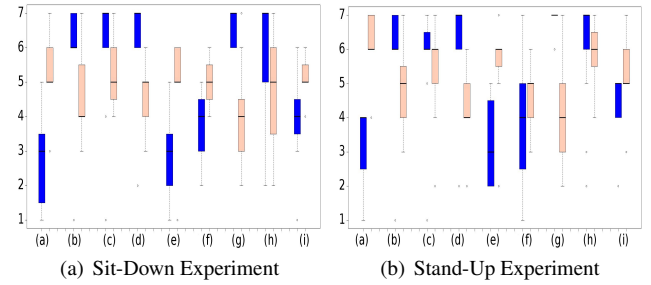
### 4.3.3 Subjective Questionnaire

After Sit-Down and Stand-Up configurations, a preference questionnaire was proposed in which participants had to grade from 1 (low appreciation) to 7 (high appreciation) the four different conditions (SD-Ctrl, SD-WIP, SU-Ctrl, SU-WIP) according to 9 subjective criteria: (a) Fun, (b) Easiness of Use, (c) Intuitive, (d) Accuracy, (e) Presence, (f) Walking realism, (g) Fatigue, (h) Cybersickness and (i) Global appreciation. Figure 7 shows the results concerning the grades (Likert-scale) obtained by the two different techniques for each of the subjective criteria, for the two experimental conditions (SD and SU). The grade 7 for Fatigue and Cybersickness respectively means that the technique does not induce any fatigue and does not imply any cybersickness feeling.

Concerning SD configuration, no significant effect was found for the following criteria: Intuitive ( $p = 0.052$ ) and Cybersickness ( $p = 0.12$ ). Concerning SU configuration, no significant effect was found for the following criteria: Intuitive ( $p = 0.3$ ), Walking realism ( $p = 0.19$ ) and Cybersickness ( $p = 0.21$ ). We found a significant effect for all other criteria. In particular, our technique was better ranked for Fun, Presence and Global Appreciation, for both configurations.

## 5 Discussion

Our results suggest that our technique can allow efficient navigation even compared with standard and well-known input devices such



**Figure 7:** Results for subjective rating for the two different techniques for (a) SD experiments and (b) SU experiments. The blue and red light colors correspond to the Ctrl and WIP conditions respectively. The subjective criteria are (a) Fun, (b) Easiness of Use, (c) Intuitive, (d) Accuracy, (e) Presence, (f) Walking realism, (g) Fatigue, (h) Cybersickness and (i) Global appreciation. Each boxplot is delimited by the quartile (25% quantile and 75% quantile) of the distribution of the condition over the individuals. The median is also represented for each condition.

as keyboards and gamepads. The participants could sometimes go even faster with WIP, without any strong loss in precision. The WIP-based interaction seems also fast to learn, after only a couple of trials. The technique is well appreciated and perceived as more immersive and more fun than classical configurations.

The quick learning of our technique could be explained by the fact that interfaces based on webcam are generally intuitive and simple to learn [Polaine 2005]. After the learning phase, WIP tends to become faster than the keyboard in sitting condition. One explanation could be that with our technique (but also with the joystick in the standing condition) we could observe that participants tended to turn without stopping their advance motion. On the contrary, with the keyboard condition, participants tended to walk and turn sequentially, which might have globally increased the task completion time. Another explanation could be that, in our implementation of WIP, the advance speed is influenced by the speed of lateral movements. The seated position allows the user to make faster oscillations than the standing position, and thus to accelerate the walking motion by making fast lateral oscillations. Interestingly, these faster motions did not impair the precision of users.

The longer task completion time observed for WIP in the steeple paths (involving jumping and crawling motions) could be due to unexpected behaviors which induced incorrect transitions in our locomotion automaton. Indeed, some participants acted as if they were "anticipating" the jumps and bent down prior jumping. We could easily fix this problem in the future by using additional conditions in our automaton based on both speed and position.

Results from the questionnaire are very consistent with previous subjective evaluations of WIP [Usoh et al. 1999]. In our study, the WIP is more appreciated, and is perceived as more fun, and improving presence. As expected, more standard techniques (i.e. joystick and keyboard) are found easier to use, more precise, and less tiring (as they induce less physical movements). Interestingly, impression of cybersickness is not increased by WIP. This could be due to our desktop (and thus less immersive) configurations. Last, realism of walking in the VE was significantly improved only in the sitting condition. The perception of walking with WIP is actually quite complex, as participants wrote: "we have the impression to be a video game character", "the motions are exaggerated", or "we really have the sensation of walking, and not running". In the standing condition, some participants found that the physical motions were closer to "skiing" or "skating", as they noticed that they did not lift

their feet from the ground but only oscillated their body. For these people we could further stress in the future that our implementation of WIP still works very well when lifting the feet and walking in place, as the oscillations of the head can be captured the same way in both situations (lifting the feet or not).

Taken together, our results suggest that our Shake-Your-Head technique could be used in a wide range of applications, when navigating in a 3D world, in sitting or standing configurations. It seems to be both a low-cost and an efficient paradigm that can match a lot of walking motions. It could thus be used for training in VR with more physical engagement (military infantry, vocational procedures), or more realistic virtual visits such as for project review in architecture or urban planning.

## 6 Conclusion

This paper proposed to revisit the whole pipeline of the Walking-In-Place technique to match a larger set of configurations and apply it notably to the context of Desktop Virtual Reality. We have introduced the Shake-Your-Head technique using as sole input the head movements of the user. It can be used in a desktop configuration with the possibility for the user to sit down and to navigate in the VE through small screens and standard input devices such as a basic webcam for tracking. We implemented various motions such as turning, jumping and crawling in the locomotion simulation. We also introduced the use of additional visual feedback based on camera motions to enhance the walking sensation.

We conducted an experiment in order to evaluate our technique compared to standard techniques such as keyboard and joystick. In this experiment, participants had to walk through a series of gates forming a slalom path. The evaluation was performed both in an immersive and desktop configurations. We notably found that WIP technique only requires a small learning time to allow faster navigation in seated position compared to the keyboard. Moreover, our technique was more appreciated and considered as more fun and inducing more presence than the other classical techniques.

Future work will focus on improving the underlying locomotion simulation. We could add new features such as running or backward movements in the VE. We could also improve the control laws with a more biomechanically-inspired description. Second, we also plan to conduct more evaluations, especially for further studying the perceptive aspects and measuring the influence of WIP on estimating spatial properties of VE.

## References

BOWMAN, D. A., KRUIJFF, E., LAVIOLA, J. J., AND POUPLYREV, I. 2005. *3D User Interfaces: Theory and Practice*. Addison-Wesley Boston (MA).

BRADSKI, G. R. 1998. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal* 2, 2, 12–21.

CIRIO, G., MARCHAL, M., REGIA-CORTE, T., AND LÉCUYER, A. 2009. The magic barrier tape: a novel metaphor for infinite navigation in virtual worlds with a restricted walking workspace. In *Proceedings of the ACM Symposium on VRST*, 155–162.

COURTINE, G., AND SCHIEPPATI, M. 2003. Human walking along a curved path. i. body trajectory, segment orientation and the effect of vision. *Eu. J. of Neuroscience* 18, 1, 177–190.

FEASEL, J., WHITTON, M. C., AND WENDT, J. 2008. Llcwip: Low-latency, continuous-motion walking-in-place. In *IEEE Symposium on 3D User Interfaces*, 97–104.

FERNANDES, K., RAJA, V., AND EYRE, J. 2003. Cybersphere: the fully immersive spherical projection system. *Communications of the ACM* 46, 9, 146.

HILLAIRE, S., LÉCUYER, A., COZOT, R., AND CASIEZ, G. 2008. Using an eye-tracking system to improve camera motions and depth-of-field blur effects in virtual environments. In *Proceedings of IEEE Virtual Reality*, 47–50.

IWATA, H., AND YOSHIDA, Y. 1999. Path reproduction tests using a torus treadmill. *Presence: Teleoperators and Virt. Env.* 8, 6, 587–597.

IWATA, H., YANO, H., AND NAKAIZUMI, F. 2001. Gait master: A versatile locomotion interface for uneven virtual terrain. In *Proceedings of IEEE Virtual Reality*, 131–137.

IWATA, H., YANO, H., FUKUSHIMA, H., AND NOMA, H. 2005. Circulafloor: A locomotion interface using circulation of movable tiles. In *Proceedings of IEEE Virtual Reality*, 223–230.

LÉCUYER, A., BURKHARDT, J.-M., HENAFF, J.-M., AND DONKIAN, S. 2006. Camera motions improve the sensation of walking in virtual environments. In *Proceedings of IEEE Virtual Reality*, 11–18.

PAUSCH, R., BURNETTE, T., BROCKWAY, D., AND WEIBLEN, M. E. 1995. Navigation and locomotion in virtual worlds via flight into hand-held miniatures. In *Proceedings of the SIGGRAPH '95*, 399–400.

POLAINE, A. 2005. The flow principle in interactivity. In *Proceedings of the 2nd Au. conf. on Interactive Entertainment*, 158.

RAZZAQUE, S., SWAPP, D., SLATER, M., WHITTON, M. C., AND STEED, A. 2002. Redirected walking in place. In *Proceedings of the EGVE '02 workshop on Virtual environments*, 123–130.

SLATER, M., USOH, M., AND STEED, A. 1995. Taking steps: The influence of a walking technique on presence in virtual reality. *ACM Trans. on Computer-Human Interaction* 2, 3, 201–219.

STANNEY, K. M., ET AL. 2002. *Handbook of Virtual Environments: Design, Implementation, and Applications*. Lawrence Erlbaum Associates.

TEMPLEMAN, J. N., DENBROOK, P. S., AND SIBERT, L. E. 1999. Virtual locomotion: Walking in place through virtual environments. *Presence: Teleoperators and Virt. Env.* 8, 6, 598–617.

TERZIMAN, L., LÉCUYER, A., HILLAIRE, S., AND WIENER, J. M. 2009. Can camera motions improve the perception of traveled distance in virtual environments? In *Proceedings of IEEE Virtual Reality*, 131–134.

USOH, M., ARTHUR, K., WHITTON, M. C., BASTOS, R., STEED, A., SLATER, M., AND BROOKS, JR., F. P. 1999. Walking > walking-in-place > flying, in virtual environments. In *Proceedings of SIGGRAPH '99*, 359–364.

WENDT, J., WHITTON, M. C., AND BROOKS, F. 2010. Gud wip: Gait-understanding-driven walking-in-place. In *Proceedings of IEEE Virtual Reality*, 51–58.

WHITTON, M. C., COHN, J., FEASEL, J., ZIMMONS, P., RAZZAQUE, S., POULTON, S., MCLEOD, B., AND JR, F. B. 2005. Comparing VE locomotion interfaces. In *Proceedings of IEEE Virtual Reality*, 123–130.

YAN, L., ALLISON, R., AND RUSHTON, S. 2004. New simple virtual walking method - walking on the spot. In *Proceedings of the IPT Symposium*.