Multi-Technique Redirected Walking method

Jesus Mayor, Laura Raya, Sofia Bayona and Alberto Sanchez

Abstract—The Room-Scale locomotion method is one of the most realistic locomotion methods used in virtual reality technologies. This is due to the natural interaction obtained through the tracking of its controllers and the head-mounted display with six degrees of freedom. However, its mapping by position between the physical and the virtual world limits the user's movement to the physical workspace provided by the corresponding device. Redirected Walking methods use gain algorithms that add modifications to the virtual movement of the user, optimizing the virtual workspace in the same physical workspace. In this paper, we develop a new Redirected Walking method, which combines the modification of three gain algorithms (curvature gain, rotation gain, and translation gain), a new algorithm (deviation gain), a path predictive method of the user's locomotion, a *proportional distance to the center* function, and a *user direction smoothing* function that softens the effect of the algorithms. Complementing the new method, we offer an automatic calibration system that allows the user to use our method in a personalized way.

Index Terms—Virtual reality, Motion, Interaction Techniques.

1 Introduction

Virtual Reality (VR) has shown great technological advances, decreasing the economic cost of its devices and, consequently, enabling access to a large number of users in recent years. Specifically, this technology foresees an increase in the volume of users of more than 16 times by 2022 with respect to 2018 [1], selling more than 29 million immersive Head Mounted Displays (HMD). With this growth in the number of users, its use has also been further expanded in different sectors such as medicine [2], psychology [3], and accessibility [4].

A characteristic of VR experiences is the user's ability to interact and move within the computer-generated virtual environment. The choice of a method to navigate or interact within the 3D virtual scenario depends on the needs of each project [5]. These locomotion methods are defined as illusions of self-motion obtained through translations and rotations of the virtual environment [6]. There are currently several methods of locomotion used in virtual experiences, such as Points of Interest, Gamepad, Teleport or Room-Scale [7]. Depending on the locomotion method chosen for the virtual world, the perceptual characteristic of presence [8] or the sensation of cybersickness can vary significantly [9].

In Mayor *et al.* [7] we analyzed different locomotion methods depending on the characteristics of presence, cybersickness, and usability. This study shows that the Room-Scale locomotion method is one of the methods that gives the user the most presence, causing less cybersickness by offering a model of interaction and locomotion similar to that used in the real world [10]. This type of interactions and natural locomotions use the movements of the hands through a controller and the absolute positioning of the HMD [10]. Natural locomotion can achieve high levels of realism, presence, and comfort with a low level of cybersickness. However, the displacement through the physical environment is reduced to the workspace provided by the tracking device when using a position mapping between the physical world and the virtual one. This limits the type of experiences that can use the Room-Scale locomotion method.

To use the Room-Scale method without limiting the design of virtual experiences, the state of the art includes different methods known as Redirected Walking (RW) [11]. These methods use the natural locomotion and interaction of the Room-Scale, but add subtle modifications to the physical movement performed by the user. In this way, the user is not fully aware of these movement alterations, thus reducing the possible cybersickness that these alterations could cause. The development of new RW methods has been a subject of recurrent research in recent years allowing to optimize a limited workspace [12].

RW methods are classified according to the type of redirection they perform: if the redirection method directs the user to the center of the room, it is called *Steer-to-Center* (S2C) [13]; if the redirection target consists of multiple central points, it is called *Steer-to-Multiple-Targets* [14]; and if the user is oriented towards a specific orbit, it is called *Steer-to-Orbit* (S2O) [13].

In order for the redirection to be effective it is necessary to predict the user path through predictive methods or reactive methods [15]. Predictive methods predict future displacements of the user from his/her previous movements. Therefore, they obtain better results than reactive methods, which react making decisions based on the current state of the user at each point.

Currently, RW methods can make users walk in a straight line within the virtual space without touching the walls, as long as the workspace is at least $31m \times 31m$ [16]. However, it is not always possible to have a workspace of these dimensions, arising the need to optimize such methods for smaller workspaces.

- *J. Mayor is with Universidad Rey Juan Carlos, U-tad and Universidad Politécnica de Madrid.*
E-mail: j.mayor.m90@gmail.com
- *L. Raya is with U-tad.*
E-mail: laura.raya@u-tad.com
- *S. Bayona is with Universidad Rey Juan Carlos and CCS.*
E-mail: sofia.bayona@urjc.es
- *A. Sanchez is with Universidad Rey Juan Carlos and CCS.*
E-mail: alberto.sanchez@urjc.es

XXXX–XXXX/0X/$XX.00 © 201X IEEE Published by the IEEE Computer Society

In this paper, we propose a new technique called deviation gain, which can be grouped into so-called gain algorithms. These gain algorithms are responsible for keeping the user away from the boundaries of the workspace through subtle modifications in the perceived virtual movement. This paper also proposes a new RW method, called *Multi-technique RW (MTRW)*, which combines three previously existing gain algorithms along with our deviation gain technique. This new method works autonomously, without the need for previous configurations by virtual experience designers. To optimize the operation of our MTRW method depending on the user, we designed a calibration system that automatically obtains the sensitivity thresholds for each of the algorithms belonging to MTRW.

The next sections are organized as follows. Section 2 analyzes the state of the art concerning current algorithms used to redirect users. Section 3 defines the proposed gain technique and the new locomotion method. Also, we detail the advances made in RW gain algorithms. Section 4 describes a user study to analyze to what extent the workspace is increased in a virtual experience. Section 5 shows the results obtained in the study. Finally, Section 6 presents a discussion, adding relevant information detected during the experiment.

2 Related Work

It is common to find papers that present methods based on RW [17], [18], [19]. The design of these methods is based on the use of redirection or repositioning techniques that, separately, apply slight modifications in the user's virtual displacement from his/her physical displacement. These techniques, also called gain algorithms [20], [21], are classified into three main types: translation or position gain, rotation gain, and curvature gain. Suma *et al.* [22] created a taxonomy in which the name of these non-perceptible algorithms by the user was specified:

- **Curvature gain:** This algorithm uses the HMD-tracked user's displacement to rotate his/her position, curving the virtual space in which he is located. In this way, the user is redirected to the center of the scene (applying S2C) or to a specific orbit (using S2O). The technique avoids the limits of the workspace and increases the virtual space traveled before reaching those limits. The curvature gain algorithm, applied in large workspaces, allows users to virtually walk in a straight line within the virtual environment indefinitely.
  Matsumoto *et al.* [23] managed, using the haptic response produced by a curved wall, to reduce the workspace necessary for a user to walk indefinitely without perceiving the rotation. The perception of curvature gain also depends on the speed at which users move [24], i.e. if the users move fast (running instead of walking) they will be more sensitive to the applied curvature gain. Additionally, users adapt to this gain algorithm with the exposure time [25], allowing an increase of the gain over time without being perceived.
  Some authors vary the curvature gain proportionally to the curvature of a shown virtual path, thus achieving better results in these gains, which is known as bending gain [26].
- **Rotation gain:** This algorithm takes advantage of the user's HMD rotations, increasing or decreasing their effect. Unlike curvature gain, this type of algorithm is applied both when the user is moving and when the user is standing and rotating the head. Bruder *et al.* [27] determine the limits that can be increased or decreased in rotations. Schmitz *et al.* [28] redefine these limits indicating that presence decreases if the user perceives the rotation gain (limits are discussed in Section 6). Additionally, these results may vary depending on the density of objects in the scene, becoming more noticeable in scenes with a greater number of objects [29].
- **Translation gain:** This algorithm is based on virtually re-scaling the translation performed by the user in the physical world. Taking advantage of the user's natural locomotion, a virtual and proportional displacement is applied to the performed one, to cover a greater virtual space. Wilson *et al.* [30] determine, through different re-scaling coefficients, to what extent the translation gain can affect efficiency when performing precise tasks. To do this, they studied the effect of several constant values of translation gain in their experiments, detecting that the ones with the highest re-escalation negatively affected the accomplishment of tasks. Additionally, some studies analyze non-constant translational gains [31], determining that users are susceptible to abrupt changes in the re-scalation of the translation, making necessary to make subtle variations.
  In any case, by walking continuously in one direction using the translation gain, one quickly reaches the limits of the workspace. To achieve a redirection of the user, it is necessary to combine it with other gain algorithms, for example, rotating the users by predicting their direction in the physical space. For this reason, the translation gain algorithm improves the use of the workspace but needs to be accompanied by other algorithms to achieve a larger virtual space.

In the state of the art, these three algorithms are often used independently, creating an RW method with a single gain algorithm, usually calibrated independently by the 2-alternative forced-choice (2AFC) method [21]. This method consists in forcing the user to choose between two possible options. In this way, after a time of exposure in the virtual world, the users are asked if they consider that there has been an increase or a decrease in their movement. The user's analysis stops when finding an optimal equilibrium point based on the answers, considering this point as the non-noticeable limit. However, this type of evaluation is not applicable in studies that use several gain algorithms because of the time and physical effort that the user should invest to analyze each of the algorithms separately [32].

Some studies combine the application of several gain algorithms to obtain greater profits. For example, Grechkin *et al.* [33] raise the possibility of generating new comple-

TABLE 1
Algorithms classified according to the type of transformation applied based on the movement carried out. The new designed algorithm is highlighted.

Name	Input movement	Output addition
Curvature Gain	Translation	Rotation
Rotation Gain	Rotation	Rotation
Translation Gain	Translation	Translation
Deviation Gain	Rotation	Translation

mentary algorithms to make RW methods more efficient by mixing existing gain algorithms with their own algorithms. Other examples combine rotation gain with curvature gain [11], [34], using a motion prediction method, or translation gain with curvature gain [32]; calculating the maximum application thresholds of both methods. Razzaque presented this combination of algorithms in [13], where he combined curvature and rotation gains proposing as a possible improvement the inclusion of translation gain.

To measure the final gain from the combination of various algorithms, in the article presented by Azmandian *et al.* [15], they proposed algorithms to evaluate the total gain applied without the need to perform tests with real subjects. However, to use this evaluation system without test subjects, it is necessary to know the perceptual thresholds of application of the algorithms used [35]. In the case of subjecting these algorithms to modifications or in the case of defining new algorithms, it is necessary to calculate these perceptual thresholds again to understand their effect on the user.

In this paper, we present deviation gain as a new gain algorithm. While rotation gain alters rotations and translation gain alters translations, deviation gain transforms rotation into translation, as opposed to the curvature gain algorithm (see Table 1). Since current HMDs can have up to 6 degrees of freedom (6DOF) in rotation and position, we explore the possibility of designing a new algorithm that transforms rotation into translation, allowing the user to be subtly translated when the head is rotated.

In relation to the deviation gain, we can find initiatives taking advantage of the moments in which the user is not able to notice the modifications applied (in the orientation or position) by the gain algorithms. Specifically, there are articles that take advantage of the saccadic suppression of the eyes to introduce subtle rotations or displacements of the user achieving promising results [36], [37]. Langbehn *et al.* [38] present a simpler solution using eye blinking to introduce these modifications. However, all these techniques are far from being used with commercial devices, since they require the use of an eye tracker, which is not available in most devices. On the contrary, deviation gain takes advantage of HMD rotations to apply translations to the user when the user is susceptible, since the rotation of the perceived images could hide this displacement.

This paper also presents a new RW method, MTRW, which aims to optimize the workspace to achieve natural interaction without significantly limiting the workspace. Our method simultaneously incorporates the algorithms of curvature gain, rotation gain and translation gain, and adds the new designed algorithm, deviation gain.

Regardless of the gain algorithms used, a large workspace is needed so that these algorithms manage to direct the user without ever reaching the limit of physical space. If the user walks into the virtual environment in a straight line, a minimum space of $31m \times 31m$ is required not to reach the limits [16]. Since it is not always possible to have a workspace of this size, there are so-called reorientation techniques that explicitly re-position the user in the physical environment.

Some of the reorientation techniques freeze the image or fade to black when the user reaches the limits of the workspace, in such a way that the user can change his/her real orientation without affecting the virtual orientation. In this way, a user can continue walking indefinitely through the virtual world [39]. To avoid the possible diminution of presence that reorientation techniques can cause, the narrative of the virtual experience can be dynamically adjusted [33], allowing the users to continue on their original path later.

In order to optimize the functioning of the MTRW method for a particular user, we designed a calibration system that obtains the sensitivity thresholds for each of the algorithms contained in our method. In the following sections, we explain in detail the complete composition of MTRW and, consequently, the deviation gain algorithm.

3 Proposed RW method

The method designed here presents a solution built with the following phases. It combines four gain algorithms, three of which are existing algorithms, and a fourth which has been specifically designed for MTRW. Additionally, to make the gain provided by these algorithms work, we designed a new path predictive method of the user's locomotion. We call it Bi-state Smoothing, and it allows, without needing special configurations, to obtain a valid prediction for most cases. The new path predictive method offers an new solution easy to apply for all types of virtual environments.

3.1 Bi-state Smoothing Predictive method

A predictive method can optimize the gain offered by algorithms by estimating the next user motion. Most current predictive methods define a set of waypoints and a prediction of the path the user will follow through those points [40], reducing the difficulty to perform a search of the chosen path. However, in this approach, waypoints in the virtual environment need to be defined beforehand. This is not always possible because, ideally, the user should be free to explore the virtual scenario, moving in 360 degrees (without having to follow the prediction of predefined routes). To eliminate this limitation in predictive methods, we looked for new solutions free of waypoints that predict the user path based on the previous movements. The designed solution is based on research found by Nescher and Kunz [41]. They use a *double exponential smoothing* method to select between different paths.

The double exponential smoothing mathematical method applied to a position delta vector $\vec{w}_t$ and a $\vec{s}_t$ prediction is defined as:

$$\vec{s}_0 = \vec{w}_0 = 0$$
$$\vec{s}_t = \alpha\vec{w}_t + (1-\alpha)(\vec{s}_{t-1} + \vec{b}_{t-1})$$
$$\vec{b}_t = \beta(\vec{s}_t - \vec{s}_{t-1}) + (1-\beta)\vec{b}_{t-1}$$

This method is based on a double linear interpolation using $\vec{b}_t$ as an intermediate step for the second interpolation. $\alpha \in \mathbb{R}(0,1)$ is defined as the *smoothing factor* and $\beta \in \mathbb{R}(0,1)$ as the *trend smoothing factor*. Smaller values of α or β will make the prediction less affected by new input data or new trends, resulting in the permanence of the prediction made.

This method predicts user trends well, but has certain limitations: (i) the method is dependent on a discretization of time. If this discretization has a high frequency, the prediction is noticeably affected by small changes in movement; (ii) the calculated trends detect movement situations well, but not situations where the user is stopped. In these situations, the double exponential smoothing method tends to fluctuate in all directions producing errors in the prediction. Since the solution provided by [41] was used to differentiate between two paths, this was not a problem.

In this paper, as an essential part of MTRW, we have designed a path predictive method that we call Bi-state smoothing, which uses double exponential smoothing but applying it with a wide discretization of time and differentiating between two new states (Walking and Idle) to avoid the fluctuations produced when the user stops.

The $\vec{s}_t$ prediction is calculated differently in each of these states.

3.1.1 *Walking State*

The predictive method presented is in the Walking state when the user is moving around the physical environment. In this state, the double exponential smoothing method works correctly and does not need to be modified.

Given the function $\mathrm{angle}(\vec{v_1}, \vec{v_2})$, which calculates the signed angle defined between two two-dimensional vectors $\vec{v}_1$ and $\vec{v}_2 \in \mathbb{R}^2$, the method switches to the Idle state when $\cos(\mathrm{angle}(\vec{s}_t, \vec{b}_{t-1})) < 0$. In cases where the difference between $\vec{s}_t \in \mathbb{R}^2$ (the new prediction) and $\vec{b}_{t-1} \in \mathbb{R}^2$ (the trend of the previous frame) is greater than $90°$, the new state is activated. This way, every time a perceptive trend shows that the user is slowing down and could stop, the predictive method will switch to the Idle state.

3.1.2 *Idle State*

The idle state is defined as the state when the user is stopping or drastically changing direction. A standing user may produce slight variations in the movement when exploring the environment or simply by the positioning of his/her body (e.g., swinging forward or backward with no intention of moving). For that reason, if the user is standing but swinging, the double exponential smoothing provides invalid predictions. Nescher and Kunz [41] indicate that it is possible to use the facing vector, a two-dimensional vector indicating the orientation of the user's head, $\vec{f}_t \in \mathbb{R}^2$. However, using a *facing vector* when the user is standing as a direct basis for this prediction can produce very changing predictions causing an erratic effect on the result of the prediction, negatively affecting the RW method.

For that reason, in this proposal, we discretized the *facing vector* into two cases: (i) if the user looks straight ahead, the prediction will continue like the last prediction calculated before switching to this state; (ii) if the user turns around, the prediction will be the opposite. Thus, in our method, the user must turn more than 90 degrees while standing to get an opposite prediction. With $\vec{v} \in \mathbb{R}^2$ being the last valid prediction before switching to the Idle state, the prediction will be maintained as long as the direction of the head is not substantially changed. The condition is given by the function $\vec{s}_t = \vec{s}_{t-1} \cdot \cos(\mathrm{angle}(\vec{v}, \vec{f}_t))$ where the angle between the *facing vector* $\vec{f}_t$ and the *last valid prediction* $\vec{v}$ is calculated, so that the prediction is inverted if the user turns around (since the cosine value will be negative).

Finally, the switching condition to the Walking state is defined by the function $|P_0 - P_t| > \gamma$, in which $P_0 \in \mathbb{R}^2$ is the last position collected before the state changed to Idle, P_t is the current position of the HMD (in two horizontal dimensions) and γ a constant indicating the extension of the rest area. When users move within the defined rest area, they are considered to be in the idle state. The function detects when users get out of the rest area, and switches to the walking state.

The new MTRW method is composed of four gain algorithms that use this new *Bi-state smoothing predictive* method, in addition to other heuristics such as the distance to the center of the workspace, as explained below.

3.2 Heuristics related to the center

We propose a method of S2C that smoothes its application depending on the distance between the user's position and the center, as well as how the prediction of the movement is oriented with respect to the center. Most redirection methods found in the state of the art do not take into account the distance to the center [15], [32]. We found solutions that use orbits (S2O) for redirection, but most of the time with a constant amount of redirection throughout the experience [23], [30]. In the case of the MTRW method, we propose to use the approach taken by some authors [11], [42], who smoothed out the application of the gain algorithms according to the distance between the user's position and the center of the workspace. Therefore, we applied for all the gain algorithms the function *proportional distance to the center* (μ_t) that smoothes its effect according to the redirection needs, prioritizing its application when the user is close to the limits of the workspace.

To do this, we differentiate the coordinates of the virtual world called $\vec{w}_t$, from the real or physical coordinates of the user with respect to his/her workspace, $\vec{w'}_t$. Being $\mu_t \in \mathbb{R}$ the *proportional distance to the center* function, the calculation of μ_t is expressed as $\mu_t = \left|\frac{||\vec{p'}_t||}{||\vec{c'}_t||}\right|$, where $\vec{p'}_t$ is the real or physical position of the user with respect to that center and $\vec{c'}_t \in \mathbb{R}^2$ the distance from a corner of the workspace to the center of the room.

According to the calculation of μ_t, if the user is in the center of the room, he/she will not receive any subtle redirection or repositioning. As the user reaches the limits of the workspace, the effect of the redirection will be maximum.

Nevertheless, the radius of action of the *proportional distance to the center* function can be reduced, thus decreasing the effect of the smoothing and making it possible to apply the algorithms in a wider area at the highest level. To do this, it is necessary to use the function $\mathrm{clamp}_{[a,b]}(x)$, which defines the minimum ($a \in \mathbb{R}$) and maximum ($b \in \mathbb{R}$) values that $x \in \mathbb{R}$ can take. In this way, by applying (1), the radius of application can be reduced according to the variable $\kappa \in \mathbb{R}$.

$$\mu_t = \mathrm{clamp}_{[0,1]}\left(\frac{||\vec{p\prime}_t||}{\kappa \cdot ||\vec{c\prime}_t||}\right) \quad (1)$$

According to the different values taken by κ, if its value is equal to 1, the radius of application of the algorithms will reach the limits of the working space. Instead, if this value is 0.5, the maximum application of the algorithms will be made from half the distance between the center and the limits of the workspace.

In addition, a similar heuristic has been used to soften the effect of the gain algorithms according to the orientation of the prediction with respect to the center using a λ variable. However, each gain algorithm needs a different approach to this smoothing. Therefore, each λ will be described in the next section where the gain algorithms used in MTRW are defined.

3.3 Redirection algorithms implementation

As stated before, several authors propose the union between several redirection or repositioning techniques or gain algorithms compatible with each other [13], [32], [34]. The design of our method is based on the Bi-state smoothing and combines four gain algorithms: curvature gain, rotation gain, translation gain, and the new deviation gain algorithm.

3.3.1 Curvature gain

This gain algorithm is based on transforming the displacement made by the user $\vec{w}_t$ into rotation of the virtual environment with a specific radius of curvature. This radius is given as a constant which we call *curvature constant*, C_c. This algorithm is the main responsible for achieving that a user can infinitely walk in a straight line inside the virtual environment in big workspaces, because if C_c contains small values, the user will never go beyond those limits.

For the implementation of this gain algorithm, we based on the solution provided by Azmandian *et al.* [15]. However, we used the real movement of the user $\vec{w\prime}_t$, instead of the virtual movement $\vec{w}_t$. Otherwise, using this algorithm simultaneously with other algorithms would generate a wrong response. For example, while some algorithms can add translation (such as translation gain), curvature gain could take advantage of that additional translation to add rotation inappropriately.

One of the common drawbacks of the traditional implementation of this gain algorithm is that, if the user moves fully towards the center, there is an erratic movement in opposite directions since the algorithm redirects the user towards the center but, as soon as the user crosses the center, the algorithm redirects the user towards the center again, just in the opposite direction.

Azmandian *et al.* [15] use a smoothing function (λ) based on the sine of the angle formed with the user's direction and that center, $\sin(\mathrm{angle}(\vec{p\prime}_t, \vec{s}_t))$, being $\vec{p\prime}_t$ the actual position of the user relative to that center, and $\vec{s}_t$ the user's prediction. This changes the direction of the curvature application automatically and smooths the effect if the user is already heading towards the center.

In our case, we modified the Azmandian's smoothing function (λ_t^a) preventing smoothing when the angle of the prediction is opposite to the direction of the center. This modification is due to the fact that, when the angle of the prediction ($\vec{s}_t$) is opposite to the center ($\vec{p\prime}_t$), the need for rotation of our RW method is greatest. Remembering that ($\vec{w\prime}_t$) was the physical displacement made by the user, the following equation defines the equations applied in the curvature gain calculation (CG_{t+1}):

$$\lambda_t^a = \sin(\mathrm{clamp}_{[-\frac{\pi}{2},\frac{\pi}{2}]}(\mathrm{angle}(\vec{p\prime}_t, \vec{s}_t)))$$

$$CG_{t+1} = \frac{||\vec{w\prime}_t||}{C_c} \cdot \lambda_t^a \cdot \mu_t$$

3.3.2 Rotation gain

This method modifies the rotation produced by the user with respect to the vertical axis (yaw axis) by adding or decreasing its effect. Unlike curvature gain, rotation gain does not require an offset to be applied, so it is important to make a correct prediction also when the user is standing still. For the development of this algorithm we based on the code provided by Azmandian *et al.* [15], which combines curvature gain and rotation gain applying the gain with the highest absolute value between both techniques as Razzaque *et al.* [11] defined in previous works.

As defined in the following equation, the calculation of the rotation gain is given by the actual angular velocity of the head, $r\prime_t \in \mathbb{R}$, and a constant of rotation increase, $R_c \in \mathbb{R}$.

$$RG_{t+1} = ||r\prime_t|| \cdot R_c \cdot \lambda_t^a \cdot \mu_t$$

However, in our case, depending on the movement prediction $\vec{s}_t$ (used in the smoothing function λ_t^a), the users will be redirected to the center of the room, whether they are standing or walking.

3.3.3 Translation gain

The translation gain algorithm transforms the user's actual displacement by adding or decreasing the virtual displacement. In the state of the art, fixed amplitudes are often applied regardless of the user's movement prediction [19], [30]. In the present case, similarly to the previous algorithms, the application of this gain is variable and dependent on the user's prediction. Our approach takes into account the possibility that the user does not always have to walk great virtual distances. Thus, our algorithm addresses the application of translation gain applied in a variable way with the objective of keeping the user in the center of the workspace as much as possible. To achieve this effect, we used this new smoothing function:

$$\lambda_t^b = \cos(\mathrm{angle}(\vec{p\prime}_t, \vec{s}_{(t)}))$$

In this case, as the method does not use rotations, we applied the algorithm independently to each axis, making its application not radial but quadrangular (such as the workspace of VR devices). In the following equation, the translation gain $(TG_{x(t+1)})$ is defined, where T_c is the *translation constant* that regulates the effect and $\operatorname{sgn}(x)$ is the function that returns the sign (1 or -1) depending on the sign of the number indicated by the translation in the x axis:

$$TG_{x(t+1)} = -\operatorname{sgn}(p\prime_{x(t)}) \cdot |w\prime_{x(t)}| \cdot T_c \cdot |\lambda_t^b| \cdot \mu_{x(t)}$$

3.3.4 Deviation gain

Specifically, for MTRW, we designed a new technique that allows transforming the user's rotations into translations. The algorithm is based on curvature gain but applying the necessary modifications to perform this inverse conversion. This algorithm can take effect whether the user is walking or standing (using prediction). However, its application has been limited when the added movements are lateral to the user's gaze, thus avoiding the possible cybersickness that this could cause.

In this case, a deviation constant D_c transforms the real rotation $r\prime_t$ into virtual displacement. Thus, while the resulting curvature gain (GC) had units of rad/m and defined a curvature per unit of distance applied, in the deviation gain DC is defined in m/rad units and it defines a rotation necessary to move a certain distance. The smoothing factor λ_t^c uses the facing vector $\vec{f_t}$ to apply the effect of the deviation gain DG_{t+1}.

$$\lambda_t^c = \cos(\operatorname{angle}(\vec{f_t}, \vec{s_t}))$$

$$DG_{t+1} = \frac{\vec{s_t}}{||\vec{s_t}||} \cdot \frac{|r\prime_t|}{D_c} \cdot |\lambda_t^c| \cdot \mu_t$$

Because the result of dividing the rotation $(|r\prime_t|)$ by the rotation constant (D_c) is a scalar, it must be applied to a specific direction. It is worth noting the use of $\vec{s_t}$ as the direction of application of this algorithm, which uses the normalized prediction vector to direct the new translation gain. As in the previous techniques, the distance factor μ_t gives priority to the algorithm applying a greater or lesser effect depending on the distance to the center. The same goes for the smoothing factor (λ_t^c). However, for large values of D_c, the effect is decreased; smaller values achieve a higher translation.

Like translation gain, deviation gain does not redirect the users to the center but moves them to where the prediction indicates they will go. In this way, this algorithm causes the users to achieve their goal sooner, as long as they are not already in the center of the room.

Analogously to the case of curvature and rotation gains [11], we combined the algorithms of translation gain and deviation gain by applying the gain with the highest absolute value obtained from both techniques.

The nature of this new gain algorithm allows it to be applied whether the user is walking or not. However, the prediction will be more accurate if the user is walking, being the deviation gain more effective. Even so, this method can also displace the user if he is standing and moving the head and the prediction is still adequate.

To summarize the MTRW algorithm, Fig. 1 shows a diagram that relates and summarizes the parts of the developed RW method. Likewise, in the supplementary material of this article, you can observe the functioning of the four algorithms.

4 Experiment

We designed an automatic calibration system to determine the values of the gain algorithms separately and together. Additionally, we created an experimental environment to apply the calibration system and analyze the behavior of MTRW.

The study consists of two parts. The first is a preliminary study to understand the perceptual ranges expected for each of the gain algorithms in users. The second study uses that range to progressively detect the real perceptual threshold for each user and then compares MTRW to Room-Scale. These studies are detailed in the following sections.

4.1 Specification of objectives

The study objectives are:

- To calculate the maximum limits that optimize the gain but are not noticeable by the user for each of the separate gain algorithms.
- To explore the optimal use of the workspace by combining the different algorithms compared with Room-Scale. For this purpose, the effort made by users with each algorithm is analyzed, as well as the number of times the limits of the workspace are reached.
- To study the users' sensations regarding the use of MTRW compared with Room-Scale using a post-exposure questionnaire.

4.2 Method

For this study, we used the HTC Vive HMD, because it covers a fairly large workspace thanks to its tracking technology. It is based on two laser emitting lighthouses that allow tracking the HMD. This device enables the natural interaction required for this method of RW, allowing the user to move naturally (in 6DOF) and interact with two controls that represent their own hands (also in 6DOF). The experiment was carried out in an area of $4m \times 4m$, this being the limit allowed by this device.

4.2.1 Study population

The study was conducted with an initial group of seven subjects and an experimental group of 32 participants,: university students and faculty related to technology and arts, of whom 17 were men and 15 women, with a median age of 20 years (range = 18-48, mean = 22.75, standard deviation (SD) = 7.47). Of this population, 40.63% had never used VR technology, 53.13% had used it less than 10 times, and 6.26% more than 10 times.

The studies conducted were blind, the users being unknown of the methods being applied. An informed consent was given to all users indicating that the collected data would be stored anonymously.

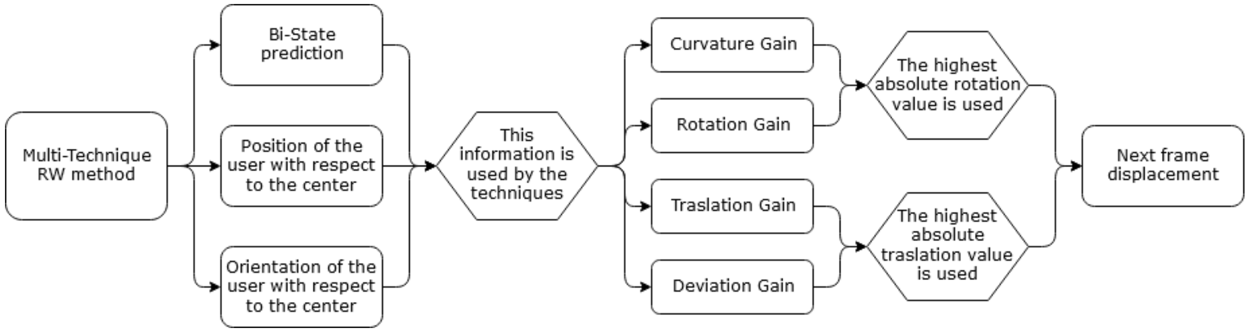


Fig. 1. Steps followed by the Multi-technique RW method (MTRW) to calculate the subtle displacement needed for the next frame.

4.2.2 Procedures

The intervention protocol assured that all experiments were conducted by the same supervisor, who read the instructions at the beginning of the experiment, providing the same explanations to all subjects. The duration of each part of the experiment was defined in advance and will be described in their corresponding section.

All users went through the same habituation method programmed into the virtual experience as a tutorial, which introduced the basic controls to pick elements and to report their perceptual thresholds.

The experiments were carried out in the same conditions for all users: with the same temperature and in the same room. The room was dark with completely black and opaque walls to avoid reflective surfaces (to prevent possible malfunctions of the device).

4.2.3 Experimental setup

Since the workspace used in the experiment is that defined by the HTC Vive ($4m \times 4m$), it is necessary to implement a reorientation technique that explicitly expands the space if the limits are reached by the user [43]. This technique consists in that, once the user reaches the limits of the workspace, a fade to black is performed and the user stops visualizing the virtual world. After this, arrows indicate that the user must turn 180º, forcing him to reorient himself towards the center of the physical workspace. During this turn, the virtual environment is blocked so that once the user returns to the virtual scene, the environment appears exactly as before. This allows the users to continue moving towards their objective and thus travel greater distances in the virtual environment.

Additionally, the correct prediction of the predictive method depends on the configuration of its factors α and β. Therefore, we used a configuration similar to that defined by Nescher and Kunz [41], with a higher time discretization (0.2s) and, proportionally, values for its constants of $\alpha = 0.3$ and $\beta = 0.3$. In our case, this discretization of time improves the efficiency of the algorithm, managing to optimize the computation by decreasing the frequency of calculation of the new prediction, and obtaining a satisfactory result. Likewise, increasing the computation time helps to maintain a more stable prediction, since, in that time, it is difficult for the user to change direction several times. The validity of this configuration was checked during the preliminary study with 7 subjects, verifying that the values worked as desired. On the other hand, the smoothing of the algorithms based on the distance to the center of the room was configured with a value of $\kappa = 0.5$. Thus, a radius of half of the workspace was used, smoothing the effect of the gain algorithms from the center to this radius.

Finally, the significance level for all hypothesis testing performed was set at ($p < 0.05$).

4.2.4 Study variables

The variables that are intended to be obtained through the studies are the constants defined for each of the algorithms.

- Curvature constant (C_c) measured in m/rad, lower values meaning greater rotational gain.
- Rotation constant (R_c) measured in rad/rad, higher values meaning greater rotational gain.
- Translation constant (T_c) measured in m/m, higher values meaning greater positional gain.
- Deviation constant (D_c) measured in rad/m, lower values meaning greater positional gain.

We also consider the questionnaires and metrics used to evaluate MTRW against Room-Scale. By obtaining the values of these constants and applying them for subsequent MTRW analysis, it is possible to solve the previously raised objectives to be analyzed. Both the preliminary study and the final study are explained below.

4.3 Preliminary study

Before starting with the system that allows the calibration of the different algorithms, we carried out a preliminary study with an initial group of 7 subjects, to estimate the ranges of the constants of the different gain algorithms used that will define an initial configuration for the calibration system. This preliminary study aimed to define ranges in which to increase these constants progressively. The perceptual thresholds to be sought in the calibration phase should be found within these ranges.

To obtain these ranges, the subjects had to explore the virtual environment by walking. Their sensitivity to the different techniques was analyzed with two different criteria:

- Unconscious criterion: Neither the algorithms were explained to the users nor what they should perceive. They were told only that if they noticed strange behavior in the operation of the HMD, they should report it immediately.

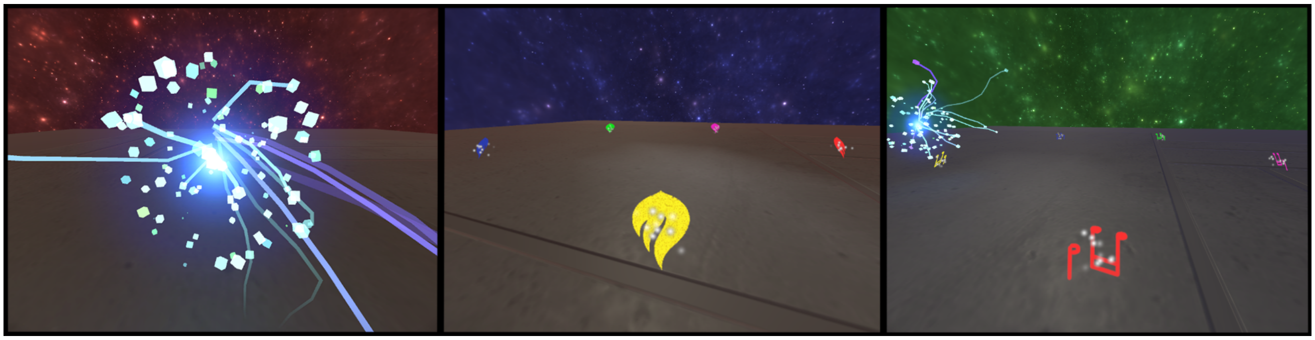


Fig. 2. Captures of the virtual environment of the calibration system and the experimental evaluation environment. In the figure on the left, the spirit that guides the user through the narration in the calibration system. The central image represents the MTRW method. In the image on the right, the objects that must be touched in the task to be performed with the Room-Scale method.

TABLE 2
Averages of the perceptual thresholds in the preliminary analysis according to each criterion.

	Unconscious criterion	Conscious criterion	Scaled limits
Curvature (C_c)	2.93 ± 0.71	4.81 ± 0.87	[1.99, 5.75]
Rotation (R_c)	0.66 ± 0.12	0.24 ± 0.05	[0.03, 0.87]
Translation (T_c)	0.96 ± 0.34	0.47 ± 0.23	[0.23, 1.21]
Deviation (D_c)	2.70 ± 0.70	4.63 ± 0.05	[1.74, 5.60]

- Conscious criterion: After the unconscious criterion, the experiment was repeated with each of the algorithms separately, but this time indicating to the users what they should perceive in each case. In this way, the user was aware of the method and a lower perceptual limit was obtained.

The values of the constants were progressively increased every 20 seconds until users verbally reported detecting them. From the arithmetic means obtained with the different subjects, a range of values was defined which was scaled double, thus ensuring that the perceptual threshold of the experiment population was within the limits of this range (for each of the algorithm constants). Table 2 shows the arithmetic means obtained for each of the constants and the range finally used to be progressively covered in the calibration system.

4.4 Multi-technique RW study

We developed an experimental environment in Unity 2018.2.6f (see Fig. 2) to calibrate and evaluate the MTRW method. The experience was provided with a short narrative to guide the users and give them feedback as they went through the different phases. The experimental environment consists of a large virtual space with virtual dimensions larger than the physical dimensions of the device's workspace.

4.4.1 Algorithms calibration system

In the calibration system, we used the determined range in which the general perceptual values of the users are found. The calibration system allows us to obtain the values adapted to each user to optimize the operation of MTRW.

In this calibration, for each constant to be calibrated, every second, 0.5% of its value is increased, starting from the minimum value of that range calculated in the previous initial experiment until reaching the maximum value. Due to this increasing rate, the maximum time allowed to calibrate each gain algorithm is 200 seconds, the time needed to reach the maximum value with the defined increment. Users are instructed to immediately press the button on the HTC Vive control to alert the calibration system when they notice anything unusual in the movement. At that moment, the system records the previous perceptual value for which the user did not detect any anomaly, and the user goes to a state of rest where no algorithms are applied.

To enable the subjects to move through the virtual environment of the calibration system, the system provides the users with a task that forces them to continually move through the scenario in the direction of different randomly positioned target points. We indicated by five virtual objects the points that the subjects must reach, including a correct object, which allows the users to continue with their task. The selection of which virtual object is the correct one is randomly chosen from the five available each time. The users' way of indicating their choice between the different virtual objects will be by interacting with the virtual objects using the HTC Vive device. Once the user finds the correct object, that object is split into five new objects moved 1.5 meters away from the point of the last correct object selected. The randomness of the target points makes it easy for the user to move in any direction without having defined waypoints in the scene and to move around the environment within a circle (formed by the five objects) of 3 meters in diameter from the last correct selected object. The image of the center of Fig. 2 shows the elements of different colors that must be reached by the user during the whole experience. This task is repeated for the four gain algorithms that are calibrated in random order and separately, to avoid possible dependencies between the algorithms.

Therefore, the system automatically records the largest value not indicated as perceptible by the subject for the constants C_c, R_c, T_c, and D_c. With these values, the user can use our MTRW method in a customized way, without the need to use average values obtained from a population. Likewise, the arithmetic mean of the calibrations of each algorithm is used as the users' perceptual threshold, to be later compared to previous works. We developed this calibration system independently so that it can be reused in other experiments.

4.4.2 Multi-technique RW evaluation

To evaluate MTRW, two phases are shown to the users. One phase uses MTRW method and the other phase uses the Room-Scale method. We balanced the order of the phases for the different users. The color of the sky of the environment changes according to the locomotion method being tested by the user, which allows us to subsequently identify the user's perceptual sensations with each method keeping users blind to the applied method (see Fig. 2).

Analogously to the calibration study, the subjects performed a task that forced them to navigate through the designed virtual environment, taking them to the extremes of the workspace. The user must reach different objectives randomly located in the virtual environment. Both the environment and the task to be performed by the user are the same for both methods. Navigation duration with each of the locomotion methods was set at 2 minutes.

For the analysis of each method, we recorded the following metrics: the real and virtual distances traveled by each user, and the times the limits of the workspace were reached. Once the task was completed, each user filled out a questionnaire to obtain information about three specific perceptual sensations and we compared if there were differences between MTRW and Room-Scale.

The questions asked to the users were:

1) How comfortable did you feel using the method of the environment with the blue/green sky?
2) Were you aware of covering more virtual space before reaching the limits of the real space when the sky was blue/green?
3) Were you aware of the orientation of the real environment when the sky was (blue/green)?

All questions were answered using a 5-point Likert scale, with 1 being not at all comfortable/aware and 5 being very comfortable/aware.

5 RESULTS

5.1 Calibration system results

Regarding the automatic calibration system, the time used for a complete calibration was dependent on the user's perception. The average time for the four gain algorithms together was 632.2 seconds, with a standard deviation of 218.09 seconds. The average thresholds obtained in the calibration system for the constants of each of the algorithms are: Curvature constant (C_c) = 3.29±1.10 m/rad, Rotation constant (R_c) = 0.496±0.221 rad/rad, Translation constant (T_c) = 0.740±0.288 m/m and Deviation constant (D_c) = 4.11±1.29 rad/m.

Knowing these mean constants for each of the gain algorithms, we calculated the upper limit of application of each one of them, since it is when the algorithm has more effect. This limit occurs when the user is at a maximum application distance (1.41 meters from the center) and the prediction is contrary to the center $\mu_t = 1$ and $\lambda_t = 1$. In this way, applying the limits obtained to the formulas described in the methods, we would obtain average thresholds of perception of CG = ± 0.30 rad/m, RG = ± 0.50 rad/rad per real degree, TG = ± 0.74 m/m (for each axis in the horizontal plane), and DG = ± 0.24 m/rad. Under this limit approach of λ_t and μ_t, the achieved curvature would be 6.58 meters in diameter at the moment it begins to be perceptible by the user.

Additionally, there is extra use of the rotations through the rotation gain algorithm, which could further reduce the necessary working space. Likewise, the algorithms that add a translation (translation gain and deviation gain) allow the users to reach their objective sooner, giving versatility to the MTRW.

TABLE 3
The MTRW method is compared with the Room-Scale (RS) method for the variables of the physical distance traveled and the physical rotations performed (these metrics are referred to the mean per second in the experiment). T. limits variable represents the number of times users have reached the limits of the workspace.

	Physical distance	Physical rotation	T. limits
Sig.	.001	.513	.001
MTRW	0.181m/s±0.066m/s	54.44º/s±16.67º/s	2.44±2.21
RS	0.197m/s±0.061m/s	53.78º/s±16.03º/s	4.63±2.72

5.2 MTRW evaluation results

Regarding the analysis of our method vs the Room-Scale method, users tested MTRW method with their own values obtained during calibration. For each user, we recorded the amount of real displacement, amount of real rotation and the number of times the limits were reached. Applying the Shapiro-Wilk test for normality to the movement variables, the sample did not follow a normal distribution.

A Wilcoxon Signed-rank test was performed for the virtual space metrics, to compare the MTRW and Room-Scale methods. The difference between both is that MTRW introduces subtle modifications to the natural locomotion through the gain algorithms detailed above and Room-Scale uses a clean natural locomotion, mapping the real displacement with the virtual one. Table 3 shows the results obtained during the 2 minutes of the experiment for each of the methods.

The results show significant differences in the amount of real displacement performed in both methods ($p = .001$), with Room-Scale users tending to perform a slightly higher physical displacement (with a difference of 0.015 m/s). However, in physical rotation, we did not find significant differences, so we cannot say that the head is rotated more in one method than in the other, despite the modifications made by the algorithms.

Specifically, we applied movement modifications of 0.160m/s ±0.095m/s with respect to the actual 0.181m/s traveled. Therefore, a 88.4% was finally applied in movement modifications during the experiment. In rotations, the movement alterations applied were 12.783º/s ±6.67º/s with respect to the actual 55.44º/s rotations made, so a 23.06% was finally applied in gain modifications to the rotation during the experiment.

These modifications applied in translation (0.160m/s) and rotation (12.783º/s) resulted in a median reach of the

TABLE 4
MTRW versus Room-Scale questionnaire comparison.

	Sig.	MTRW median	RS median
Comfort	.000	2.0	4.0
Cons.Space	.728	3.0	3.0
Orientation	.982	1.5	2.0

limits using MTRW of 2.5 (mean=2.44) times during the course of the experiment (2min), compared to the median reach of the limits in Room-Scale of 4.5 (mean=4.63) times. This implies that, on average, users reach the limits 52.7% times less in the $4m \times 4m$ physical workspace in which the experiment was performed. Again, we performed a Wilcoxon Signed-rank test on both samples, showing that this difference is significant (p = .001).

Therefore, when navigating with the MTRW method, there is a significant improvement in the effort made by the users (meaning a smaller displacement of the users) and a significant reduction of the number of times that the users reach the limits of a $4m \times 4m$ workspace (increasing more the virtual space traveled).

5.3 Perceptual user analysis

After experimenting with each of the methods, users carried out the questionnaire to evaluate their perceptive sensations. Again, we performed a Shapiro-Wilk test to check the distribution of the sample obtained, which turned out to be non-parametric for all its variables. Therefore, the subsequent analysis was performed with non-parametric tests.

We performed a Wilcoxon Signed-rank test to compare two non-parametric samples. In particular, the responses given for the blue-sky environment (Room-Scale method) were compared to the ones given for the green sky environment (MTRW method). Table 4 shows the results obtained in the significance test, as well as the medians obtained for each of the questions asked.

In spite of the variations of movement applied by MTRW, both in rotation and in translation, the results showed no significant differences concerning orientation or awareness of virtual space between the two methods. On the other hand, user comfort decreased significantly, users feeling generally more comfortable with the Room-Scale method ($p = .000$).

6 Discussion

This article has designed deviation gain as a new gain algorithm, which has been part of the new MTRW method. In addition, a calibration method was designed that allows the gain algorithms (CG, RG, TG, and DG) to be calibrated so that they may be used individually or jointly. One of our objectives was to analyze the perceptible limits obtained through the modifications made in the different algorithms and to compare them with other studies. It should be noted that our calibration method did not use 2AFC in the capture of these perceptual limits for the algorithms, instead, the non-noticeable threshold limit was detected for each person.

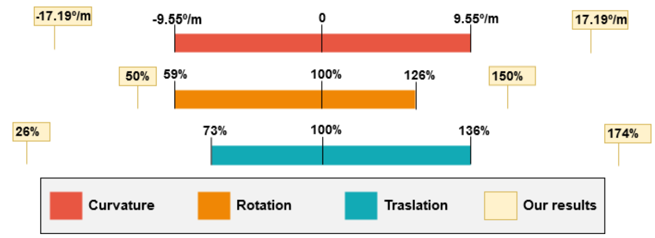


Fig. 3. Graph that compares the scales of each of the results obtained with our calibration. The colored bars represent the largest decremental and incremental margins collected by Langbehn and Steinicke [21].

Thanks to the survey carried out by Langbehn and Steinicke [21], in which they compiled all the metrics obtained in numerous studies for GC, GR and TG algorithms, the perceptual values obtained in previous studies are available. In the state of the art, we can observe that the articles differentiate, for rotations and translations, a lower limit of decrease of the movement and an upper limit of increase of the movement performed. Due to the characteristics of our study, we have not differentiated between both, applying the same amount of gain in both increase and decrease of the movement.

Comparing to the studies reported in the paper of Langbehn and Steinicke [21], the algorithms of our method obtained promising results. In particular, for curvature, where our method obtained 17.19º/m, the maximum value of the other studies was 9.55º/m [17]. For rotation, our method obtained a limit in decrease and increase of $\pm 0.50\%$ (0.50 - 1.50), whereas the highest limits obtained by other studies are 0.59 - 1.1 [35] (highest limit in decrease) or .68 - 1.26 [18] (highest limit in increase). Finally, for translation, our method obtained a perceptual limit of $\pm 0.74\%$ (0.26 - 1.74), and the highest limits obtained by other studies are 0.73 - 1.25 [19] (highest decreasing limit) or 0.94 - 1.36 [18] (highest increasing limit). Finally, for deviation gain, the method obtained a perceptual limit of 4.11 rad/m. However, being a new algorithm, there is no reference in the literature against which to compare it. Fig. 3 visually summarizes these comparisons in a graph.

From the results obtained through the contrast of hypotheses, it is interesting to highlight the significant differences found in the amount of real space traveled in MTRW method with respect to Room-Scale. These differences show that, with the Room-Scale method, users tend to walk more to reach the same objectives, making a greater physical effort. Likewise, the number of times the limits of the workspace are reached was notably reduced. However, the results obtained showed that, despite not being aware of the greater space traveled or feeling more disoriented, there is a decrease in the general comfort of the users. This result is to be expected since RW methods should only be used in those cases where Room-Scale does not allow for adequate functionality because of the limited space. Our MTRW algorithm aims to be an RW method that improves the Room-Scale workspace in those cases where the requirements of the immersive experience need wide movements from the user without frequently reaching the limits. However, in experiences that can be performed within the defined workspace, it is preferable to use Room-Scale without including movement alterations to avoid discomfort.

Previous studies focused on the application of the algorithms in a subtle way, trying to be imperceptible. In our case, we looked for the previous point before the users notice the movement gain. In future studies it would be interesting to analyze, through specialized questionnaires, whether this new approach affects factors such as presence (with the IPQ [44]) or cybersickness (with the SSQ [45]). In this way, a generally acceptable point of application could be found for the user, with good levels in these three factors, being redirection noticeable or not.

During the study, the new gain algorithm designed, deviation gain, showed a wide acceptance by users. Once detected, 6 of the subjects tried to use head rotation to move around without walking. This indicates that it could be interesting to analyze this algorithm as an independent 6DOF locomotion method.

In spite of implementing four algorithms, MTRW still does not manage to take advantage of the workspace enough to never reach the limits in medium reduced spaces ($4m \times 4m$). Based on the average perception threshold obtained for the curvature gain, we could estimate the minimum necessary workspace. Under ideal conditions, in which the smoothing functions would keep the effect of the algorithms at maximum all the time, the required workspace should be at least $6.58m \times 6.58m$ to prevent the user to reach the limits. Note that, this is only an ideal case, and it would be necessary to carry out future experiments to identify the minimum space needed to apply MTRW without reaching the limits of the workspace, thus avoiding the use of reorientation techniques.

Likewise, other types of technological improvements could be applied to reduce even more the workspace used, for example, through the use of eye-tracking technology, which can add perceptual improvements to RW methods [36], [37].

ACKNOWLEDGMENTS

This work has been partly supported by the Spanish Ministry of Science, Innovation and Universities (grant RTI2018-098694-B-I00 and TIN2017-83132-C2), Indra and Universia Foundation (CicerOn Virtual Speech Coach).

REFERENCES

[1] "Virtual reality device market declines in 2018 but outlook remains positive." https://www.ccsinsight.com/press/company-news/3726-virtual-reality-device-market-declines-in-2018-but-outlook-remains-positive/. Accessed: 2019-04-29.

[2] J. L. McGrath, J. M. Taekman, and et al., "Using virtual reality simulation environments to assess competence for emergency medicine learners," *Academic Emergency Medicine*, vol. 25, no. 2, pp. 186–195, 2018.

[3] M. Peskin, K. Wyka, J. Cukor, M. Olden, M. Altemus, F. S. Lee, and J. Difede, "The relationship between posttraumatic and depressive symptoms during virtual reality exposure therapy with a cognitive enhancer," *Journal of Anxiety Disorders*, 2018.

[4] D. Rojo, J. Mayor, J. J. G. Rueda, and L. Raya, "A virtual reality training application for adults with asperger's syndrome," *IEEE Computer Graphics and Applications*, vol. 39, pp. 104–111, March 2019.

[5] E. Langbehn, P. Lubos, and F. Steinicke, "Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking," in *Proceedings of the Virtual Reality International Conference (VRIC)*, (Laval), 2018.

[6] K. S. Hale and K. M. Stanney, *Handbook of Virtual Environments: Design, Implementation, and Applications*. Boca Raton, FL, USA: CRC Press, Inc., 2nd ed., 2014.

[7] J. Mayor, L. Raya, and A. Sanchez, "A comparative study of virtual reality methods of interaction and locomotion based on presence, cybersickness and usability," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2019.

[8] M. Slater and S. Wilbur, "A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 6, pp. 603–616, 1997.

[9] M. E. McCauley and T. J. Sharkey, "Cybersickness: Perception of self-motion in virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 1, no. 3, pp. 311–318, 1992.

[10] C. Bassano, F. Solari, and M. Chessa, "Studying natural human-computer interaction in immersive virtual reality: A comparison between actions in the peripersonal and in the near-action space," in - *HUCAPP,*, INSTICC, SciTePress, 2018.

[11] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected walking," in *Eurographics 2001 - Short Presentations*, Eurographics Association, 2001.

[12] E. R. Bachmann, E. Hodgson, C. Hoffbauer, and J. Messinger, "Multi-user redirected walking and resetting using artificial potential fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 2022–2031, May 2019.

[13] S. Razzaque, *Redirected Walking*. PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 2005. AAI3190299.

[14] E. Hodgson and E. Bachmann, "Comparing four approaches to generalized redirected walking: Simulation and live user data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 634–643, April 2013.

[15] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma, "The redirected walking toolkit: a unified development platform for exploring large virtual environments," in *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, pp. 9–14, March 2016.

[16] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma, "Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance," in *ICAT-EGVE 2015 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments* (M. Imura, P. Figueroa, and B. Mohler, eds.), The Eurographics Association, 2015.

[17] *Detection Thresholds in Audio-visual Redirected*, Zenodo, Aug. 2016.

[18] G. Bruder, V. Interrante, L. Phillips, and F. Steinicke, "Redirecting walking and driving for natural navigation in immersive virtual environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 538–545, April 2012.

[19] L. Kruse, E. Langbehn, and F. Stelnlcke, "I can see on my feet while walking: Sensitivity to translation gains with visible feet," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 305–312, March 2018.

[20] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, F. Steinicke, and E. S. Rosenberg, "15 years of research on redirected walking in immersive virtual environments," *IEEE Computer Graphics and Applications*, vol. 38, pp. 44–56, Mar 2018.

[21] E. Langbehn and F. Steinicke, *Redirected Walking in Virtual Reality*, pp. 1 – 11. Springer Encyclopedia of Computer Graphics and Games, 2018.

[22] E. A. Suma, G. Bruder, F. Steinicke, D. M. Krum, and M. Bolas, "A taxonomy for deploying redirection techniques in immersive virtual environments," in *2012 IEEE Virtual Reality Workshops (VRW)*, pp. 43–46, March 2012.

[23] K. Matsumoto, Y. Ban, T. Narumi, Y. Yanase, T. Tanikawa, and M. Hirose, "Unlimited corridor: Redirected walking techniques using visuo haptic interaction," in *ACM SIGGRAPH 2016 Emerging Technologies*, SIGGRAPH '16, (New York, NY, USA), pp. 20:1–20:2, ACM, 2016.

[24] C. T. Neth, J. L. Souman, D. Engel, U. Kloos, H. H. Bulthoff, and B. J. Mohler, "Velocity-dependent dynamic curvature gain for redirected walking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 1041–1052, July 2012.

[25] L. Bölling, N. Stein, F. Steinicke, and M. Lappe, "Shrinking circles: Adaptation to increased curvature gain in redirected walking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 2032–2039, May 2019.

[26] E. Langbehn, P. Lubos, G. Bruder, and F. Steinicke, "Bending the curve: Sensitivity to bending of curved paths and application in

room-scale vr," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 1389–1398, April 2017.

[27] G. Bruder, F. Steinicke, K. Hinrichs, and M. Lappe, "Reorientation during body turns," in *Proceedings of the 15th Joint Virtual Reality Eurographics Conference on Virtual Environments*, JVRC'09, (Aire-la-Ville, Switzerland, Switzerland), pp. 145–152, Eurographics Association, 2009.

[28] P. Schmitz, J. Hildebrandt, A. C. Valdez, L. Kobbelt, and M. Ziefle, "You spin my head right round: Threshold of limited immersion for rotation gains in redirected walking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, pp. 1623–1632, April 2018.

[29] A. Paludan, J. Elbaek, M. Mortensen, M. Zobbe, N. C. Nilsson, R. Nordahl, L. Reng, and S. Serafin, "Disguising rotational gain for redirected walking in virtual reality: Effect of visual density," in *2016 IEEE Virtual Reality (VR)*, pp. 259–260, March 2016.

[30] G. Wilson, M. McGill, M. Jamieson, J. R. Williamson, and S. A. Brewster, "Object manipulation in virtual reality under increasing levels of translational gain," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, (New York, NY, USA), pp. 99:1–99:13, ACM, 2018.

[31] R. Zhang, B. Li, and S. A. Kuhl, "Human sensitivity to dynamic translational gains in head-mounted displays," in *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*, SUI '14, (New York, NY, USA), pp. 62–65, ACM, 2014.

[32] T. Grechkin, J. Thomas, M. Azmandian, M. Bolas, and E. Suma, "Revisiting detection thresholds for redirected walking: Combining translation and curvature gains," in *Proceedings of the ACM Symposium on Applied Perception*, SAP '16, (New York, NY, USA), pp. 113–120, ACM, 2016.

[33] T. Grechkin, M. Azmandian, M. Bolas, and E. Suma, "Towards context-sensitive reorientation for real walking in virtual reality," in *2015 IEEE Virtual Reality (VR)*, pp. 185–186, March 2015.

[34] T. Nescher, Y.-Y. Huang, and A. Kunz, "Planning redirection techniques for optimal free walking experience using model predictive control," in *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 111–118, March 2014.

[35] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, "Analyses of human sensitivity to redirected walking," in *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, (New York, NY, USA), pp. 149–156, ACM, 2008.

[36] B. Bolte and M. Lappe, "Subliminal reorientation and repositioning in immersive virtual environments using saccadic suppression," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 545–552, 2015.

[37] Q. Sun, A. Patney, L.-Y. Wei, O. Shapira, J. Lu, P. Asente, S. Zhu, M. Mcguire, D. Luebke, and A. Kaufman, "Towards virtual reality infinite walking: Dynamic saccadic redirection," *ACM Trans. Graph.*, vol. 37, pp. 67:1–67:13, July 2018.

[38] E. Langbehn, F. Steinicke, M. Lappe, G. F. Welch, and G. Bruder, "In the blink of an eye: Leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality," *ACM Trans. Graph.*, vol. 37, July 2018.

[39] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. Rieser, and B. Bodenheimer, "Exploring large virtual environments with an HMD when physical space is limited," in *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization*, APGV '07, (New York, NY, USA), pp. 41–48, ACM, 2007.

[40] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma, "Automated path prediction for redirected walking using navigation meshes," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 63–66, March 2016.

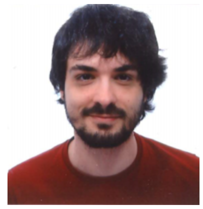
[41] T. Nescher and A. Kunz, "Using head tracking data for robust short term path prediction of human locomotion," in *Transactions on Computational Science XVIII* (M. L. Gavrilova, C. J. K. Tan, and A. Kuijper, eds.), (Berlin, Heidelberg), pp. 172–191, Springer Berlin Heidelberg, 2013.

[42] E. Hodgson, E. Bachmann, and D. Waller, "Redirected walking to explore virtual environments: Assessing the potential for spatial interference," *ACM Trans. Appl. Percept.*, vol. 8, Dec. 2008.

[43] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. Rieser, and B. Bodenheimer, "Exploring large virtual environments with an hmd on foot," in *ACM SIGGRAPH 2006 Research Posters*, SIGGRAPH '06, (New York, NY, USA), ACM, 2006.

[44] T. Schubert, F. Friedmann, and H. Regenbrecht, "The experience of presence: Factor analytic insights," *Presence: Teleoperators and Virtual Environments*, vol. 10, no. 3, pp. 266–281, 2001.

[45] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *The International Journal of Aviation Psychology*, vol. 3, no. 3, pp. 203–220, 1993.



Jesus Mayor is a Ph.D. in Computer science at Universidad Rey Juan Carlos (Madrid, Spain) since 2020. He also holds an MS degree in Computer Graphics and Simulation from Centro Universitario de Tecnología y Arte Digital (U-tad). In 2013 he worked as a research fellow at Universidad de Alcala. In 2016 he was a full-time lecturer and virtual reality community organizer at U-tad. Since 2019 he has been a full-time lecturer and researcher in Universidad Politécnica de Madrid. His main fields of research are virtual reality, user experience, computer graphics, and acoustic simulation.

Laura Raya is a professor and researcher at Centro Universitario de Tecnología y Arte Digital (U-tad), Spain. She received an MS degree in Computer Science, an MS degree in Computer Graphics and a PhD degree in Computer Science from Universidad Rey Juan Carlos of Madrid in 2008, 2010 and 2014, respectively. Since 2013 she is the head of the master's degree and the manager of Virtual Reality Projects at the Department of Computer Science, at U-tad (Madrid, Spain).

Sofia Bayona obtained her PhD in Computer Science from the URJC. She worked at INRIA Rhone-Alpes (France) in 2003 and at Imperial College London in 2009-2010 when she was awarded an FP7 Marie Curie Intra European Fellow (grant PIEF-GA-2009-236642). She holds an Associate Professor position at Universidad Rey Juan Carlos. Her research interests include virtual reality, scientific visualization, and education.

Alberto Sanchez is an associate professor and researcher at Universidad Rey Juan Carlos and CCS, Spain. He received MS and PhD degrees, obtaining the Extraordinary PhD Award, in Computer Science from Universidad Politecnica de Madrid in 2004 and 2008, respectively. In 2006 he obtained his MS degree in Marketing at the Universidad Autonoma de Madrid (UAM). He has also done long placement abroad in some prestigious international researching centers, such as CERN, NeSC, NRC and the University of Melbourne. His primary research areas are high-performance and large-scale computing, data analysis and information visualization.