Real-Time Generation of Leg Animation for Walking-in-Place Techniques

Jingbo Zhao  
China Agricultural University, Beijing, China  
zhao.jingbo@cau.edu.cn

Yiqin Peng  
China Agricultural University, Beijing, China  
2018310060423@cau.edu.cn

Zhetao Wang  
China Agricultural University, Beijing, China  
2018308250211@cau.edu.cn

Yaojun Wang  
China Agricultural University, Beijing, China  
wangyaojun@cau.edu.cn

ABSTRACT
Generating forward-backward self-representation leg animation in virtual environments for walking-in-place (WIP) techniques is an underexplored research topic. A challenging aspect of the problem is to find an appropriate mapping from tracked vertical foot motion to natural cyclical movements of real walking. In this work, we present a kinematic approach based on animation rigging to generating real-time leg animation. Our method works by tracking vertical in-place foot movements of a user with a Kinect v2 sensor and mapping tracked foot height to inverse kinematics (IK) targets. These IK targets were aligned with an avatar’s feet to guide the virtual feet to perform cyclic walking motions. We conducted a user study to evaluate our approach. Results showed that the proposed method produced compelling forward-backward leg animation during walking. We show that the proposed technique can be easily integrated into existing WIP techniques.

CCS CONCEPTS  
• Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality;  
• Human-centered computing—Human computer interaction (HCI)—HCI design and evaluation methods—User studies;

KEYWORDS
Leg animation, walking-in-place, virtual locomotion

ACM Reference Format:
Jingbo Zhao, Zhetao Wang, Yiqin Peng, and Yaojun Wang. 2022. Real-Time Generation of Leg Animation for Walking-in-Place Techniques. In The 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI ’22), December 27–29, 2022, Guangzhou, China. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3574131.3574446

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

VRCAI ’22, December 27–29, 2022, Guangzhou, China  
© 2022 Association for Computing Machinery.  
ACM ISBN 978-8-4007-0031-6/22/12. . . $15.00  
https://doi.org/10.1145/3574131.3574446

1 INTRODUCTION
Estimating forward walking speed by monitoring in-place lower-body movements for walking-in-place (WIP) techniques has been a well-studied topic [Slater et al., 1995; Templeman et al., 1999; Feasel et al., 2008; Wendt et al., 2010; Williams et al., 2011; Bruno et al., 2013; Bruno et al., 2017; Tregillus and Folmer, 2016; Hanson et al., 2019]. However, few studies addressed the problem of visualizing and representing a user’s legs in virtual reality (VR) during in-place walking. Animating users’ legs during virtual walking enhances one’s sense of body ownership and presence in virtual environments [Lee et al., 2020]. It is desirable and beneficial to add leg animation to WIP techniques to give users a better VR locomotion experience.

In the physical space, we produce forward-backward reciprocating leg motions when we naturally walk. However, during in-place walking in VR, our feet move upward and downward. We wish to animate our legs in virtual environments such that the virtual leg movement patterns are similar to that of our natural walking.

Generating forward-backward leg animation for WIP techniques generally requires finding a set of mappings between tracked vertical leg movements and forward-backward leg movements in virtual environments. In the current study, we present a kinematic approach to render a user’s leg animation while a user is walking in place with a Kinect sensor that tracks a user’s foot height. Our main approach is to use IK targets, with a set of derived kinematic equations, to map a user’s tracked foot height to cyclic motions of the IK targets. These IK targets are aligned with the feet of a user’s avatar in virtual environments. Thus, cyclic motions of the IK targets guide an avatar’s feet to produce forward-backward leg animation. Since the derived kinematic equations are dependent on gait phases, we implement gait detection state machines to monitor the gait phases of a user’s feet.

The main contribution of the study is that we present a kinematic approach to generating real-time forward-backward leg animation for WIP techniques based on animation rigging using foot tracking data given by the Kinect sensor. Compared to previous methods [Lee et al., 2020; Freiwald et al., 2020; Park and Jang, 2019], the present approach does not need a cycling device and allows leg animation to match physical leg movements as gait phases are directly detected from leg movements. We show that this technique can be easily incorporated into existing WIP techniques that can detect gait phases to generate real-time leg animation during virtual walking. The present paper is an extended version of a previously published abstract [Zhao et al., 2022]. The present paper extends
the description of the proposed methods with more details, and includes the experiment that evaluated the proposed methods.

2 RELATED WORK

Only a very limited number of studies have attempted to solve this leg animation generation problem for WIP techniques. For example, previous work by [Lee et al., 2020] tackled this problem by using a 1-D convolutional neural network (CNN) to detect gait phases from the head tracking signals of a VR headset and using the detected gait phases as key-frames to animate pre-defined leg animation. A limitation of this approach is that the predicted gait phases may not be consistent with the actual gait phases of the users. Thus, the rendered leg animation may not match a user’s actual leg movements. [Freiwald et al., 2020] presented a locomotion interface called VR strider, which consisted of a mini exercise bike, tracking and vibrotactile feedback devices. When a user pedaled the mini exercise bike, tracked circular motions were mapped to the inverse kinematics (IK) targets aligned with the avatar’s feet. Leg animation was generated with the guidance of the IK targets. However, this approach requires users to have this modified mini exercise bike as the platform to use this locomotion approach. [Park and Jang, 2019] presented a WIP method that rendered leg animation based on the GUD-WIP technique [Wendt et al., 2010]. A step period was estimated from a user’s gait. The estimated step period was subsequently linearly mapped to the angles of the lower limb joints. However, this approach does not allow leg animation to match the motion of users as a step period needs to be estimated first before leg animation can be rendered. To the best knowledge of the authors, our work is the first that aims to solve this problem by using IK targets and kinematic equations driven by tracked foot height to generate compelling leg animation for WIP techniques. Compared to previous work, our method does not require specialized hardware and allows the avatar’s leg animation to match the leg motion of a user.

3 METHODS

3.1 Hardware and Software of the VR system

Our experimental VR hardware consisted of a Kinect v2 sensor to track a user’s body movements, an Oculus Quest 2 headset to present virtual environments, a computer with an Intel-i5 11400F CPU, 16 GB memory and a GeForce RTX 3060 graphics card with 12 GB graphics memory to host the experimental application software. Implementation of the proposed method and prototyping of the experimental virtual environments were completed using the Unity 2019.4. Animation rigging was supported by the Unity’s Animation Rigging version 0.2.7-preview package. The avatar model used in our experiment was selected from the Microsoft Rocketbox Avatar library [Gonzalez-Franco et al., 2020b]. The rate of the game logic was fixed as 30 Hz in sync with the sampling rate of the Kinect v2 sensor.

3.2 Mapping Foot Height to Positions of IK Targets

We used the ankle joints tracked by the Kinect v2 sensor as the reference to monitor a user’s foot height. The height of the floor plane was estimated by capturing the ankle joints for 3 s when a user stood still during the calibration procedure. The data were then averaged to get a reference height of the floor plane. For both feet, we buffered two data sequences of the y-component of the ankle joints with a length of thirty elements using two queues. This corresponded to 1 s of data as the sampling rate of the Kinect v2 sensor is 30 Hz. Vertical speed (z\(_{\text{left}}\) and z\(_{\text{right}}\)) and velocity (y\(_{\text{left}}\) and y\(_{\text{right}}\)) were calculated from the y-position data for further processing steps. We further smoothed the position data with two low-pass Butterworth filters of 4 Hz to suppress noise. By inspection, it was not necessary to smooth the speed and velocity data. Velocity data were used for gait phase transitions and speed data were used for calculating the average speed between transitions of gait cycles to estimate forward walking speed [Bruno et al., 2013]. The tracked height values of both feet are denoted as h\(_{\text{left}}\) and h\(_{\text{right}}\), which are the most recent values from the smoothed data sequences.

We created two two-bone IKs using the recent Animation Rigging package from the Unity and respectively assigned the knee joints and the foot joints to IK hints and IK targets for both legs. According to the documentation of the Animation Rigging package, IK hints are optional, and they control the bend normal of the middle nodes of the two-bone IKs. Thus, IK hints were positioned by trial and error until desirable leg animation could be produced. They were finally placed 0.29 m in front of the avatar’s knees in depth. Other transform parameters of IK hints were aligned with the knee joints. The transforms of the IK targets were fully aligned with the avatar’s foot joints. This completes the setup for the rest of the calculations and discussion.

Assume that an avatar is producing forward-backward walking animation in place without moving forward. The step length and step height that an avatar can achieve for both feet are denoted by z\(_{\text{left}}\), z\(_{\text{right}}\), y\(_{\text{left}}\) and y\(_{\text{right}}\). These values were mapped to the IK targets in their local coordinate to guide the forward-backward movement of an avatar’s feet. The mapping is symmetrical about the avatar’s trunk in depth. Thus, z\(_{\text{left}}\) and z\(_{\text{right}}\) can be negative or positive. The horizontal values of the IK targets along the x-axis were fixed.

For analyses of the IK targets’ movements, we expand our discussion into three cases: a user’s first step, consecutive walking steps and the stopping step. In addition, we consider three gait phases: the double support phase, the ascending phase, and the descending phase, for updating the leg animation. Other gait phases, including the initial swing, the mid-swing, and the terminal swing [Rose and Gamble, 2006], are transitions of the gait detection state machine triggered by velocity and position conditions. Leg animation is not updated in these transitions. The kinematic equations derived in this study came from our inspection on recorded treadmill foot position data and direct observation on human walking. Since a walking cycle consists of three different gait phases and it is also necessary to process the first step, consecutive steps and the stopping step, using a single equation to fit recorded foot position data obtained from treadmill walking is not feasible. Thus equations 1-4 have been derived as piecewise functions to produce foot trajectories in depth, based on a user’s tracked foot height.

3.2.1 First Step. Initially, the avatar stands still in an initial double support phase with both feet centered on the origin in their local
coordinate. The step length and step height for both feet are zeros.
The user also starts with a double support phase (Figure 1a). Assume
that the user lifts their left foot first (Figure 1b) and ascends to
the mid-swing phase. This requires us to map the left stride of
the avatar forward in depth and the right stride backward during
the left foot’s ascending phase (denoted by ↑ in the equations for
compact representation). Once we detect the mid-swing, the left
foot enters the descending phase (denoted by ↓) and plants on the
ground subsequently. We describe this process using two piece-
wise functions for ascending and descending phases of both feet
for an avatar:

\[
\begin{align*}
    z_{\text{left}} &= \begin{cases} 
        -0.25z_{\max} \cos \left(2\pi h_{\text{left}} / H \right) + 0.25z_{\max}, & \uparrow \\
        0.25z_{\max} \cos \left(2\pi h_{\text{left}} / H \right) + 0.25z_{\max}, & \downarrow 
    \end{cases} \\
    z_{\text{right}} &= \begin{cases} 
        0.25z_{\max} \cos \left(2\pi h_{\text{left}} / H \right) - 0.25z_{\max}, & \uparrow \\
        -0.25z_{\max} \cos \left(2\pi h_{\text{left}} / H \right) - 0.25z_{\max}, & \downarrow 
    \end{cases}
\end{align*}
\]

where \( H \) is a scale factor that enables the function to complete
a gait cycle and \( H = 4y_{\max} = 1.2 \text{ m}. \) \( z_{\max} \) \((z_{\max} = 0.6 \text{ m})\) is the maximum step length that an avatar can produce and
\( y_{\max} \) \((y_{\max} = 0.3 \text{ m})\) is the maximum step height of a user. See
Figure 2a and Figure 2b for the plotted curves of equations 1 and 2.

When the avatar finishes the very first step, the avatar goes into
double support with the left stride and right stride symmetrical
about the trunk. From this gait phase, we discuss the animation
mapping of consecutive walking steps.

3.2.2 Consecutive Walking Steps. Subsequently, the user supports
the trunk using the left foot. The right foot (opposite foot) is lifted
(Figure 1c) and ascends to the mid-swing and then descends to foot
support again. Since the mapped avatar leg motion is symmetrical
about the trunk, we describe this process using another piece-wise

\[
z_{\text{right}} = \begin{cases} 
    -0.5z_{\max} \cos \left(2\pi h_{\text{right}} / H \right), & \uparrow \\
    0.5z_{\max} \cos \left(2\pi h_{\text{right}} / H \right) + 0.25z_{\max} \sin \left(4\pi h_{\text{right}} / H \right), & \downarrow 
\end{cases}
\]

In equation 3, a correction term \( 0.25z_{\max} \sin \left(4\pi h_{\text{right}} / H \right) \) is
added to enable the leading foot to strike the ground more nat-
urally during the terminal swing.

When the right foot of the avatars is mapped to move forward,
the left foot needs to be retracted and trailed behind the trunk, and
this can be described by:

\[
z_{\text{left}} = \begin{cases} 
    0.5z_{\max} \cos \left(2\pi h_{\text{right}} / H \right), & \uparrow \\
    -0.5z_{\max} \cos \left(2\pi h_{\text{right}} / H \right), & \downarrow 
\end{cases}
\]

Note that \( h_{\text{right}} \) is used in equations 3 and 4 as the animation of
this second step is controlled by the user’s right foot. Equations 3
and 4 hold for the next consecutive step (i.e. left foot) by swapping
the symbols representing the parameters of a user’s left and right
feet. See Figure 2c and Figure 2d for the plotted curves of equations
3 and 4.

The equations described above were implemented using two gait
detection state machines (discussed in Section 3.4) to respectively
monitor the gait phases of a user’s left foot and right foot and control
their respective IK targets. Motions of IK targets are smoothed using
the lerp() function from the Unity engine.

3.2.3 Stopping Step. We consecutively monitor the elapsed time \( t \)
in the double support phase of the user (Figure 1d). Whenever \( t \)
is larger than an empirical value of 1.5 s. We consider that the user
intends to stop and we set \( z_{\text{left}} \) and \( z_{\text{right}} \) to zeros using the lerp() function
to move the avatar’s legs to the initial standing state with
double support.

Figure 1: Illustration of the user gait phases for the movement mapping analyses in Section 3. Here, an avatar model is used
to illustrate the foot movements of the user during in-place walking. The present model was selected from the Microsoft
Rocketbox Avatar Library.
3.3 Manipulation of an Avatar’s Vertical Position

When we naturally walk on the ground, our body moves up and down. If our left and right strides reach the maximum position in depth (double support), our height will be much lower compared to normal standing. For avatar walking animation, such height change during walking also needs to be considered. We derive an equation that relates the avatar’s vertical position to the step length of the trailing foot (left foot in our analysis):

\[
\Delta h = l_{\text{leg}} - \sqrt{l_{\text{leg}}^2 - z_{\text{left}}^2}
\]  

where \(\Delta h\) is the amount of the adjustment for the avatar’s vertical position, \(l_{\text{leg}}\) the leg length of the avatar (\(l_{\text{leg}} = 0.8\) m in our selected model for analysis), manually determined from the avatar model parameters, and \(z_{\text{left}}\) the z-position of the trailing foot assume that the right foot is the leading foot.

The vertical position of the avatar during walking is updated as:

\[
y_{\text{curr}} = y_{\text{init}} - \Delta h
\]

where \(y_{\text{init}}\) is the initial vertical position before the avatar starts walking, and \(y_{\text{curr}}\) the updated vertical position value assigned to the avatar.

The vertical position of the avatar when they stop is updated as:

\[
y_{\text{curr}} = y_{\text{init}}
\]

Finally, for both cases: the first step and consecutive walking steps, mapping the vertical foot height \(y_{\text{left}}\) and \(y_{\text{right}}\) for the avatar is trivial. Assume that the user’s left foot ascends and then descends in motion and the right foot supports the trunk during the process, the mapping can be derived as:

\[
\begin{align*}
y_{\text{left}} &= 0.5y_{\text{left}} + \Delta h, \quad \text{↑ and ↓} \\
y_{\text{right}} &= \Delta h, \quad \text{↑ and ↓}
\end{align*}
\]

for both ascending and descending phases. We comment that a scale factor of 0.5 is an empirical value. Further investigation is required to determine if a scale factor of 0.5 or other value is appropriate for vertical mapping. Parameter \(\Delta h\) adjusts the vertical position considering the height changes of the avatar body during walking.
3.4 Gait Phases Detection and Walking Speed Estimation

We selected the WIP technique by [Bruno et al., 2013] and integrated our animation method into this technique to implement our method. To monitor the gait phases, we implemented two state machines to monitor the gait phases of two feet of a user. The original implementation of the Bruno et al.’s work [Bruno et al., 2013] was based on an optical tracking system with higher precision and a faster sampling rate than the Kinect v2 sensor. We empirically adjusted the parameters of the state machines to enable them to monitor gait phases. Walking speed estimation and the stopping criterion for producing zero speed were based on the work by [Bruno et al., 2017]. The readers are referred to the speed estimation model in their work for walking speed estimation and the criterion for determining walking stop. As in their work, we only adapted the walking speed whenever the mid-swing and the terminal swing were detected. The estimated walking speed was used to set the forward moving speed of the avatar along the z-axis. The required time threshold to stop walking can be calculated using the stop time estimation equation in Bruno et al.’s work [Bruno et al., 2017]. However, the determination of stopping to move the avatar’s leg back to the neutral position in double support required an empirical constant time threshold of 1.5 s to avoid false positive detection of walking stop.

In Figure 3, we present a state diagram to summarize the relationship between the gait phases and the derived equations assuming the left foot is lifted first. Applying the state diagram to the right (opposite) foot only requires changing the parameter symbols from the left foot to the right foot (i.e. \( h_{\text{left}} \rightarrow h_{\text{right}} \) and \( v_{\text{left}} \rightarrow v_{\text{right}} \), etc.). Values of position and velocity thresholds to determine state transition are also given in Figure 3.

While in the current study we selected the WIP technique by Bruno et al. [Bruno et al., 2013; Bruno et al., 2017] and integrated our leg animation approach into this WIP technique to test our method, our method also can be incorporated into other WIP techniques, such as the GUD-WIP [Wendt et al., 2010], that have gait phases detection mechanisms. Since our method only relies on users’ tracked foot height to generate foot trajectories, it can also work with other sensor systems that can track users’ foot height.

3.5 Key-frame Leg Animation for Experimental Comparison

We designed two key-frame leg animations for walking and standing postures using the animation rigging tool to compare our proposed method. The same gait detection state machine (described in Section 3.4) was used to detect whether a user’s feet are moving or not. The real-time IK targets trajectory generation equations were removed. Whenever we detected that a user’s feet ascended or descended, the walking animation was played. When the user stopped, the standing animation was played. The key-frame animation was not synchronized with leg movements. Leg animation was played when ascending and descending phases were detected by state machines.
Figure 5: First-person view during walking. Participants were asked to look down and observe their virtual legs to evaluate animation results.

4 EXPERIMENT

4.1 Introduction

The goal of the experiment was to evaluate whether our proposed method can generate compelling leg animation when users walk in virtual environments and understand difference between two animation conditions: the key-frame animation and the real-time animation.

4.2 Participants

We invited 12 volunteer undergraduate students (8 males and 4 females, age: 20-22, height: 155 cm-183 cm) to participate in our experiment. All had normal or corrected-to-normal vision. Informed consents were provided to participants and completed before experiments.

4.3 Design

We designed a 3D environment that included a plane with a repeated grass texture representing the ground and a walking path with concrete squares texture for virtual walking. The trees placed along the walking path were selected from the Free SpeedTrees Package from the Unity Asset Store. An experimental session was designed to have two experimental conditions (blocks): (a) real-time leg animation generated using the method described in Section 3; (b) replay of key-frame leg animation generated using pre-defined key frames with the animation rigging tool.

For each block, there were one practice trial for participants to get familiar with the task and two experimental trials. A questionnaire was adapted from [Kim et al., 2021] based on a 7-point Likert scale (from strongly disagree to strongly agree) to ask participants to subjectively evaluate their leg animation. The questionnaire included the following statements: (1) Leg animation was smooth; (2) Leg animation was natural; (3) Leg animation was matched to my leg motion; (4) Leg animation was the same as I intended. For short notation, we refer to these factors as smoothness, naturalness, match and intention. Whenever a participant finished a complete block, they were asked to complete the questionnaire to assess their leg animation during walking. We controlled for order effects by dividing participants into two groups and counter-balancing the order of the blocks. We created a rigid body for the avatar used in the experiment and disabled its gravity option in the Unity. The estimated forward walking speed was assigned to the z-axis of the rigid body of the avatar to enable it to move forward. We limited the maximum forward walking speed to 3.5 m/s during experiments. Each experimental trial required a participant to walk 50 m to complete the trial. A collision volume is placed 50 m ahead of the avatar to determine the completion of the trial. The Oculus OVR camera rig was placed at the eye position of the avatar model for participants to view the virtual environment and their virtual body.

4.4 Procedure

We mounted the Kinect v2 sensor on a tripod approximately 60 cm high from the ground, with the camera view capturing a participant (see Figure 4 for the experimental setup). During an experimental session, we first introduced the experimental task to a participant and provided an informed consent for the participant to sign. The participant was then asked to stand approximately 1.8 m in front of the Kinect sensor wearing the Quest 2 headset. The Quest 2 headset was wirelessly connected to the host computer using the Airlink feature through a TP-Link Wi-Fi 6 router. Participants were asked to look down and observe their virtual legs during walking. The researcher then asked the participant to level their heads and reset the view so that the participant’s view could be aligned to the eye position of the avatar. Then, a 3-s calibration was conducted, capturing the participant’s foot position to determine the reference height of the floor plane for calculating foot height. After calibration was done and when the participant was ready, the researcher pressed the start button to initiate the trial with a 3-s countdown timer. When this timer was up and a “start” prompt was shown, the participant started walking until they reached the collision volume.

Figure 6: An exemplar sequence of screenshots of an avatar walking captured from a side view while a user was walking in place, with foot height tracked by a Kinect sensor.
to implement leg animation in a VR system, it should be reasonable to use this new technique as it is easy to implement and enables better matching between physical leg motions and rendered leg animation.

5 DISCUSSION

One limitation with the proposed approach is that the maximum step length and height are fixed, so it is impossible to generate leg animation with variable step length and height based on one’s tracked foot height. Future work should focus on deriving more flexible mappings to enable leg animation to have variable step length and step height. The Quest Pro headset has been recently released by Meta and it has been shown that the device is capable of tracking motions of a user’s body parts without using additional hardware or trackers. This makes it possible to remove the use of an optical sensor, such as the Kinect sensor, to track body motions for the proposed leg animation technique. The proposed method can be adapted to this new device, making it accessible to general VR users, who often only possess a VR headset.

Further study of the proposed method includes comparing it to methods proposed by other researchers [Lee et al., 2020; Park and Jang, 2019], investigating its utility in more general virtual locomotion cases and comparing the effects of the present method and other methods on gait variability. In addition, [Waltemate et al., 2018] presented a method for generating personalized avatars for users in virtual environments. It will be interesting to leverage the method to create personalized avatars for individual users and investigate if such manipulation affects user experience on animation, body ownership and presence when animating users’ virtual legs during walking. Research by [Gonzalez-Franco et al., 2020a] has shown that users tend to follow the movements of the avatars and [Willaert, 2021] reported that users’ gait can be modulated by avatars. While setting up the key-frame leg animation experimental condition, we also observed that users may follow the unsynchronized leg motions under this condition. Further research is needed to study the influence of the self-avatar follow effect on gait parameters, presence, and body ownership when leg animation is not in sync with a user’s leg motion.

6 CONCLUSION

In this work, we presented a kinematic approach to generating forward-backward leg animation based on animation rigging, with foot height tracked by a Kinect v2 sensor. A set of kinematic equations were proposed to guide the legs of an avatar to generate leg animation. Our user study shows that the present method produces compelling leg animation and better matching between a user’s physical leg and an avatar’s leg than the keyframe based animation. Thus, this method is promising to be applied to practical VR WIP locomotion applications or experiences, where rendering of leg animation is needed.

REFERENCES

Bruno, L., Pereira, J., and Jorge, J. 2013. A New Approach to Walking in Place. INTER- ACT, 370–387.

Bruno, L., Sousa, M., Ferreira, A., Pereira, J.M., and Jorge, J. 2017. Hip-directed Walking-in-place using a Single Depth Camera. International Journal of Human-Computer Studies 105, 1–11.
Caine, K. 2016. Local Standards for Sample Size at CHI. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 981–992.

Feasel, J., Whitton, M.C., and Wendt, J.D. 2008. LLCM-WIP: Low-Latency Continuous-Motion Walking-in-Place. *2008 IEEE Symposium on 3D User Interfaces*, 97–104.

Fehnwald, J.P., Ariza, O., Janeh, O., and Steinicke, F. 2020. Walking by Cycling: A Novel In-Place Locomotion User Interface for Seated Virtual Reality Experiences. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, ACM, 1–12.

Gonzalez-Franco, M., Cohn, B., Ofek, E., Burin, D., and Maselli, A. 2020a. The Self-Avatar Follower Effect in Virtual Reality. *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 18–25.

Gonzalez-Franco, M., Ofek, E., Pan, Y., et al. 2020b. The Rocketbox Library and the Utility of Freely Available Rigged Avatars. *Frontiers in Virtual Reality* 1, 20.

Hanson, S., Paris, R.A., Adams, H.A., and Bodenheimer, B. 2019. Improving Walking in Place Methods with Individualization and Deep Networks. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 367–376.

Kim, H., Ali, G., Kim, S., Kim, G.J., and Hwang, J.-I. 2021. Auto-generating Virtual Human Behavior by Understanding User Contexts. *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 591–592.

Lee, J., Lee, M., Kim, G.J., and Hwang, J.-I. 2020. Effects of Synchronized Leg Motion in Walk-in-Place Utilizing Deep Neural Networks for Enhanced Body Ownership and Sense of Presence in VR. *26th ACM Symposium on Virtual Reality Software and Technology*, ACM, 1–10.

Park, C. and Jang, K. 2019. Investigation of Visual Self-Representation for a Walking-in-Place Navigation System in Virtual Reality. *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 1114–1115.

Rose, J. and Gamble, J.G., eds. 2006. *Human walking*. Lippincott Williams & Wilkins, Philadelphia.

Slater, M., Steed, A., and Usoh, M. 1995. The Virtual Treadmill: A Naturalistic Metaphor for Navigation in Immersive Virtual Environments. *Virtual Environments*, 135–148.

Templeman, J.N., Denbrook, P.S., and Sibert, L.E. 1999. Virtual Locomotion: Walking in Place through Virtual Environments. *Presence* 8, 6, 598–617.

Tregillus, S. and Folmer, E. 2016. VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1250–1255.

Waldemate, T., Gall, D., Roth, D., Botsch, M., and Latoschik, M.E. 2018. The Impact of Avatar Personalization and Immersion on Virtual Body Ownership, Presence, and Emotional Response. *IEEE Transactions on Visualization and Computer Graphics* 24, 4, 1643–1652.

Wendt, J.D., Whitton, M.C., and Brooks, F.P. 2010. GUD WIP: Gait Understanding-Driven Walking-In-Place. *Proceedings of the IEEE Virtual Reality Conference*, 51–58.

Willaert, I. 2021. [DC] The Effect of Modulating The Step Length of an Embodied Self-Avatars on Gait Symmetry During Treadmill Walking. *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 729–730.

Williams, B., Bailey, S., Narasimham, G., Li, M., and Bodenheimer, B. 2011. Evaluation of Walking in Place on a Wii Balance Board to Explore a Virtual Environment. *ACM Trans. Appl. Percept.* 8, 3, 19:1-19:14.

Zhao, J., Wang, Z., Peng, Y., and Wang, Y. 2022. Generating Leg Animation for Walking-in-Place Techniques using a Kinect Sensor. *28th ACM Symposium on Virtual Reality Software and Technology (VRST ’22)*.