

Real-Time Step Detection Using the Integrated Sensors of a Head-Mounted Display

Polona Caserman*, Patrick Krabbe[†] *Member, IEEE*,
Janis Wojtus[‡] *Member, IEEE*, Oskar von Stryk[‡] *Member, IEEE*

Simulation, Systems Optimization and Robotics Group, Technische Universität Darmstadt, Germany

*polona.caserman@stud.tu-darmstadt.de

[†]patrickhubert.krabbe@stud.tu-darmstadt.de

[‡]<wojtusch|stryk>@sim.tu-darmstadt.de

Abstract—Recent improvements in virtual reality technology and head-mounted displays have led to a number of novel and innovative applications in entertainment, education, science and healthcare. The primary goal in most of these applications is to give the user a sensation of being part of the virtual reality. This focus on presence or immersion requires to create a connection between the user and the virtual environment but also between the user and his virtual avatar. Synchronizing the body movement of the user and his avatar can help to improve the feeling of presence by enhancing the experience of agency and body ownership over the avatar. A typical example is the combination of a head-mounted display and a treadmill to create a realistic walking or running simulation in a virtual environment. In this scenario, the synchronization of leg movement improves the feeling of presence and allows to further enhance the virtual experience, for example by adjusting gait parameters of the avatar or triggering customized stepping sounds.

This paper presents a robust real-time step detector that uses the integrated sensors of a state-of-the-art head-mounted display and allows to recognize the pattern of individual steps. No additional sensors on the trunk or lower body are required. By applying a coordinate transformation and straightforward signal processing, it is possible to discriminate between left and right steps and detect the current walking speed. This information is used to animate a virtual avatar by scaling a predefined walking trajectory and to control the walking speed within a virtual environment.

Index Terms—Step Detection, Virtual Reality, Head-Mounted Display, Gaussian Filter

I. INTRODUCTION

Virtual reality (VR) increases user performance via immersion, giving an enhanced sense of “being there” [1], [2]. The feeling of immersion allows the sense of belief that the user is “present” in the virtual environment [3]. To improve the immersion when using a head-mounted display (HMD), it is crucial that the body movements correspond to the changes in the depicted VR scene [4]. This can be done by tracking the body movements of the user and synchronizing the avatar’s humanoid body to improve the feeling of presence by enhancing the experience of agency and body ownership over the avatar [5]. By combining a HMD and a treadmill, realistic walking or running in a virtual environment can be simulated. A typical application example is a rehabilitation training, e.g., [6], [7]. Moreover, VR applications are generally

designed to be more interesting and enjoyable than traditional therapy, thereby encouraging higher numbers of repetitions [8]. Compared to normal treadmill training, a treadmill training in combination with VR shows a more stimulating experience which can be increased with real-time feedback [9].

Development of VR systems has already been conducted during the 1950s and 1960s within NASA and US Air Force research programs [10]. VR was used for many purposes that would be dangerous or impractical to do in reality, such as pilot and military training as well as surgical procedure training. The breakthrough came in the 1990s. Cruz-Neira and Sandin et al. [11] developed a projection-based VR system called CAVE (CAVE Automatic Virtual Environment). This immersive VR facility was designed for exploration of and interaction with spatially engaging environments. It consists of three rear-projection screens for walls and a down-projection screen for the floor. Nowadays, VR can be experienced wearing novel HMDs, which allows deeper immersion since the display is mounted directly in front of the eyes. During last years, the development in VR and HMD technology turned out to play an increasing role. Among other areas, great progress was made in improving HMD screen resolutions. One prominent example is the Oculus Rift¹. The Development Kit 1 presented in 2013 featured a resolution of 1280×800 pixels. Development Kit 2 introduced in 2014 already provided an high-definition resolution of 1920×1080 pixels. The Consumer Version 1 released in 2016 has an even higher resolution of 2160×1200 pixels. Furthermore, other HMDs have been presented recently, e.g., Sony PlayStation VR², Samsung Gear VR³ and HTC Vive⁴. With systems like Google Cardboard⁵, a customary smartphone can be used to experience VR in a simple and affordable way. Due to the HMD development, the number of novel and innovative VR applications and simulations in entertainment, education, science and healthcare has been increased, e.g., [6], [7], [12], [13]. However, the

¹<https://www.oculus.com/en-us/rift/>

²<https://www.playstation.com/en-us/explore/playstation-vr/>

³<http://www.samsung.com/global/galaxy/wearables/gear-vr/>

⁴<https://www.htcvive.com/>

⁵<https://www.google.com/get/cardboard/>



(a) Subject walking on the treadmill and wearing a HMD. (b) Avatar in the VR environment with synchronized steps.

Fig. 1: Combination of a VR environment with a treadmill.

combination of VR with a treadmill and a real-time step detection is not well explored yet.

In the past years several authors have introduced real-time step detection in order to support users in various ways. Ying et al. [14] proposed algorithms using an accelerometer for automatic step detection that can be used for neurological rehabilitation research. An accelerometer sensor has to be fastened on a foot in order to detect whether the foot lifts off and the heel strikes the ground. Jasiewicz et al. [15] also introduced different methods for gait detection using miniature foot-mounted linear accelerometers. In their study they demonstrated that either linear acceleration or angular velocity sensors can be used to detect gait events. Yang et al. [16] presented the development of a wearable accelerometer system for real-time gait recognition. The wearable motion detector is a single waist-mounted device that is able to measure trunk accelerations while walking. With an autocorrelation procedure several gait cycle parameters, such as cadence, step regularity, stride regularity, and step symmetry can be estimated in real-time.

These approaches use body-mounted accelerometers which are fastened not far from the foot or the waist for monitoring and are capable of gait detection during walking. Encouraged by these promising results, a new and robust step detector using only integrated sensors of a HMD is proposed in this paper. With this step detector, the motion of an avatar can be synchronized with the actual gait of the user. Thereby, the person exploring VR would perceive the humanoid avatar body and movements are more realistic while looking down at the virtual body. This synchronization greatly supports immersive VR [4], [5]. The applied experimental setup with a subject on a treadmill and synchronized avatar gait is depicted in Figure 1.

The rest of this paper is structured as follows. In Section II, four different algorithms for step detection using the integrated sensors of a HMD are described. An experimental evaluation of the presented approaches is given in Section III. A discus-

sion of the results and a conclusion follow in Section IV.

II. APPROACH

In this section, four different approaches for real-time step detection algorithms using the integrated sensors of an Oculus Rift Development Kit 2 are presented in detail and explained on an example dataset of measured sensor data. Manually marked steps in the example dataset are applied as reference points. The Development Kit 2 has an integrated inertial measurement unit with a three-dimensional accelerometer, gyroscope and magnetometer and a camera-based position tracker. Integrated sensor fusion and filtering algorithms provide smooth measurements of linear and rotational position, velocity and acceleration in space with an update rate of 75 Hz. The virtual environment is modeled by applying the game engine Unity5⁶.

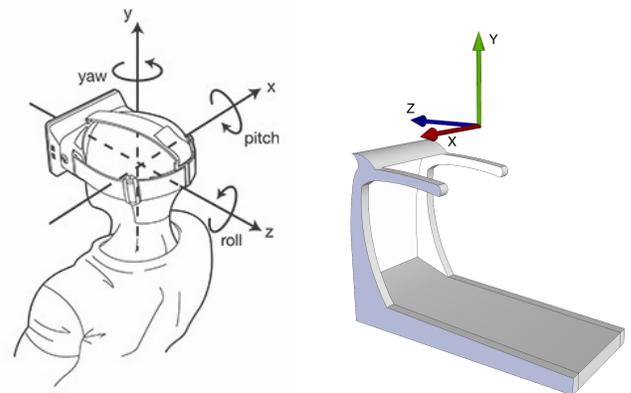
A. Overview

The sensor measurements are provided in a head-fixed reference frame (RF) as shown in Figure 2a. The x-axis points to the right, the y-axis points upwards and the z-axis is normal to the plane spanned by the other axes and points backwards. The origin is the center of the head. Before using the measured values, they have to be transformed to the fixed world reference frame as shown in Figure 2b. Let $\mathbf{q}[t]$ be a quaternion that represents the head rotation at time step t with respect to a fixed world RF. The rotation matrix is calculated using the inverse rotation $\mathbf{q}[t]^{-1}$:

$$\mathbf{T}[t] = \text{Trans}(0, 0, 0) \cdot \text{Rot}(\mathbf{q}[t]^{-1}) \cdot \text{Scale}(1, 1, 1).$$

To get the transformed sensor measurements $\mathbf{a}_{trans}[t]$, a raw data vector $\mathbf{a}_{raw}[t]$ has to be multiplied with the rotation matrix, $\mathbf{a}_{trans}[t] = \mathbf{T}[t] \cdot \mathbf{a}_{raw}[t]$. The head-fixed RF is fixed to the initial position of the user's head towards the treadmill, i.e. it is relative to the treadmill. The design of the treadmill

⁶<https://www.unity3d.com/5/>



(a) Head-fixed HMD reference frame [17]. (b) Treadmill-fixed world reference frame.

Fig. 2: Relation between head-fixed HMD and world-fixed treadmill reference frames.

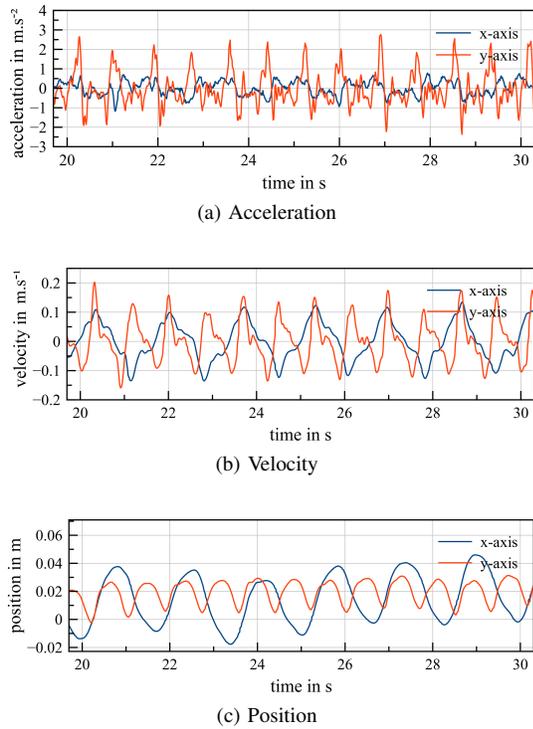


Fig. 3: Measurement signals for walking at 3 km.h^{-1}

constrains the user's torso to a forward-oriented pose. When the user rotates or translates the head, the world RF stays the same. This is beneficial as the world RF always stays the same relative to the user's walking direction. In summary, the transformation of the sensor measurements from the head-fixed RF to the world RF allows to express them in a RF that is natural to the user's body orientation and walking direction.

Four different step detection algorithms have been developed using linear acceleration, linear and rotational velocity and linear position measurements, respectively. Figure 3 shows the different linear measurement signals of the transformed x- and y-axes for walking at 3 km.h^{-1} . While the acceleration measurements are quite noisy, the velocity and position mea-

surements are much smoother.

B. Step Detection Algorithm using Acceleration Signals

For each step, a positive peak of the acceleration signal in the y-axis occurs when the foot is hitting the ground or more specifically when the heel strikes. Furthermore, this positive peak in the y-axis appears alternately coincident with a positive and negative deflection of the acceleration signal in the x-axis. In most cases, a positive deflection indicates a right step and a negative deflection implies a left step. This behavior results from the natural head motion during walking and running. Figure 4a shows two right steps and one left step where each heel strike event is manually marked with a cross.

The algorithm applies an adaptive threshold for peak searching and an adaptive side decision. In each time frame, the rotated acceleration vector is calculated. The transformed acceleration signal in the y-axis has to be greater than an adaptive step threshold in order to detect a step. The heel strike event is defined as the positive peak while the toe off event is defined as the following local peak. The step threshold is adapted by calculating the moving average of the 32 most recent average of the measurement values for toe off and heel strike events.

When a step is detected, the algorithm checks if the current step is more likely to be a right or a left step. Therefore, the acceleration signal in x-axis signal is interpreted. The current measurement value is compared with the interquartile mean (IQM) of all detected steps in the current trial. The IQM is computed discarding the upper 25% and lower 25% of the measurements and calculating the average of the remaining values. This approach is quite robust against deviations. Figure 5 illustrates a flow chart that explains the algorithm in schematic form.

C. Step Detection Algorithm using Velocity Signals

As the linear velocity is the integral of the linear acceleration, the graph of the velocity measurement signal shown in Figure 4b is much smoother. One can see that the zero-crossings for the velocity signal in the y-axis roughly coincide with the heel strikes marked with crosses and the positive

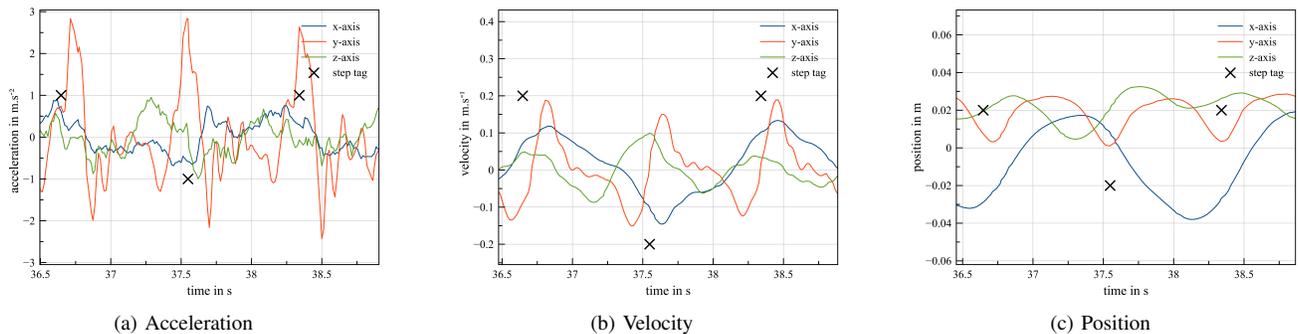


Fig. 4: Measurement signals for three steps of walking at 3 km.h^{-1} with manually marked step tags "x". A positive step tag stands for a right and a negative step tag stands for a left step.

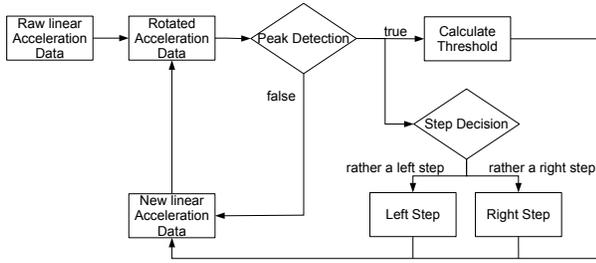


Fig. 5: Flow chart for acceleration-based step detection.

peaks in the z-axis. The peaks in the y-axis coincide with the extrema in the x-axis. The general pattern of the linear acceleration is repeated in the linear velocity, the base frequencies of the signals in the y-axis and z-axis are half the frequency of the signal in the x-axis.

The general concept of the algorithm is signal binarization using hysteresis in conjunction with a finite state machine. The flow chart for this algorithm is presented in Figure 6. The linear velocity is influenced by angular velocity as the rotational center of the head rotations seldom lies within the origin of the head-fixed RF. The velocity consists of a linear and a rotational part, hence $\mathbf{v} = \mathbf{v}_{lin} + \mathbf{v}_{rot}$ which is equivalent to $\mathbf{v}_{lin} = \mathbf{v} - \mathbf{v}_{rot}$, where $\mathbf{v}_{rot} = \mathbf{R} \cdot \boldsymbol{\omega}_{raw}$. The factors $\mathbf{R} = \text{antidiag}(r_x, r_y)$ represent the offset of the rotational center from the head-fixed RF origin which were found to be $(-0.15, 0.08)$ on average in the measurements. The entire

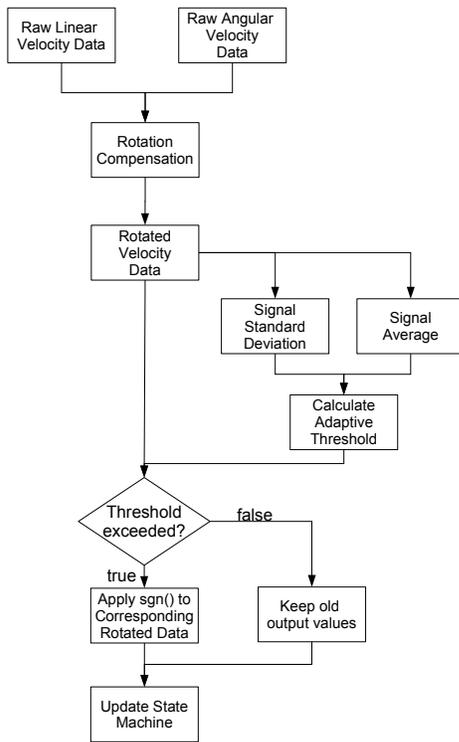


Fig. 6: Flow chart for velocity-based step detection.

calculation is given by

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}_{raw} + \begin{bmatrix} 0 & r_y \\ r_x & 0 \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}_{raw}$$

Although these constant factors cannot completely eliminate the effects of head rotation, they do significantly mitigate them.

These rotation-compensated measurements are then transformed into the world RF and fed into buffers with a capacity of 150 measurements. The buffers separately store x- and y-axis signals with positive and negative sign, i.e. four buffers in total. This way, the step detector also works for asymmetric gaits which may be prevalent in pathological gait.

For each of these buffers, the average and standard deviation is calculated. The adaptive threshold is calculated for each of the signals. The positive threshold is the sum of the positive buffer mean added by one standard deviation, the negative threshold is the buffer mean subtracted by one standard deviation. These thresholds are used to binarize the signals. If the current input signal exceeds the upper threshold, the binarized variable is set to +1 and if it is beneath the lower threshold, it is set to -1. If none of the previous conditions apply, the binarized variable is kept at its old value. The binarized signals are then used to update the state machine shown in Figure 7. The transition 2-tuple is of the form (X, Y) . Valid transitions are marked with arrows. In order to assure validity of the currently detected step phase, the current state of the state machine is only used, if four consecutive transitions were valid.

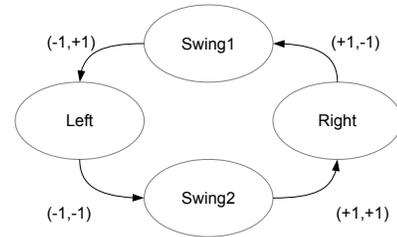


Fig. 7: State machine for velocity-based step detection.

D. Step Detection Algorithm using Position Signals

The position signal is the integral of the linear velocity signal and it shows a much smoother behaviour as illustrated in Figure 4c. Here, the minima of the signal in the y-axis coincide with heel strikes and the position in the x-axis indicates whether it is the left or right heel.

For this algorithm, a simple peak detection algorithm is used to determine the step state. The algorithm checks, if a local minimum occurred in the signal in the y-axis. If not, no further computations are performed. If there is a local minimum, it is checked whether the last extremum of the x-axis signal was a maximum or a minimum. If it was a maximum, the step is classified as a left step otherwise as a right step. The corresponding flowchart is shown in Figure 8.

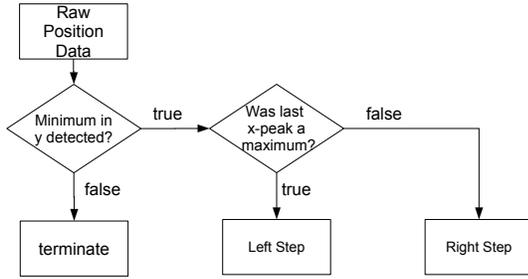


Fig. 8: Flow chart for position-based step detection.

E. Step Detection Algorithm using Gaussian Low-Pass Filter

This algorithm uses a low-pass filtered signal with a Gaussian window of constant width. The desirable properties of this filter are the rejection of high-frequency noise and introduction of a constant delay that can be accounted for. This filter is applied to the integrated sensor measurements of the HMD including acceleration, velocity and position.

The first step in this approach is to filter the transformed measurement signals of linear acceleration and velocity as well as position with a Gaussian window. The Gaussian window is computed with

$$w[n] = e^{-\frac{1}{2} \left(\frac{n}{\sigma_{\text{samples}}} \right)^2},$$

where $n \in [-\frac{l}{2}, \frac{l}{2}]$ denotes the current sample index in the window. To compute the filter length l , let $\sigma = 0.1\text{s}$ and $\sigma_{\text{samples}} = \frac{\sigma}{\Delta T}$. Let σ_{breadth} represent how wide the window should be, e.g., $\sigma_{\text{breadth}} = 6$. Now the filter length can easily be computed with $l = \lceil \sigma_{\text{breadth}} \cdot \sigma_{\text{samples}} \rceil$. To apply the Gaussian window, the scalar product of the raw sensor measurements and the Gaussian window has to be computed. Thereby, the filtered sensor measurements are obtained. In a second step, a local extremum can be determined from the filtered signals in the x- and y-axes. When either a local minimum or a local maximum is detected, the distance between these two extrema is computed. The next heel strike is predicted with the average distance. As the filtered sensor measurements are delayed, the window delay has to be subtracted from the predicted time of the next heel strike. In a last step, the decision whether it is a left or right one is done. If the last observed extremum was a local maximum, the predicted step is a left step. Otherwise, the predicted step is a right step. Figure 9 illustrates a flow chart that explains the algorithm in schematic form.

III. EVALUATION

In order to assess the accuracy of the four presented approaches, the outputs obtained from the step detectors and manually labeled events are compared. The correctly recognized steps are counted and compared with the actual recognized steps. Two subjects participated in the trials. The subjects walked on the treadmill at 1 km.h^{-1} , 3 km.h^{-1} , 5 km.h^{-1} and jogging at 7 km.h^{-1} while looking straight and at 3 km.h^{-1} with intentional head movements. Six trials per

walking test scenario and two trials per jogging test scenario were performed. The results are presented in Figure 10.

The position-based step detector provided best results at all walking speeds, even when the subject was looking around, correctly detecting more than 90% of the steps in most trials. When the subject was jogging at 7 km.h^{-1} , however, the position-based step detector shows only poor results. The Gaussian filtered step detector, while performing acceptably, did not match the position-based step detector performance in any of the trials. The accuracy of the worst and best trial lies between 50% and 98.7%. The acceleration-based step detector performed similarly to the Gaussian filtered step detector at 3 km.h^{-1} and 5 km.h^{-1} with the accuracy between 47.3% and 92.9%, but worse at the other speeds. The velocity-based step detector showed mixed results. While jogging it provided best results with the accuracy of 65%. At the speed of 1 km.h^{-1} it was second best with the result of 94.1%. At all other speeds it performed inferior to the other detectors.

None of the step detector approaches show satisfactory results when the gait was changed to jogging as the movement pattern does not fit the detector data model. Therefore, the developed step detector approaches are not applicable for jogging movements in the current form. The velocity-based detector was the only one that did not suffer significantly while jogging, though it performed on a low level. For all speeds in walking gait, the position-based step detector performed best. For jogging the velocity-based step detector performed best. As different detectors performed best in the different gaits, it should be considered to detect the current gait and change the detector type accordingly.

Although many steps are not recognized properly or are overlooked, the animation synchronization works sufficiently well in many scenarios. For the synchronization it is not crucial if some steps are overlooked. When no new steps are detected, the animation will simply continue to play the step motion. It is much more significant that false positives are detected in order to interpolate the animation smoothly. In the case of false positives, it looks for the user as if he is stumbling.

The main reason for false recognized or overlooked steps is due to the interdependence between head rotation represented by angular velocity and the linear sensor measurements. With-

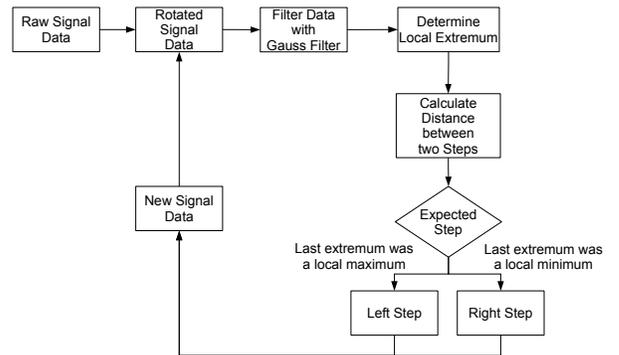


Fig. 9: Flow chart for Gaussian filtered step detection.

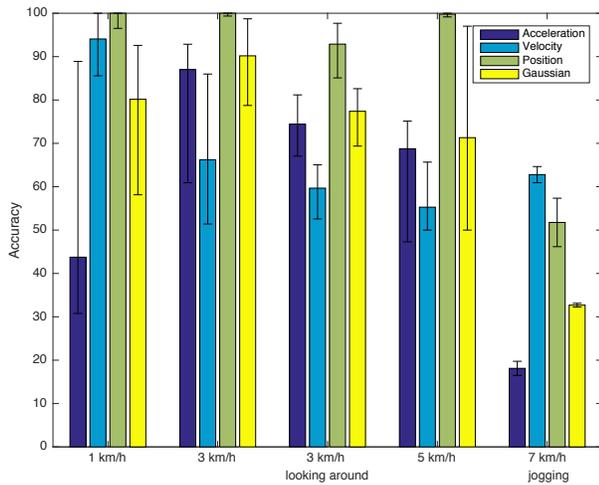


Fig. 10: Accuracy of the step detection algorithms in various scenarios. The error bars indicate the accuracy in the best and worst trials.

out head rotation, the error of all step detector algorithms is reduced noticeably. Improved versions of the algorithms should consider the angular velocity. Since the step detector works very well when looking forward, it can be used to synchronize stepping sounds with the heel strikes to further improve the experience for the user.

IV. CONCLUSION

This research examined the development of four different algorithms for real-time step detection using the integrated sensors of a HMD. Based on a brief analysis of the related work as well as the observed sensor data, it could be ascertained that gait detection is realizable using current widely becoming available head-mounted sensors. The developed step detection algorithms can be used for applications in entertainment, education, science and healthcare. Correctly identified steps can be used to animate and synchronize a virtual avatar and to integrate other effects like stepping sounds. This further improves the experience of agency and body ownership over the avatar while moving in a virtual environment.

The acceleration-based step detection algorithm occasionally detects steps where actually no steps occur, while the velocity-based step detection algorithm overlooks many steps especially if the user is looking around. Even though the Gaussian filtered step detection was designed with regard to robustness against disturbances, such as head movements, it does not surpass the accuracy of the position-based step detection algorithm. Thus, the position-based step detection algorithm performs best with a median accuracy of over 92% across all considered test scenarios. As outlined before, the design of the position-based step detector is very simple and easy to implement. This also implies a good performance regarding computation time.

To make the step detection even more robust and reliable, the interdependence of angular velocity and linear sensor

measurements needs to be considered. Since the velocity-based algorithm outperformed the position-based algorithm for the jogging motion, it might be advisable to switch between different approaches by detecting the current type of gait.

ACKNOWLEDGMENT

The authors would like to thank the involved psychology students from Technische Universität Darmstadt for the good collaboration and for providing research results which were incorporated in the development of the virtual environment.

REFERENCES

- [1] C. Cruz-Neira, J. Leigh, M. Papka, C. Barnes, S. M. Cohen, S. Das, R. Engelmann, R. Hudson, T. Roy, L. Siegel, C. Vasilakis, T. A. DeFanti, and D. J. Sandin, "Scientists in Wonderland: A Report on Visualization Applications in the CAVE Virtual Reality Environment," in *Proc. of the Sym. on Research Frontiers in Virtual Reality*, 1993, pp. 59–66.
- [2] R. Pausch, D. Proffitt, and G. Williams, "Quantifying Immersion in Virtual Reality," in *Proc. of the Conf. on Computer Graphics and Interactive Techniques*, 1997, pp. 13–18.
- [3] "Immersion and Sense Of Presence," http://w3.uqo.ca/cyberpsy/en/pres_en.htm, [Online; last visited on April 26, 2016].
- [4] D. A. Bowman and L. F. Hodges, "User Interface Constraints for Immersive Virtual Environment Applications," Georgia Institute of Technology, Atlanta, Tech. Rep. 95-26, 1995.
- [5] W. Steptoe, A. Steed, and M. Slater, "Human Tails: Ownership and Control of Extended Humanoid Avatars," *IEEE Trans. on Visualization and Computer Graphics*, vol. 19, no. 4, pp. 583–590, 2013.
- [6] B. J. Darter and J. M. Wilken, "Gait Training With Virtual Reality–Based Real-Time Feedback: Improving Gait Performance Following Transfemoral Amputation," *Physical Therapy*, vol. 91, no. 9, pp. 1385–1394, 2011.
- [7] D. H. Gates, B. J. Darter, J. B. Dingwell, and J. M. Wilken, "Comparison of Walking Overground and in a Computer Assisted Rehabilitation Environment (CAREN) in Individuals with and without transtibial Amputation," *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, pp. 1–10, 2012.
- [8] K. E. Laver, S. George, S. Thomas, J. E. Deutsch, and M. Crotty, "Virtual Reality for Stroke Rehabilitation," in *Cochrane Database of Systematic Reviews*, 2011, pp. 308–318.
- [9] L. H. Sloot, M. M. van der Krogt, and J. Harlaar, "Effects of Adding a Virtual Reality Environment to Different Modes of Treadmill Walking," *Gait & Posture*, vol. 39, no. 3, pp. 939–945, 2014.
- [10] Jää-Aro and Kai-Mikael, "Virtual Reality - State of the Art," Royal Institute of Technology, Stockholm, Tech. Rep., 1993.
- [11] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE," in *Proc. of the Conf. on Computer Graphics and Interactive Techniques*, 1993, pp. 135–142.
- [12] A. S. Mathur, "Low Cost Virtual Reality for Medical Training," in *Proc. of the Int. Sym. on Virtual Reality*, 2015, pp. 345–346.
- [13] C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, S. Davidoff, J. S. Norris, and G. Longo, "Immersive and Collaborative Data Visualization using Virtual Reality Platforms," in *Proc. of the Int. Conf. on Big Data*, 2014, pp. 609–614.
- [14] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek, "Automatic Step Detection in the Accelerometer Signal," in *Proc. of the Int. Workshop on Wearable and Implantable Body Sensor Networks*, vol. 13, no. 1, 2007, pp. 80–85.
- [15] J. M. Jasiewiczza, J. H. Allumb, J. W. Middleton, A. Barriskilld, P. Condiea, B. Purcella, and R. C. T. Lia, "Gait Event Detection using Linear Accelerometers or Angular Velocity Transducers in able-bodied and Spinal-Cord Injured Individuals," vol. 24, no. 4, pp. 502–509, 2006.
- [16] C.-C. Yang, Y.-L. Hsu, K.-S. Shih, and J.-M. Lu, "Real-Time Gait Cycle Parameter Recognition Using a Wearable Accelerometry System," *Sensors*, vol. 11, no. 8, pp. 7314–7326, 2011.
- [17] "Building a Sensor for Low Latency VR," <https://www.oculus.com/en-us/blog/building-a-sensor-for-low-latency-vr/>, [Online; last visited on April 29, 2016].