Real-Time Gait Reconstruction For Virtual Reality Using a Single Sensor

Tobias Feigl*, Lisa Gruner†, and Christopher Mutschler‡
Fraunhofer Institute of Integrated Circuits (IIS)
Nürnberg

Daniel Roth§
Technical University Munich (TUM)
Munich

ABSTRACT

Embodying users through avatars based on motion tracking and reconstruction is an ongoing challenge for VR application developers. High quality VR systems use full-body tracking or inverse kinematics to reconstruct the motion of the lower extremities and control the avatar animation. Mobile systems are limited to the motion sensing of head-mounted displays (HMDs) and typically cannot offer this.

We propose an approach to reconstruct gait motions from a single head-mounted accelerometer. We train our models to map head motions to corresponding ground truth gait phases. To reconstruct leg motion, the models predict gait phases to trigger equivalent synthetic animations. We designed four models: a threshold-based, a correlation-based, a Support Vector Machine (SVM) -based and a bidirectional long-term short-term memory (BLSTM) -based model. Our experiments show that, while the BLSTM approach is the most accurate, only the correlation approach runs on a mobile VR system in real time with sufficient accuracy. Our user study with 21 test subjects examined the effects of our approach on simulator sickness and showed significantly less negative effects on disorientation.

Index Terms: Human-centered computing—Interaction paradigms—Virtual reality

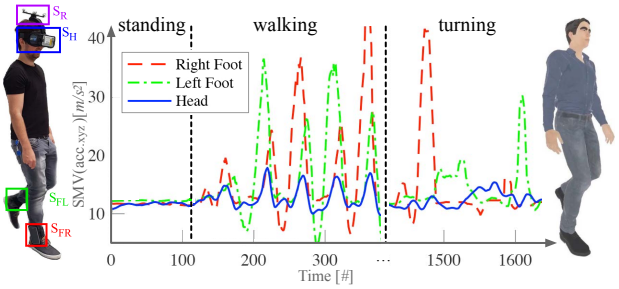

Figure 1: Our approach to gait reconstruction for mobile VR. Left: We automatically annotate data with reference inertial measurement unit (IMU) sensors (S_{FR} and S_{FL}) during the training phase. We position the user with markers of our reference system (S_R). For real-time gait reconstruction, we only use (S_H), the acceleration sensor of the head-mounted display (HMD). Center: We recognize gait phases from the signal magnitude vector (SMV) of the acceleration (blue curve): standing, walking, and turning. Right: We synchronize the gait animation of the avatar using the predicted gait phases.

1 INTRODUCTION

Avatars, virtual representations that are driven by human behavior [3], can embody the user in VR [2]. Previous work achieve embodiment of avatars in high-end VR through motion tracking and retargeting of behavior such as hand [1], body [19] or face and gaze features [29], or combinations [18]. They require data from certain marker-based tracking systems, additional (embedded) cameras or controllers to solve human poses or to morph target animations (blendshape) [20]. The resulting avatar embodiment may support the overall VR simulation experience and may avoid simulator sickness effects [17] as it provides visual orientation through motion representation [30]. A more realistic reconstruction of the physical motion may lead to a higher plausibility of the simulation [24], especially for low-end VR systems.

Such a reconstruction of motion is typically based on a motion classification and an appropriate matching to a known animation database. Its advantage is that input signals can be substituted as soon as the ground truth knowledge has been acquired. Thus, it inherits the *sensing*, the *recognition*, and the *synthesis* of the approximated motion to achieve an appropriate output visualization. Cha et al. [5] present an approach to mobile face and body tracking, but with additional and more invasive head-mounted sensors that are not available in mobile low-end VR systems. Because these low-end tracking approaches are still limited by remaining physical limitations, and therefore their motion reconstruction ability is also limited, they motivate the development of more complex motion tracking and reconstruction approaches.

We propose an approach to reconstruct the motion of the lower extremities to avatars in VR using data from a single IMU. We formulate the problem as a supervised classification problem and compare four methods: a threshold-based, a Pearson-correlation-based, a cubic SVM, and a BLTSM. We found that the BLSTM offers the highest classification confidence, but requires computational effort that does not meet real-time requirements on mobile VR devices. On the other hand, the correlation-based approach offers both sufficient accuracy and less computing effort at high refresh rates. In a study with 21 participants, we also examined our approach to determine whether the simulator sickness, which may be influenced by the lack of visual reference points such as legs or feet, can be alleviated by our reconstruction. An avatar representation using our correlation-based approach showed significantly fewer disorientation symptoms than the threshold-based version and without an avatar at all.

We organize the paper as follows. Sect. 2 reviews related work. Sect. 3 introduces our motion reconstruction methods that we evaluate in Sect. 4. Sect. 5 evaluates the correlation method and Sect. 5 discusses the results of our user study. Sect. 6 concludes.

2 RELATED WORK

User embodiment in virtual environments (VE) is described as the appropriate representation of users to others, but also to themselves [2], typically in the form of avatars, virtual characters controlled in real time by the behavior of the user [3]. Previous work shows that synchronous visuomotor or visuotactile stimulation may lead to a higher degree of body ownership and agency [11], the latter describes the feeling of control over the avatar. This in turn implies that synchronous motion reconstruction and retargeting of motions on avatars can also foster the suspension of disbelief [24]. Therefore,

*Is also with the Programming Systems Group, Friedrich-Alexander University (FAU) Erlangen-Nürnberg. e-mail: tobias.feigl@fau.de
†E-mail: lisa.gruner@fau.de
‡Is also with the Department of Statistics, Ludwig-Maximilians-University (LMU), e-mail: christopher.mutschler@iis.fraunhofer.de
§Chair for Computer Aided Medical Procedures and Augmented Reality, Technical University Munich (TUM), e-mail: daniel.roth@tum.de

978-1-7281-7675-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ISMAR-Adjunct51615.2020.00037

most of the previous work focused on mapping the reconstructed motion to a virtual representation to allow for a more sophisticated user embodiment, especially for high-end VR systems that use multiple additional sensors. In mobile low-end VR applications, however, real-time reconstruction of the visually appealing and natural human motion to the avatar representation, especially with limited sensor input, is still a challenging task.

2.1 Full body Motion Reconstruction

For high-end VR, there are approaches that recognize the user's action and search for and display appropriate full body motion from a database [10, 32]. Others synthesize a sequence of motions using existing motions in a database to reconstruct motions [14, 28]. However, database search results in an additional delay, and these methods require input signals from expensive and elaborate full body motion capturing systems to control or reconstruct the motion. To reduce costs, others use multiple inertial sensors [23, 28] and focus on synthesizing natural-looking motion sequences using a reduced number of sensors [7, 12, 13, 27] that need to be rigidly attached to the head [9], feet [13], wrists [12, 16], lower torso [16], and ankles [12] to cause a character's motion reconstruction. The latest data-driven methods work well with a motion database and may find matches directly from the sensor input [7, 14]. However, they have problems if the search does not return a hit, as this may lead to a considerable loss in quality of the synthesized result. Interpolation [21] addresses this, but cannot create new poses that are unavailable in the database, and cannot compensate for the delay.

2.2 Gait Reconstruction

Zhong et al. [34] present an overview of various approaches to analyzing the lower extremities, especially the gait, with wearable sensors. Shi et al. [22] and Ying et al. [32] propose algorithms that use a foot-mounted accelerometer for automatic foot gesture and step detection. Jasiewicza et al. [10] recognized gait events using either foot-mounted linear acceleration sensors or angular velocity sensors. Yang et al. [31] further improved their real-time gait detection. Caserman et al. [4] recognize individual (left and right) steps of the user almost in real time. They use an adaptive, threshold-based peak detector for the pre-filtered acceleration *acc* signal (provided by an Oculus Rift HMD). However, they require an initial calibration with unknown sensors and suffer from undetected steps. The error varies by almost 50% between different experiments, due to estimation errors of the head orientation and the linear acceleration. These are known problems, since even a single integration of an inertial signal accumulates high errors over a short period of time.

2.3 Quintessence

Most methods require motion capturing and knowledge bases of human motions to obtain candidate poses for the results or the basis of their synthesis. The quality of the synthesized motion and the running time of the search algorithms depend on the size of the knowledge base. In contrast, our approach does not require a database as we recognize gait phases to directly trigger an animator to reconstruct the synthetic motion of the lower body in VR. To the best of our knowledge, only Caserman et al. [4] reconstruct the gait motion with a single accelerometer mounted on the head, but with insufficient stability and accuracy. Our baseline method is based on their approach and aims to reconstruct the lower body motion for mobile low-end VR systems with low latency, low computation effort, without additional sensors, but with a natural reconstruction of the gait motion to an avatar representation.

3 Method

3.1 General Approach

Fig. 2 shows the processing pipeline of our general approach. We preprocess raw xyz-accelerations (acc_{xyz}) from a single head-mounted

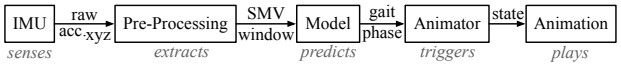

Figure 2: Overview of our processing pipeline.

IMU, before our models recognize gait phases (gp) from sliding windows that embed the rotation-invariant signal magnitude vector (SMV) of the acc_{xyz} stream. Fig. 1 shows exemplary $SMV(acc_{xyz})$s of the head-mounted IMU (blue) and two foot-mounted reference sensors (red and green) of a user who is standing, walking (cut-off), and turning. In a *training* phase, our models learn to map a known $SMV(acc_{xyz})$ to the corresponding ground truth gp that we obtained a-priori from reference sensors (that are rigidly mounted on the ankle as in [23, 28]). In a *live* phase, our models then predict the current gp from unknown $SMV(acc_{xyz})$. A gp then forces our animator to trigger a corresponding animation playback to control avatar animation states. The animator therefore controls the speed adjustment. We exploit the knowledge that a gait cycle (gc) is divided into eight individual gps that our models reliably recognize: *Initial Contact* (IC), *Loading Response* (LRE), *Mid Stance* (MST), *Terminal Stance* (TST), *Pre-Swing* (PSW), *Initial Swing* (ISW), *Mid Swing* (MSW), and *Terminal Swing* (TSW), see Fig. 3.

Gait Phase Classification. To animate the reconstructed motion in VR, we aim to recognize the gp before the user "lifts" the leg (MST to TST) and to track the transition from MST to ISW to ensure that the signal really represents a valid gc of one leg, see the upper graph in Fig. 3).While the right foot finishes its gc (from samples 0 to 50), the left foot stands still in the LRE phase (0 to 25). The left foot begins to move from the MST (at 25) to the ICO and LRE (at 85) (when we raise the leg). The head-mounted *acc* (blue curve) clearly represents the MST (at 25) and ISW (at 55) of the left foot (green) in a pattern that also repeats for the right foot (red curve) and shows MST (at 80) and ISW (at 105) of the right foot, which occur with TSW and LRE of the left foot. However, only using *acc* values also poses the challenge to reliably separate unrelated head motions from motions that indicate gps.

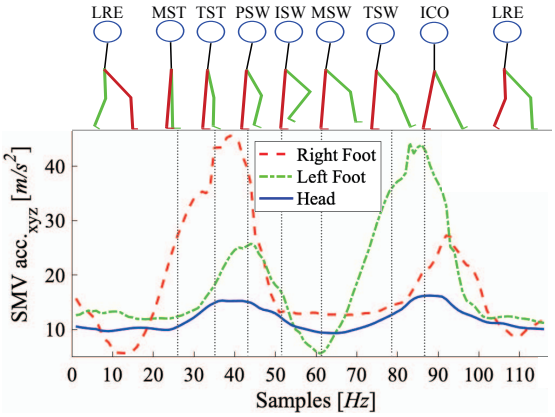

Figure 3: Gait cycle and its phases (top) and SMV of the head-mounted *acc* (blue) and its synchronized reference SMVs of *acc* from sensors attached to the left (green) and right (red) foot (bottom).

3.2 Pre-Processing

Windowing. We first capture the acc_{xyz} input stream and slice it into overlapping consecutive windows. We found that a window size w=50Hz and an overlap of 1Hz yield the highest accuracy, are robust against turning and rotary movements, cover all gps, and are

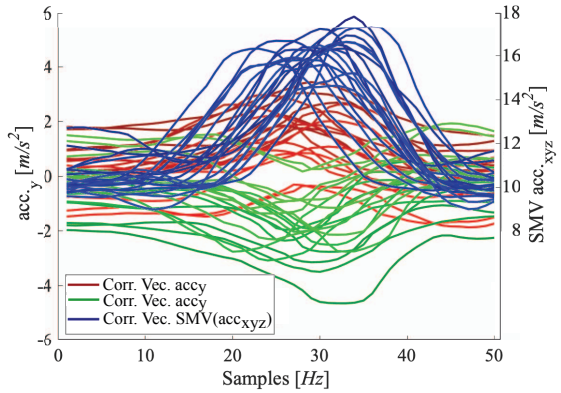

Figure 4: Correlation vectors from the head-mounted accelerometer: C_g, a general one (blue) to recognize the gp and two $C_{g_{side}}$, that recognize if its either a gp from the left (green) or right (red) foot.

long enough to predict future gps from history and context. Hence, as the head-mounted accelerometer samples at $100Hz$, already $0.5s$ of head motion cover enough motion characteristics to accurately and reliably reconstruct leg motion.

SMV. To reduce the computational effort, we reduce the input dimension to a single axis as we calculate the:

$$SMV(acc_{xyz}) = \sqrt{acc_x^2 + acc_y^2 + acc_z^2}. \quad (1)$$

Since the gravitational component of the *acc* on the body axis is not affected when standing, walking, and turning, we subtract it from the SMV and thus, also eliminate the deflections caused by it. Because head rotations do not affect linear *acc* on the body axis, our models can separate them from walking, standing or turning. Our models therefore reliably estimate gps, as they only process on rotation-invariant linear *acc* [33].

3.3 Threshold-based Method (THR)

We implemented THR according to Caserman et al. [4]. THR recognizes a gp above a certain threshold T in an incoming window of $SMV(acc_{xyz})$ data. In a training phase, we derive user-specific threshold values T_s, that yield in the same number of gps as the reference sensors and that are the smallest to keep the delay low. From there we derive general thresholds T_g ($=\frac{1}{n}\sum_{i=1}^{n} T_{s(i)}$) that we use in a live phase with unknown users, when we have no T_s. Each value v_i of the SMV is compared with T_g: If $v_i > T_g$, we recognize a gp, else no gp. We derive an additional general threshold $T_{g(side)}$ on acc_y data to determine its specific foot. If $v_i > T_{g(side)}$, we recognize a left foot gp, else a right foot. Hence, $T_{g(side)}$ indicates a lateral inclination of a person's body, which in turn indicates the upward movement of the opposite leg. By clearance, those general thresholds may not generalize well to unknown users as human gait is individual and yield incorrect predictions while rotating.

3.4 Pearson Correlation-based Method (COR)

We compare two vectors x and y to determine their linear correlation coefficient r as a measure of their dependency, with an unknown incoming sequence x=$SMV(acc_{xyz})$ and y, a known sequence that represents the gp (ISW to ISW). COR determines correlation coefficients from a general vector C_g (mean of all user-specific vectors: $mean(C_s)$) to detect the gp, see the blue curve in Fig. Fig. 4. A second general vector $C_{g_{side}}$ determines the left or right foot on acc_y data, see the green and red curves in Fig. Fig. 4. We use the correlation function for pairs of $(x_i, y_i), i$=1,...,N with a linear r [15, 26]:

$$r = \frac{\sum i(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum i(x_i - \bar{x})^2}\sqrt{\sum i(y_i - \bar{y})}}, \quad (2)$$

to describe the the correlation of $x(n)$ and $y(n) \in \{C_g, C_{g_{side}}\}$ over time with "cross" state $x \neq y$ [26]. The value of r ranges from -1 to 1: r=1 is the best match for a rising curve (r approaches 1 if x is similar to the known y, samples 0 to 25 in Fig. 4); r=0 expresses the lowest match, i.e., no gp was recognized; and r=-1 is the best match of a falling curve (r approaches -1 if x is similar to the known y, samples 25 to 50 in Fig. 4). We found that $r > |0.9|$ yields accurate gps with lowest delay and is robust to rotations.

3.5 Support Vector Machine-based Method (SVM)

In contrast to THR and COR, our cubic SVM detects gp and the corresponding foot directly from raw data. In a training phase SVM is trained based on windows of SMV(acc_{xyz}) with corresponding ground truth labels of gps (ISW to ISW) and the respective left or right foot. In a live phase SVM predicts foot-specific gps and its confidence level from unknown SMV(acc_{xyz}) of overlapping windows. Using a grid search on well-known methods (SVM, Decision Trees (DT), and Gaussian processes (GP)) and combinations of handcrafted statistical and frequency domain features, we selected SVM as the optimal model w. hyperparameters that result in the highest accuracy of the gp classification.[1]

3.6 BLSTM-based Method (BLSTM)

A BLSTM consists of two hidden forward and backward LSTM layers to capture gp and the corresponding foot directly from raw data. BLSTM captures and memorizes spatial and temporal dependencies in SMV(acc_{xyz}), as it tracks the emergence of these features in a sequence from both the past and the future, but at the expense of computational effort. Our architecture consists of a single BLSTM layer (to reduce computational effort), followed by a dropout layer (to prevent overfitting) [25]. A fully connected layer multiplies the input by a weight matrix and then adds a bias vector. Next, a Softmax layer applies a Softmax function to the input, and a final classification layer calculates the cross entropy loss for our multi-class classification problem. Using a grid search, we derived the optimal architecture w. hyperparameters, that lead to the highest accuracy of the gp classification.[2]

3.7 Motion Synthesis

Each method sends foot-specific gps to the animator. The animator then updates its internal state to trigger gait animation. We achieve motion synthesis by adding key points to the animation (captured and synchronized by our optical reference system, see Sect. 4). The predicted gps trigger them to control the animation.

State Machine. The animator also uses smooth, linear motion state transitions (which compensate for faulty classification of gp). Hence, its state machine is set to a suitable fine-grained motion state for each foot-specific gp. The more granular a method recognizes

[1] SVM grid search parameters: SVM Linear: C ∈[1,10,**100**,1000]; SVM Poly.: C ∈[1,10,**100**,1000], *degree*=**3** ∈[2:1:6]; SVM RBF: γ ∈[**0.001**,0.0001], C ∈[1,10,**100**,1000]; DT: max. depth=**97** ∈[1:1:150], max. features=**27** ∈[1:1:30], max. leaf nodes=**15** ∈[1:1:20]; GP: kernel ∈[RBF, **Matern52**, rotational quadratic], RQ α ∈[0.001,0.01,0.1,1.0,**2.0**:10], RQ length scale ∈[0.1,**1.0**,:,10], RBF length scale ∈[0.1,1.0,**3.0**,:,10], M52 length scale ∈[0.1,**1.0**,:,10], M52 ν ∈[0.1,1.0,1.5,**2.0**,2.5,:,10];

[2] BLSTM grid search parameters: solver ∈[SGD, **Adam**, rmsprop]; β_1, β_2=**0.01**; momentum=**0.9**; initial learning rate (*LR*)=**0.009** ∈[1.0:0.1:0.00001]; *LR* drop period ∈[0,10,**50**,100] epochs; *LR* drop rate=**0.9**; batch size ∈[128,256,512,**1024**,2048]; B/LSTM layers=**2**; LSTM cells per layer=**128**; shuffle ∈[no,**per epoch**]; gradient clipping=max(input); dp ∈[10,20,**50**,75]%. (Highest accuracy with bold numbers.)

Table 1: Recognition accuracy of gp.

Method	Misclassification rate [%]						
	MAE	MSE	RMSE	CEP_{25}	CEP_{50}	CEP_{75}	CEP_{95}
THR	0.512	0.327	0.571	0.311	0.464	0.737	0.951
COR	0.087	0.010	0.098	0.050	0.079	0.119	0.181
SVM	0.056	0.004	0.062	0.039	0.053	0.066	0.086
BLSTM	**0.016**	**0.003**	**0.058**	**0.001**	**0.002**	**0.005**	**0.076**

Table 2: Recognition delay.

Method	Delay [ms]			
	∅	Min.	Max.	SD
THR	+60.0	+30.0	110.0	40.0
COR	-30.0	-50.0	+40.0	20.0
SVM	-70.0	-50.0	+20.0	20.0
BLSTM	-140.0	-280.0	0.0	30.0

Table 3: Computational effort.

Method	t_{train} [h]		t_{live} [s]		t_{live} [Hz]	
	CPU	gpU	CPU	Mobile	CPU	Mobile
THR	0.2	-	0.00073	0.0096	1370	104
COR	0.3	-	0.00067	0.0114	493	87
SVM	19.6	-	0.01042	0.1666	96	6
BLSTM	12.4	3.4	0.01786	0.3333	56	3

individual gp, the more granular it triggers the state machine of our animator and the more "natural" the animation becomes. If an initial gp is recognized, the animator plays a start animation, Fig. 5 "Start Walk Right" and "Start Walk Left". We trigger animation playback at $100Hz$ (just like the accelerometer sampling rate) based on the predicted gp. Therefore, the status of the animator always corresponds to the current and individual speed of a user.

Error Handling. Since we also sample more ($100Hz$) than we render ($60Hz$), we use the intermediate estimates for error and outlier filtering. This oversampling also helps to filter incorrect predictions in a pre-processing step. Since the animator updates its states at $100Hz$, it smoothes the latest history of gps (5 time steps) to compensate for misrecognized steps, to remove outliers, and not simply to "interrupt" an ongoing animation cycle. Thus, we only perform smooth blending between two physically incorrect animations in cases of at least 3 consecutive incorrectly recognized gps. The animator switches to the second phase of the animation (see Fig. 5 "Finish Walk Right" and "Finish Walk Left"), restores the status "standing", and does not animate the entire gait cycle. As the state machine of the animator guarantees and limits possible successive sub-animations, we never end up in a "sudden" animation break.

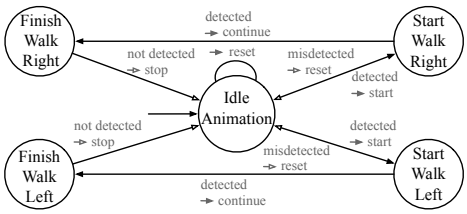

Figure 5: Coarse-grained state machine of our animator.

4 Technical Evaluation

4.1 Experimental Setup

Reference Systems. We recorded 6DoF reference positions and orientations with 28 cameras (that cover a volume of $11.025m^3$, $45 \times 35 \times 7m$) of the millimeter-accurate optical motion tracking system (Qualisys) with a spherical error probable (SEP_{95}) $\leqslant 5mm$ and $\leqslant 0.1°$). The subjects wore 4 small trackable reflective markers, attached to an elastic ribbon of the HMD, see Fig. 1, to track the calibrated 6DoF pose of each HMD with a constant $300Hz$. The reference poses were broadcasted via $5GHz$ WiFi to render the pose of the VR camera. We recorded the reference gps with two Samsung Galaxy S7 phones with accelerometer and gyroscope sensors (STMicroelectronics LSM6DS3 samples acc at $\pm 16G$ and gyr at $\pm 1000dps$ at quasi constant $100Hz$) that were rigidly mounted on each leg (phones in inverse portrait mode), see Fig. 1. With reflective markers mounted on the phones, we found that these reference sensors provide the same gait patterns as Qualisys.

Measurement System. We used a Samsung Gear VR HMD in combination with a Samsung Galaxy S8 phone (Android version 8.0) that both renders the VR scene at $60Hz$ and predicts gps at about $100Hz$. The phone's IMU provides accelerations at $100Hz$. We accessed the IMU sensor data of the smartphones via the Android API (version 6, 2019) [8]. Qualysis and the three phones were connected to the same global NTP time server to store their recordings together with global NTP time-synchronized time stamps.

Dataset. To collect the training, validation, and test data, we conducted a user study with 6 participants (4 male, 2 female, age [years]: with an average of M=26.82, and standard deviation SD=4.31); height [m]: from $1.52m$ to $1.96m$, M=1.78, SD=0.07), weight [kg]: M=68.7, SD=9.82). 5 users provide training and validation data. The randomly selected "left-out" user provides test data that are unknown to the methods for evaluating their generalizability. Note that these users have never participated in our subjective user study in Sect. 5. For 10 iterations, each user started standing at the same position, next walked about $20m$, then turned (some went in a small circle while others turned in place) and returned to the starting position, similar to Fig. 7. The users were allowed to move and rotate their body and head freely. In total, we recorded about $37.5min$ of motion data (=6 subjects $\cdot$ $6.25min$, SD=$0.5min$, $729m$ per user). The complete study resulted in a total distance of approximately $4.37km$ at similar speeds (M=$6.8\frac{km}{h}$; min. $6.3\frac{km}{h}$; max. $7.4\frac{km}{h}$; SD=$0.37\frac{km}{h}$). We collected a clean data pool with 224.950 overlapping acceleration windows (w=50 at $100Hz$, and 1 sample overlap) with corresponding reference poses and gp. From there, we generated two datasets: (1) a training (70%=131.220 windows) and a validation dataset (30%=56.237 windows) from 5 users; and (2) a test dataset (37.491 windows) from the remaining "left-out" to evaluate the generalizability of the methods.

4.2 Results

Accuracy Benchmark. Table 1 describes the accuracy (misclassifications in [%]) of each method to predict the correct gp for each overlapping window by the mean absolute error (MAE), mean square error (MSE, interprets small outliers), root mean square error ($RMSE$, interprets large outliers), and circular error probable (CEP, probability of a radial error of our approach) from 25% to 95%.

The results show that BLSTM yields the lowest absolute error (MAE=0.016%), performs best (only 1.6% of all gps are misclassified), and therefore predicts a correct gp in 98.4 of all cases, while THR (worst) predicts only correct in 49 out of 100 cases. We found THR performs worst, while T_g returns fewer errors for $MAE = 0.31\%$ (69 out of 100 gps are correct), T_{gs} adds a larger error that leads to an overall error of $MAE = 51\%$. In contrast, COR performs significantly better than THR and is similar to SVM ($MAE = 8.7\%$ vs. $MAE = 5.6\%$). We think that COR, SVM, and BLSTM perform better than THR as they exploit the history of the samples, i.e., the emergence of gps between consecutive ISWs. BLSTM offers the best performance as its BLSTM structure learns from both the past and the future.

We see a similar trend in the outliers (MSE and RMSE) as in the absolute errors (MAE and CEP). THR shows a high number of small outliers ($MSE = 0.327$) that may be due to misclassification while the "left-out" is standing and looking around. Instead, the large outliers of THR ($RMSE = 0.57$) may be due to misclassification when a user turns while walking. The other three data-driven methods show a much smaller number of (small and large) outliers than THR, as they do not misclassify standing, walking, turning. In fact, COR shows slightly more outliers than SVM and BLSTM, as

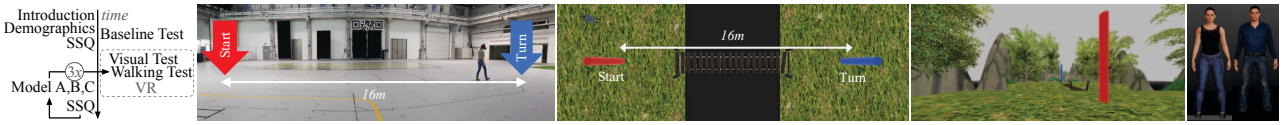

Figure 6: *Left:* Study procedure. *Center:* Scenario settings; physical world, virtual representation. *Right:* Female and male humanoid avatars.

Figure 7: Avatar visualizations. *Left to right:* No avatar, block avatar, and two walking humanoid avatars (female, male).

it may struggle to detect the correct foot-specific gp.

Computational Effort and Delay For the "left-out" dataset Table 3 lists training times t_{train}, for all windows , and t_{live}, the average time it takes to inference the gp on a single window. For t_{live} the Mobile (phone) times are significantly slower than the CPU times, as the phone cannot use its full computing power while it renders the VR scene. THR and COR predict a new gp in less than $11ms$, whereas the SVM and the BLSTM are computationally too costly.

We determine the delay for each method as the time difference (in ms) between the time of the reference gp and its time of detection. The delay is 0 if a method directly detects a gp when it happens. A delay is <0 if a method predicts a gp before it happens. The delay is >0 if a method predicts a gp after that. Table 2 shows that THR had the highest delay (+60ms on average). COR showed an average delay of -30ms. Instead, SVM and BLSTM showed an average negative delay (SVM: -70ms; BLSTM: -140ms).

Quintessence. Both the delay and the computational effort may add up and affect the visualization of the motion reconstruction. THR shows the highest delay (+60ms) with the lowest computing costs (10ms) but yields lowest accuracy. Even though SVM and BLSTM yield the highest accuracy at lowest delay (-70ms and -140ms), that may compensate for their high computing costs (167ms and 333ms), for Mobile the frame rate of the VR scene drops below $15fps$ on average while processing them. Since COR showed the best compromise between accuracy, low delay (-30ms), and low computing effort (11.5ms) it is applicable for our real user study.

5 Exploratory User Study

5.1 Design

We determined the impact of our pipeline's gait movement representation on the perception of simulator sickness [17] in a mobile VR simulation. We conducted a one-factor (avatar representation) within-subjects study that consists of a comparison of 3 VR scenes with different avatar representations (see Fig. 6, right) to a physical baseline. The subjects performed walking trials with each representation, namely i) *no avatar* representation, ii) a simplified *block avatar* representation without animation, and iii) a humanoid avatar representation that is animated using *our* motion reconstruction approach w. COR, see Fig. 7. The gait animation consists of a regular gait cycle with moderate arm swings.

5.2 Participants

21 participants (14 male, 7 female; age [years]: from 21 to 57, M=28.76, $SD = 9.60$; height [m]: from 1.55 to 1.88, $M = 1.75$, $SD = 0.09$; weight [kg]: $M = 73.90$, $SD = 16.57$) were recruited using mailing list invitations (no disturbances with equilibrium; 2 suffer from acrophobia; 4 easily get car sickness; 2 reported dyschromatopsia; all had normal or corrected visual acuity, verified by a Landolt-C test; 15 had previous VR experiences and 14 with HMD; 8 play video games daily, 8 moderately, and 5 never).

5.3 Measures

To investigate the effect of the models on the simulator sickness, we assessed the Simulator Sickness Questionnaire (SSQ) [17] after each trial and compared it with a baseline assessment.

5.4 Procedure

Fig. 6 (left) depicts the study procedure: We informed the subjects about it and assessed demographic questions and pre-exposure simulator sickness measures [17]. Then, we introduced them to the tasks, calibrated the setup, and captured baseline measures when the subjects walked back and forth (from "Start" over "Turn" to "End") a 16m distance in the physical world, see Fig. 6. Next, we gave them time to acclimatize, before the Landolt vision test [6] was presented in VR. After another calibration and acclimatization, the subjects walked again back and forth a 16m distance in VR. The VE was designed, similar to a virtual pit scenario, with a nature scene and a crossable bridge, see Fig. 6. Users were free to walk from "Start" over "Turn" to "End" on a path of their choice (some walked faster than others, some turned in circles or on spot). We did not define or display a line, except that they had to cross the bridge with the possibility to turn their heads freely under all circumstances. We only instructed them to focus on their feet to ensure exposure to their virtual representations each time they cross the bridge. In a random order (to avoid biasing the sequencing), we evaluated 3 different avatar conditions in 3 trials: (1) *No Avatar*; (2) *Block Avatar*; and (3+4) *Ours*, the gender-specific humanoid avatar with our motion reconstruction, see Fig. 6. Fig. 7 shows the conditions from the point of view of the user while looking down while walking. After each trial, we assessed the SSQ and the subjects had time to rest before starting the next condition. The entire study lasted about 30min.

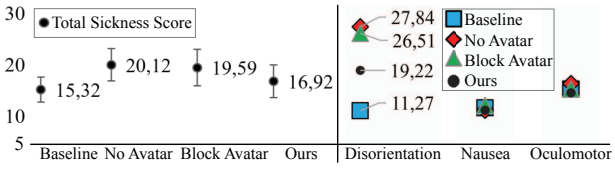

Figure 8: SSQ results: Total sickness- (left) and subscores (right). Our approach reduces the perception of disorientation significantly.

5.5 Results

Simulator Sickness. We calculated the repeated measurement analysis of variance (ANOVA) for the SSQ total score (TSS) and

the subscores. Fig. 8 shows descriptive results for the values for disorientation, oculomotor, and nausea. There was no major effect for the TSS. Overall, the baseline measure scored the lowest, followed by *Ours*. Instead, the analysis of the subscores showed a significant main effect on disorientation; $F(3,40.38) = 5.298$, $p = .009$, $\eta_p^2 = 0.209$. Pairwise comparisons showed that the *No Avatar* condition results in a significantly higher degree of disorientation than the *Baseline* condition ($p = .025$), while *Ours* is closest to the *Baseline*.

6 Conclusion

In this paper we compare 4 different methods on their performance to reconstruct gait motion, as well as their reproduction to avatar motion in a mobile low-end VR application. While our BLSTM-based approach always predicts the correct gait phase at lowest delay but drastically drops the refresh rate of the virtual scene, our Pearson correlation-based approach is an optimal compromise in delay, computational effort, and accuracy that also reduces disorientation effects based on its avatar-based motion reproduction, in particularly to improve mobile low-end VR applications.

Acknowledgments

This work was supported by the Bavarian Ministry for Economic Affairs, Infrastructure, Transport and Technology through the Center for Analytics Data Applications (ADA-Center) within the framework of "BAYERN DIGITAL II".

References

[1] F. Argelaguet, L. Hoyet, M. Trico, and A. Lécuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *Proc. Int. Conf. on Virtual Reality and 3D User Interfaces*, pp. 3–10. Greenville, SC, 2016.

[2] S. Benford, J. Bowers, L. E. Fahlén, C. Greenhalgh, and D. Snowdon. User embodiment in collaborative virtual environments. In *Proc. Intl. Conf. on Human factors in Computing Systems (SIGCHI)*, pp. 242–249. Denver, Colorado, 1995.

[3] J. Blascovich and J. Bailenson. *Infinite reality: Avatars, eternal life, new worlds, and the dawn of the virtual revolution*. William Morrow & Co, 2011.

[4] P. Caserman, P. Krabbe, J. Wojtusch, and O. von Stryk. Real-time step detection using the integrated sensors of a head-mounted display. In *Proc. Int. Conf. on Systems, Man, and Cybernetics*, pp. 770–778. Budapest, Hungary, 2016.

[5] Y.-W. Cha, T. Price, Z. Wei, X. Lu, N. Rewkowski, R. Chabra, Z. Qin, H. Kim, Z. Su, Y. Liu, et al. Towards fully mobile 3d face, body, and environment capture using only head-worn cameras. *Trans. on Visualization and Computer Graphics*, 24(11):2993–3004, 2018.

[6] H. Dietze. Die bestimmung der sehschärfe. *Klinische Monatsblatter fur Augenheilkunde*, 235(9):1057–1075, 2018.

[7] H. Eom, B. Choi, and J. Noh. Data-driven reconstruction of human locomotion using a single smartphone. In *Proc. Intl. Conf. on Computer Graphics Forum*, pp. 11–19. Zurich, Switzerland, 2014.

[8] T. Feigl, S. Kram, P. Woller, R. H. Siddiqui, M. Philippsen, and C. Mutschler. Rnn-aided human velocity estimation from a single imu. *Sensors*, 20(13):3656–3682, 2020.

[9] T. Feigl, C. Mutschler, and M. Philippsen. Supervised learning for yaw orientation estimation. In *Intl. Conf. on Indoor Positioning and Indoor Navigation*, pp. 206–212. Nantes, France, 2018.

[10] J. M. Jasiewicz, J. H. J. Allum, J. W. Middleton, A. Barriskill, P. Condie, B. Purcell, and R. C. T. Li. Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals. *Gait & Posture*, 24(4):502–509, 2006.

[11] K. Kilteni, R. Groten, and M. Slater. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387, 2012.

[12] H. Kim and S.-H. Lee. Reconstructing whole-body motions with wrist trajectories. *Graphical Models*, 75(6):328–345, 2013.

[13] C. Mousas. Full-body locomotion reconstruction of virtual characters using a single inertial measurement unit. *Sensors*, 17(11):2589–2597, 2017.

[14] C. Mousas, P. Newbury, and C.-N. Anagnostopoulos. Data-driven motion reconstruction using local regression models. In *Proc. Intl. Conf. on Artificial Intelligence Applications and Innovations*, pp. 364–374. Rhodes, Greece, 2014.

[15] W. H. Press. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 2002.

[16] Q. Riaz, G. Tao, B. Krüger, and A. Weber. Motion reconstruction using very few accelerometers and ground contacts. *Graphical Models*, 79(4):23–38, 2015.

[17] Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *Aviation Psychology*, 3(3):203–220, 1993.

[18] D. Roth, G. Bente, P. Kullmann, D. Mal, C. F. Purps, K. Vogeley, and M. E. Latoschik. Technologies for Social Augmentations in User-Embodied Virtual Reality. In *Proc. Intl. Conf. on Virtual Reality Software and Technology*, pp. 770–778. New York, NY, USA, 2019.

[19] D. Roth, J.-L. Lugrin, D. Galakhov, A. Hofmann, G. Bente, M. E. Latoschik, and A. Fuhrmann. Avatar realism and social interaction quality in virtual reality. In *Proc. Intl. Conf. on Virtual Reality and 3D User Interfaces*, pp. 277–278. Greenville, SC, 2016.

[20] D. Roth, J.-P. Stauffert, and M. E. Latoschik. Avatar embodiment, behavior replication, and kinematics in virtual reality. In *VR Developer Gems*, pp. 321–346. AK Peters/CRC Press, 2019.

[21] A. Safonova and J. K. Hodgins. Construction and optimal search of interpolated motion graphs. In *Proc. Intl. Conf. on SIGGRAPH*, pp. 106–112. San Diego, CA, 2007.

[22] X. Shi, J. Pan, Z. Hu, J. Lin, S. Guo, M. Liao, Y. Pan, and L. Liu. Accurate and fast classification of foot gestures for virtual locomotion. In *Intl. Symp. on Mixed and Augmented Reality*, pp. 178–189. Beijing, China, 2019.

[23] T. Shiratori and J. K. Hodgins. Accelerometer-based user interfaces for the control of a physically simulated character. *Trans. on Graphics*, 27(5):1–9, 2008.

[24] M. Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Phi. Trans. of the Royal Society B: Bio. Sci.*, 364(1535):3549–3557, 2009.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Machine Learning Research*, 15(1):1929–1958, 2014.

[26] P. Stoica and R. L. Moses. *Introduction to Spectral Analysis*. New York, NY, USA, 2005.

[27] J. Tautges, B. Krüger, A. Zinke, and A. Weber. Reconstruction of human motions using few sensors. *Virtuelle und Erweiterte Realität: Workshop der GI-Fachgruppe VR/AR*, 2(13):1–12, 2008.

[28] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H.-P. Seidel, and B. Eberhardt. Motion reconstruction using sparse accelerometer data. *Trans. on Graphics*, 30(3):1–12, 2011.

[29] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *Trans. on Graphics*, 30(4):1–10, 2011.

[30] C. Wienrich, C. K. Weidner, C. Schatto, D. Obremski, and J. H. Israel. A virtual nose as a rest-frame - the impact on simulator sickness and game experience. In *Proc. Intl. Conf. on Virtual Worlds and Games for Serious Applications*, pp. 1–8. Würzburg, Germany, 2018.

[31] 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*, 11(8):7314–7326, 2011.

[32] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek. Automatic step detection in the accelerometer signal. In *Proc. Intl. Conf. on Wearable and Implantable Body Sensor Networks*, pp. 80–85. Springer, Berlin, Heidelberg, 2007.

[33] A. Yurtman and B. Barshan. Activity recognition invariant to sensor orientation with wearable motion sensors. *Sensors*, 17(8):1838, 2017.

[34] R. Zhong, P.-L. P. Rau, and X. Yan. Gait assessment of younger and older adults with portable motion-sensing methods: A user study. *Mobile Information Systems*, 7(3), 2019.