International Journal of Semantic Computing
© World Scientific Publishing Company

Machine Learning Prediction of Locomotion Intention from Walking and Gaze Data

Gianni Bremer*, Niklas Stein* and Markus Lappe
Institute for Psychology, University of Muenster
Muenster, Germany
gianni.bremer@uni-muenster.de
niklas.stein@uni-muenster.de
mlappe@uni-muenster.de
**Gianni Bremer and Niklas Stein are co-first authors.*

Preprint of an article published in the International Journal of Semantic Computing, 2022
[doi.org/10.1142/S1793351X22490010]
© [copyright World Scientific Publishing Company]
[www.worldscientific.com/worldscinet/ijsc]

In many applications of human-computer interaction, a prediction of the human's next intended action is highly valuable. To control direction and orientation of the body when walking towards a goal, a walking person relies on visual input obtained by eye and head movements. The analysis of these parameters might allow us to infer the intended goal of the walker. However, such a prediction of human locomotion intentions is a challenging task, since interactions between these parameters are non-linear and highly dynamic. We employed machine learning models to investigate if walk and gaze data can be used for locomotor prediction. We collected training data for the models in a virtual reality experiment in which 18 participants walked freely through a virtual environment while performing various tasks (walking in a curve, avoiding obstacles and searching for a target). The recorded position, orientation- and eye-tracking data was used to train an LSTM model to predict the future position of the walker on two different time scales, short-term predictions of 50 ms and long-term predictions of 2.5 seconds. The trained LSTM model predicted free walking paths with a mean error of 5.14 mm for the short-term prediction and 65.73 cm for the long-term prediction. We then investigated how much the different features (direction and orientation of the head and body and direction of gaze) contributed to the prediction quality. For short-term predictions, position was the most important feature while orientation and gaze did not provide a substantial benefit. In long-term predictions, gaze and orientation of the head and body provided significant contributions. Gaze offered the greatest predictive utility in situations in which participants were walking short distances or in which participants changed their walking speed.

Keywords: LSTM; Virtual Reality; Eye Tracking; Locomotion; Path prediction; Machine Learning; Gaze.

1. Introduction

When we see people walk, we can infer where they want to go from their current trajectory [1]. This ability is used by animals and humans to avoid collisions in

everyday life. The same task has to be solved technically for hardware that physically interacts with walking humans: The need to improve driver assistance systems in cars has made accurate predictions of pedestrian walking behavior a necessity [2] and the anticipation of human actions, such as walking, can also play a key role in the development of assistive robots [3]. Prediction of locomotor intention can also be used to expand highly immersive virtual reality (VR) applications, in which complex environments can be explored by walking, which has been shown to be perceived as natural and presence-enhancing by users [4] and also allows them to acquire spatial knowledge about the virtual environment intuitively [5]. Various predicting methods for future trajectories have been proposed in the past [e.g. 6, 7, 8]. A recent approach is the use of artificial neural networks, particularly recurrent neural networks (RNNs). RNNs share parameters over a data sequence instead of treating every data point seperately. Thus, if a piece of information occurs at a slightly different point in the sequence, it is not offset against the weights of a completely different parameter. Therefore, RNNs have been used for the purpose of human motion prediction in different contexts [e.g. 9, 10, 11, 12, 13]. A common RNN approach is Long Short-Term Memory network (LSTM). LSTMs were first introduced by Hochreiter and Schmidhuber [14] and have already been used to predict the user position after 1 second based on sequential position and orientation data [15]. The same approach has also been used to create a controller model for redirected walking [16], a technique in which the paths of VR users can be imperceptibly manipulated to make maximum use of the given physical space [17, 18, 19].

RNNs use the time series of features of an ongoing behavior to predict future outcomes (labels). Relevant features for the prediction of walking behavior are parameters that need to be controlled during walking, notably the direction of walking and the orientation of the body, and measures that correlate with the goal or waypoints that the walker intends to reach, like head and gaze direction.

1.1. *Gaze Behavior During Walking*

Gaze is linked to motor action because we need to move our eye to targets of interest to collect the visual information we need for good action control [20]. Because eye movements usually precede any other motor action [21, 22] they can be informative for predicting action intention [23, 24, 25, 26]. Accordingly, walkers usually direct their gaze towards a target immediately before approaching it [27, 28]. However, at other points in time, gaze is also directed to obstacles. Walkers often look at the ground in front a few steps ahead for safe placements of the feet, particularly in uneven terrain [29, 30, 31, 32, 33]. Although this gaze behavior does not directly identify the ultimate target, it nevertheless indicates waypoints that the walker will use on a short timescale of the next few steps. Gaze behavior typically involves not only the eyes but also the head. When looking at the ground in front, walkers pitch their head downwards [34]. In addition, eye movements are linked to changes

of direction [27]. When walking in a curve, for example, walkers typically direct their gaze inward from the curve [35, 36]. Eye movements are also involved in deciding between alternative targets [25, 37] and in searching for targets between distractors [38]. Thus, eye movements during walking depend on task demands [39]. In summary, although gaze contains useful information about future actions, using this information to predict future locomotion behavior is a complex task. Therefore, deep learning models could be helpful to reveal intentions and future walking directions of users.

1.2. *Prediction Methods*

In the past, deep learning has already been used to predict eye-related parameters such as pupil diameter and fixation targets [e.g. 40, 41, 42]. Typically, these analyses were focusing on the analysis of the visual stimuli shown to users and thus either used Convolutional Neural Network (CNN) [e.g. 43] or combinations of CNN and RNN features [e.g. 44, 45, 46]. Cornia et al. [47] used the aforementioned LSTM architecture to predict so-called saliency maps for specific points in time, estimating the most likely fixation targets of a subject. Instead of using environmental information (such as the structure of the scene) for locomotion prediction, it is also possible to create a prediction model based on egocentric subject behavior data, to make the method more applicable. Because similar egocentric data can also be collected using inside-out tracking, head-worn IMUs and head-worn eye trackers, the prediction method can also be transferred to augmented reality scenarios.

Zank & Kunz [25] developed an algorithm using egocentric eye tracking to predict a walker's choice between two locomotion targets. In their experiment, participants were instructed to either freely choose one of the targets or to go to a specified target. Then, different locomotion prediction models were evaluated that used previous movements of the walkers to calculate probabilities for the two targets based on either assumptions about human walking behavior [26, 48, 49] or graph representations of the environment [7]. In narrow T-shaped corridors without open space, models including eye data were able to provide accurate predictions earlier than models without eye data. Later in a trial and in cases with open space, prediction accuracy was overall higher and eye data provided no additional benefit. Gandrud and Interrante [24] likewise used gaze data to predict a binary choice between two walking targets. The authors compared head direction, gaze direction and the position relative to the midline of a virtual hallway to predict the walking target. They concluded that head and gaze orientation had the potential to be useful in predicting a person's future direction of locomotion. However, both of these studies only distinguished between binary walking decisions. Further scenarios with less restrictions need to be evaluated to advance the use of behavioral measures for locomotor predictions. Cho et al. [15] presented a preliminary study of implementing a deep learning model for locomotion prediction in the context of redirected walking. They used head position and orientation to train an LSTM model to pre-

dict the user's position 100 frames (about 1 second) into the future while the user navigated a maze. They reported that the prediction worked well in two example users. However, their model was limited to the pre-defined maze map they used and did not include gaze data.

1.3. *Aim of the Study*

In the present study, we extend our previous work [50] to create a machine learning locomotion path prediction model using VR position, orientation, and gaze data. We added a full comparison of our best prediction model to sub-models that used only parts of the feature sets (e.g. only gaze data, or only gaze and position data) in order to obtain additional insight into the information needed to provide a valid locomotion prediction. Moreover, we performed additional analysis of the dependence of feature contributions on walking dynamics during either the input or the output phases of the locomotion prediction algorithm. We were also interested in a comparison of the use of these features for short-term (several frames) vs long-term (several seconds) predictions and examined the influence of different feature combinations on prediction performance. Short-term predictions are useful in VR to calculate the most likely configuration of the body in the scene for the next couple of frames, which can be useful to optimize resource allocation when streaming high-resolution VR content [see 51]. Long-term predictions can be used to estimate the intention of the actor and therefore could enhance applications such as collision-avoidance and redirected walking.

2. Data Acquisition

Our data was obtained from a VR experiment in which 18 participants completed a set of natural locomotion tasks which were designed to include typical behaviors, such as searching for a target object, walking along a curve and avoiding obstacles. To promote natural walking behavior participants were given verbal task instructions instead of defined walking paths. All raw data files are freely available from `https://osf.io/b43uv/`.

2.1. *Procedure*

The virtual environment consisted of two rooms linked by a corridor. The first room contained a target object, which the participant had to search for. This target was placed among six identical looking distractors (see Figure 1a), so that the participants had to perform a search by walking freely between the objects until they found the target. The seven objects were always in the same position (but randomly rotated) and the participants could use the controller to test whether an object was a target. The result was signalled by a coloured light on top of the object and by a sound. The distance between neighbouring objects was 2 meters. For each trial and participant, we pseudo-randomised which of the seven objects was the correct target.

In the other room, the aim was to walk to a target while avoiding a possible obstacle. This room had four different conditions: obstacle centered, obstacle 30 cm to the left, obstacle 30 cm to the right and no obstacle. In that room, the participants first positioned themselves in front of a red button. Pushing the button with the controller made the button disappear, and the target and obstacle appear. The distance between button and target was 4 meters. The obstacle was placed in the middle between the button and the target (see Figure 1b). The participant then walked to the target, avoiding the obstacle. The participant repeated this task four times in each visit to this room, each time with new start positions, targets and obstacles.

The participant changed between rooms by walking through a transition corridor (see Figure 1c). The corridor followed a curve with a radius of 5.5 m. Participants completed a total of 10 trials in each room. Thus, since the participants went back and forth between the rooms, nine left curves and ten right curves were obtained for each participant. The two rooms were mapped onto the same physical space in an impossible spaces scenario [52]. Whenever the participant moved through the transition corridor to the door on the other side, an entry to the room opened on the other side and the interior changed. This was done for practical, not experiment-related reasons.

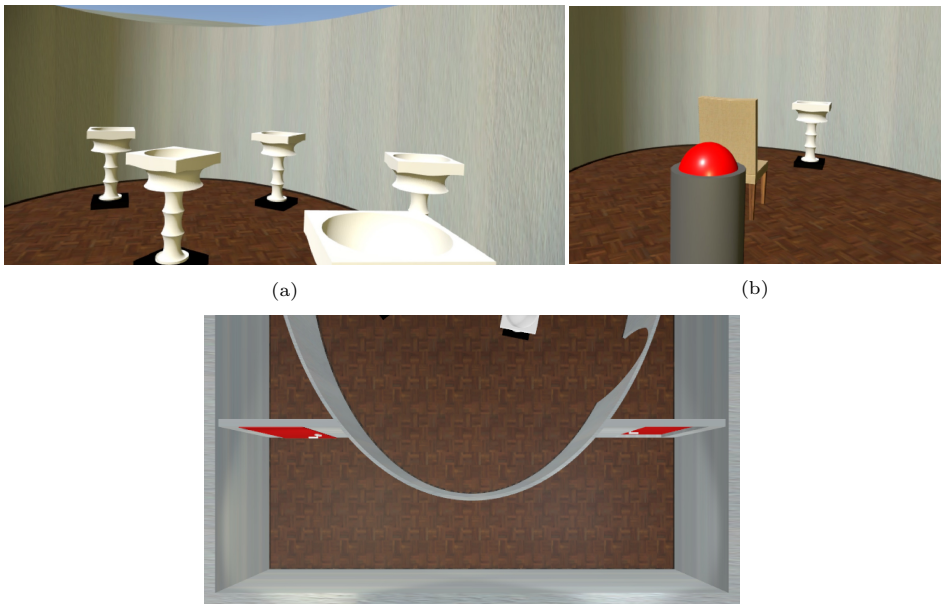
During the experiment, all positional tracking data was Kalman filtered [53]. Before testing, participants were informed about the tasks and were instructed to keep a natural pace. On average, participants needed 14 minutes to complete the data collection experiment.

2.2. *Participants*

Eighteen participants (8 female) completed the experiment. Their age ranged from 20 to 47 years ($M = 27, SD = 6.34$). Participants gave informed written consent and the experimental procedures were approved by the Ethics Committee of the University of Muenster. Two authors participated in the experiment. All other observers were naïve to the purpose of the experiment.

2.3. *Materials*

The virtual environment was presented in an HTC Vive Pro Eye HMD with a resolution of 1440×1600 pixels per eye, a frame rate of 90 Hz and a field of view of 110 degrees. Six Vive Lighthouses 2.0 were used to create a tracking area of 6×11 m. However, to prevent skewed ground planes resulting from signal loss at the borders [54], the outer parts of the tracking area where rarely used during the experiment. The virtual environment was built with Unity3D and was running on an MSI GE63VR 7RF Raider notebook with an NVIDIA GTX1070 graphics card in a backpack. This notebook was supplied with power via a cable hanging from the ceiling. This cable was attached to a rail on the ceiling, which was programmed to prevent the cable from colliding with the participants by moving the attachment.



(a) (b)

(c)

Fig. 1: The different rooms used in the VR data collection. (a) Search room. The room contained seven posts (2 m apart from each other, five posts are visible in this figure) which the user had to inspect to find the target among them. (b) Obstacle avoidance room. In this room the user had to walk from a starting location (red button) to a target post (as in the room above) while avoiding an obstacle (chair). The obstacle and the target were not visible at the beginning. Pushing the red button showed the target and the obstacle. (c) Corridor (birds eye view). The corridor linked the search and obstacle avoidance rooms. In the corridor the user had to walk along a curved path from one room to the other.

The rail was located above the center axis of the room (and also the virtual environment). A Vive tracker was attached to the backpack to measure body orientation, while the tracker in the HMD measured head orientation. A Vive controller was used as the input device. Throughout the experiment, positional and orientation data from all trackers, as well as the gaze position obtained from the eye tracker in the HMD, were recorded.

3. Prediction Model

3.1. *Data Preparation*

For the predictive models, the data was divided into 50-millisecond bins. At a sampling rate just below 90Hz, one bin corresponded to about four frames in the raw data. To form the models' inputs, sequences containing the data at the current timestamp (the time at which the prediction is calculated) and the data of some immediately preceding timestamps were then constructed. The length of the input was set to 2.5 seconds. With a resolution of 50 ms per sample point, this corresponds to a sequence of 50 samples per input. To compensate for asymmetries in the spatial

design of the experiment, every second sequence was mirrored on the XZ-plane.

Due to blinking and the nature of mobile eye trackers, the eye-tracking system was the sensor most susceptible to missing values. To deal with blinks, a single missing value in the eye-tracking data was filled using linear extrapolation based on the previous 3 frames. Data sequences with multiple subsequently missing values were excluded. Additionally, data containing prolonged standing (e.g. at the beginning of the experiment) in the HMD tracking data was excluded using a threshold of 0.15 m/s.

The positional data was output for both the HMD (X_t^H, Y_t^H, Z_t^H) and the body tracker (X_t^B, Y_t^B, Z_t^B). To reduce the complexity of the model, the Y-coordinate (elevation) was removed by projecting the three-dimensional coordinate system of the tracking area to a two-dimensional coordinate system (X_t^B, Z_t^B).

In addition to the position recordings from the room tracking, orientation data provided by the inertial measuring units (IMU) was also included in the models. All orientations are denoted as intrinsic Euler angles roll (Φ), pitch (Θ) and yaw (Ψ). Both the orientation of the HMD ($\Phi_t^H, \Theta_t^H, \Psi_t^H$) and the orientation of the body tracker ($\Phi_t^B, \Theta_t^B, \Psi_t^B$) were recorded.

Lastly, the outputs of the Vive Pro Eye's integrated eye tracker were obtained as yaw and pitch angles ($\Psi_{t-i}^E, \Theta_{t-i}^E$).

3.1.1. *Features*

Seven features were selected for model training at each time point in the sequence: the 2D head velocity ($\vec{V}_{t-i}$), yaw and pitch of the HMD ($\Psi_{t-i}^H, \Theta_{t-i}^H$) 2D gaze direction ($\Psi_{t-i}^E, \Theta_{t-i}^E$) and the yaw angle of the body tracker (Ψ_{t-i}^B). The 2D velocity $\vec{V}_{t-i}$ at each time point $t-i$ in the sequence was calculated relative to the previous time point $t-i-1$.

$$\vec{V}_{t-i} = (V_{t-i}^X, V_{t-i}^Z) = \frac{(X_{t-i}^H - X_{t-i-1}^H, Z_{t-i}^H - Z_{t-i-1}^H)}{50ms} \tag{1}$$

In this equation, the i represents the respective array index in the time sequence on which the input is based. By using velocities, this feature is independent of the coordinate system's origin.

3.1.2. *Labels*

The direction vector $\vec{F}_t$ from the current position at time t to the future position at time t+n was chosen as prediction target. To cover the different aspects of path prediction, we specified two time intervals and evaluated both of them. The time interval for the long-term prediction was set to 2.5 seconds, mirroring the input length. Regarding the short-term prediction, we used the next step of the time sequence (50 ms).

$$\vec{F}_t = (F_t^X, F_t^Z) = (X_{t+n}^H - X_t^H, Z_{t+n}^H - Z_t^H) \tag{2}$$

3.1.3. *Coordinate Systems*

Even though $\vec{F}_t$ and $\vec{V}_t$ depend on the previous positions and are therefore independent of the origin position of the coordinate system, both features and labels are still in a coordinate system defined by the axes of the virtual environment. This is undesirable, since it cannot be assumed that movements are distributed evenly across directions. In fact, the environmental architecture is likely to produce certain movement patterns associated with certain directions (e.g. the curves in the corridor). A major problem with models based on global coordinate systems like this is a lack of transferability of the same motion patterns to other orientations and positions. Therefore, it is necessary to use a relative coordinate system.

Since there is no reason to believe that a single input representation is appropriate for both long-term and short-term predictions, we evaluated two different coordinate systems to be able to select the most suitable one for each time interval. In the following, values in the new coordinate systems will be represented by lowercase letters (e.g. ψ, θ).

Mean Head Orientation Reference System

First, we evaluated a coordinate system using the average head orientation of one sequence as a reference angle.

$$\begin{aligned} \bar{\Psi}_t^R &= \frac{1}{l}\sum_{i=1}^{l} \Psi_{t-i}^H \\ \bar{\Theta}_t^R &= \frac{1}{l}\sum_{i=1}^{l} \Theta_{t-i}^H \end{aligned} \tag{3}$$

In this equation, l refers to the total number of timestamps in the input. The reference angles were identical for all steps in one time sequence and therefore provided a stable coordinate system for each single input-output-pair. In the *Mean Head Orientation Reference System* the features are expressed as:

$$\begin{aligned} \psi_{t-i}^H &= \Psi_{t-i}^H - \bar{\Psi}_t^R \\ \theta_{t-i}^H &= \Theta_{t-i}^H - \bar{\Theta}_t^R \\ \psi_{t-i}^B &= \Psi_{t-i}^B - \bar{\Psi}_t^R \\ \psi_{t-i}^E &= \Psi_{t-i}^E + \psi_{t-i}^H \\ \theta_{t-i}^E &= \Theta_{t-i}^E + \theta_{t-i}^H \end{aligned} \tag{4}$$

Since the eye data is given in the coordinate system of the HMD, it can be offset using the new HMD orientations. Finally, the velocities and labels were transferred to the *Mean Head Orientation Reference System* by point rotations:

$$\begin{aligned} v_{t-i}^x &= \cos(-\bar{\Psi}_t^R) V_{t-i}^X - \sin(-\bar{\Psi}_t^R) V_{t-i}^Z \\ v_{t-i}^z &= \sin(-\bar{\Psi}_t^R) V_{t-i}^X + \cos(-\bar{\Psi}_t^R) V_{t-i}^Z \end{aligned} \tag{5}$$

$$\begin{aligned} f_t^x &= \cos(-\bar{\Psi}_t^R)F_t^X - \sin(-\bar{\Psi}_t^R)F_t^Z \\ f_t^z &= \sin(-\bar{\Psi}_t^R)F_t^X + \cos(-\bar{\Psi}_t^R)F_t^Z \end{aligned} \tag{6}$$

Translational Motion Reference System

In the second approach, the respective direction of movement of the previous step was used as a dynamic reference angle. Accordingly, the last directions of movement were then used as labels. This means that the original pitch angles were preserved. Since the virtual environment's global Y-axis refers to the gravity axis and not to an arbitrary positioning, this is not a problem. In contrast to the *Mean Head Orientation References*, the reference angle differed at each index.

$$\Psi_{t-i}^R = \angle(\overrightarrow{V_{t-i-1}}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}) \tag{7}$$

Labels and features were expressed as:

$$\begin{aligned} \psi_{t-i}^H &= \Psi_{t-i}^H - \Psi_{t-i}^R \\ \theta_{t-i}^H &= \Theta_{t-i}^H \\ \psi_{t-i}^B &= \Psi_{t-i}^B - \Psi_{t-i}^R \\ \psi_{t-i}^E &= \Psi_{t-i}^E + \psi_{t-i}^H \\ \theta_{t-i}^E &= \Theta_{t-i}^E + \theta_{t-i}^H \end{aligned} \tag{8}$$

$$\begin{aligned} v_{t-i}^x &= \cos(-\Psi_{t-i}^R)V_{t-i}^X - \sin(-\Psi_{t-i}^R)V_{t-i}^Z \\ v_{t-i}^z &= \sin(-\Psi_{t-i}^R)V_{t-i}^X + \cos(-\Psi_{t-i}^R)V_{t-i}^Z \end{aligned} \tag{9}$$

$$\begin{aligned} f_t^x &= \cos(-\Psi_{t+1}^R)F_t^X - \sin(-\Psi_{t+1}^R)F_t^Z \\ f_t^z &= \sin(-\Psi_{t+1}^R)F_t^X + \cos(-\Psi_{t+1}^R)F_t^Z \end{aligned} \tag{10}$$

Both coordinate systems were used for models with all features. The coordinate system resulting in the lowest error was then chosen and used for further variations of the model (e.g. fewer features).

3.2. *Model Properties*

Our LSTM model had two layers of 64 hidden units each. The output of the second LSTM layer went through a dropout layer (p = 0.3) [55] resulting in the final linear dense layer with two outputs, one for each label coordinate. Figure 2 depicts this architecture. In total, the model with all features had 51,586 trainable parameters and used adam as the optimizer [56]. The learning rate was set to 0.003 and to prevent overfitting, a weight decay of 1×10^{-4} was applied. The model was trained for 20 epochs using a batch size of 64 and the mean squared error between predicted and label position as the loss function. Then the epoch with the lowest validation error was selected. To obtain a single error value on the meter scale, the mean displacement error (mde) between the true values (labels) and the predictions, i.e. the Euclidean distances between the two-dimensional points, was calculated.

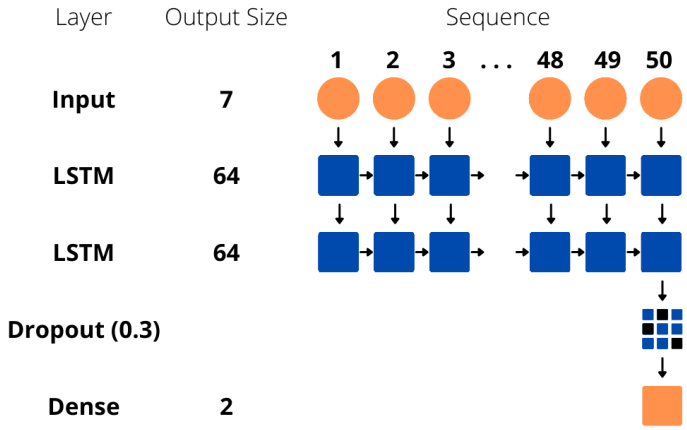


Fig. 2: The model architecture. Seven features in 50 time steps enter the model. The circles represent this input. The following 4 rows marked with squares form the 4 layers of the model. The final dense layer outputs the prediction result.

3.2.1. *Full-feature model and combination models*

The full model included all seven features presented in the data preparation and was used to determine the most appropriate coordinate system for both the long-term and short-term analyses as it contained all the information. To evaluate the contributions of individual features (gaze, position and orientation data) models with different combinations of these features were evaluated.

3.2.2. *Variants of the full-feature model*

In order to obtain a more detailed picture, we also assessed variants of the full long-term model. To assess the contribution of the specific characteristics of the LSTM architecture, we also report a model that uses gated recurrent units (GRUs). Introduced by Cho et al. [57], GRUs are another RNN variant that is similar to the LSTM architecture but reduces the number of parameters. This leads to lower computational costs. GRUs have been utilized in path prediction contexts [58].

To evaluate the possibility of a bidirectional LSTM achieving better results, we tested that as well. Additionally, a widely used approach in sequential forecasting is the prediction of an entire sequence. If sequential predictions were as accurate as single-value predictions, a detailed path could be obtained in place of the future position prediction. We evaluated this option as well by creating a variation of the model that, with an otherwise equivalent architecture, predicts a sequence of 50 position vectors. The labels consisted of a series of vectors that, like $\vec{V}_t$, always contained the information from one step to the next. The loss function was adjusted

accordingly to form the mean squared error between the predicted path at step i and the actual path. The learning rate was lowered to 0.001.

Furthermore, we also created a Bayesian version of the long-term prediction model. Bayesian methods can be used in an attempt to account for uncertainty and thus make more accurate predictions while at the same time calculating an error associated with the specific prediction. In this approach, distributions of weight parameters replace deterministic weights. Our Bayesian network was built with a library by Esposito [59], which is based on the 'Bayes by Backprop' approach introduced by Blundell et al. [60]. The Kullback-Leibler divergence between the model posterior and the observed posterior was added to the loss function. Apart from replacing the deterministic weights, the architecture of the model was kept the same. The hyperparameters were also retained with the exception of the weight decay, which had to be removed as it affects distributions differently than deterministic weights. Standard normal distributions were used as prior distributions.

3.2.3. *Benchmarks*

Since this data had never been evaluated before, cross-validated benchmarks were calculated as a reference. In addition to the mean value of the training data, we used the most recent positions to create an extrapolation benchmark. Yet this comparison is somewhat unfair, as the extrapolation is based on much less data. Therefore, we gave the exact same data into a linear model, in which the time progression of the seven features was flattened, i. e., for each of the 50 time steps, all seven features were used as individual predictors. To evaluate our model, the mde of the best LSTM model was compared to the best benchmark model.

3.3. *Evaluation*

3.3.1. *Cross-Validation*

To avoid overlapping input sequences in the training and test set and to ensure the transferability of a model to new data, cross-validation was implemented at group level. In this process, leave-3-out-cross-validation was used. In each case, the data of one participant was used as validation data and the data of the remaining two as test data generating 6 variations of the model in total. This ensured that the validation data, which was used to evaluate different hyper-parameters, did not factor into the final results. Before training, features and labels were z-standardized. To fit the scalers, only the training set was used while all data was adjusted with these scalers.

3.3.2. *Statistical Testing*

Using this cross-validation approach, individual prediction errors were calculated for each participant and test set. Moreover, to decide whether a model outperforms a reference model, a significance test provides more information than a mere

comparison of average errors. Here we have firstly compared benchmark, GRU and LSTM models and secondly compared the best LSTM model with models that only include a subset of features.

The results of two cross-validated models are based on the exact same data. Hence, the data is paired. Nadeau and Bengio [61] proposed a method to correct for the fact that the individual results of the folds are not independent of one another, since the training sets overlap. Therefore, we used the paired t-test with the correction of Nadeau and Bengio [61]. It should be mentioned that the results of these significance tests need to be treated with caution. Bouckaert and Frank [62] raised concerns about the replicability of test methods like the one used here, which depend on the partitioning of the data in the cross-validation process. The alpha level was set to 0.05. The Benjamini-Hochberg correction [63] was applied to the p-values of a single paragraph to avoid underestimation of the p-value due to multiple testing . All tests were two-sided and the assumption of normally distributed data was tested with a Shapiro-Wilk test [64] beforehand.

4. Results

4.1. *Short-Term Predictions*

For the short-term LSTM prediction the *Translational Motion Reference System* gave a far better result with a mean displacement error of 5.16 millimeters on average (the absolute error was 2.91 mm; the squared error was 4.78 mm^2) compared to the *Mean Head Orientation Reference System* with 9.77 millimeters on average (the absolute error was 5.95 mm; the squared error was 8.75 mm^2). The former gave a more accurate prediction for every participant. Thus, the *Translational Motion Reference System* was used as the coordinate system for all short-term prediction models and benchmarks. Using this method, 151,943 input-output pairs were obtained. Results are presented in Table 1.

Table 1: 50ms prediction

Architecture	Model Features	mde	sd
LSTM	all	5.16 mm	0.65 mm
LSTM	position + orientation	5.14 mm	0.64 mm
LSTM	position + gaze	5.17 mm	0.72 mm
LSTM	orientation + gaze	9.36 mm	1.26 mm
LSTM	position	5.29 mm	0.70 mm
LSTM	orientation	9.57 mm	1.34 mm
LSTM	gaze	10.28 mm	1.96 mm
Bidirectional LSTM	all	5.28 mm	0.63 mm
GRU	all	5.33 mm	0.64 mm
Linear Model	all	6.14 mm	0.82 mm
Interpolation	position	10.45 mm	1.91 mm
Mean	-	16.51 mm	1.53 mm

In 50 milliseconds, the observers traveled 3.59 cm on average. The training mde was 5.17 millimeters for the full model. The mde of the full model, the model using position and orientation and the model using position and gaze were almost identical with 5.16 mm, 5.14 mm and 5.17 mm respectively. The mde of the model only using positional data was also close with 5.29 mm. For the full model, the null hypothesis that the data is normally distributed was rejected ($W = 0.75, p = 0.02$). This was also true for the model using position and gaze ($W = 0.73, p = 0.01$). Since the model with position and orientation data performed best, we compared the other models to this one. The difference between the model using position and gaze and the model using position and orientation failed to reach statistical significance ($t(5) = -1.93, p = 0.11$). However, a significant difference was achieved when comparing the model with position and orientation data to the model using orientation and gaze data ($t(5) = -9.22, p < 0.001$), orientation data only ($t(5) = -10.60, p < 0.001$) and gaze data only ($t(5) = -6.13, p = 0.002$).

All in all, the errors of the LSTM short-term models that used positional data were quite similar. Models that only used gaze or orientation data, thus omitting the position data, performed substantially worse with 10.28 mm and 9.57 mm respectively and 9.36 mm for the combination of both.

Compared to all of the benchmark models, the LSTM models provided better predictions for each test set and each participant. The difference between the best LSTM model and the best benchmark model (linear model) reached statistical significance ($t(5) = -8.73, p < 0.001$). Nevertheless, the linear model was only one millimeter worse than the LSTM on average.

Table 2: 2.5s prediction

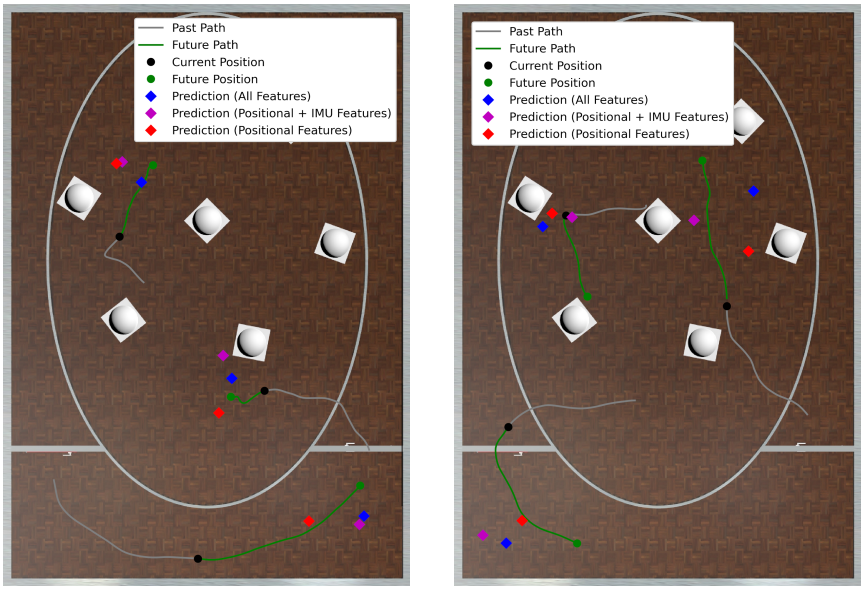
Architecture	Model Features	mde	sd
LSTM	all	65.73 cm	5.12 cm
LSTM	position + gaze	66.71 cm	5.69 cm
LSTM	position + orientation	67.56 cm	5.46 cm
LSTM	gaze + orientation	78.05 cm	7.92 cm
LSTM	gaze	78.19 cm	8.71 cm
LSTM	position	78.38 cm	6.77 cm
LSTM	orientation	81.07 cm	6.97 cm
GRU	all	66.17 cm	6.01 cm
Bidirectional LSTM	all	66.17 cm	4.83 cm
Sequence-to-Sequence LSTM	all	77.65 cm	5.50 cm
Linear Model	all	92.52 cm	8.09 cm
Interpolation	position	131.09 cm	16.16 cm
Mean	-	144.72 cm	14.65 cm

4.2. *Long-Term Predictions*

For the long-term prediction, the *Mean Head Orientation Reference System* proved superior with a mean displacement error of 65.73 centimeters on average (the absolute error was 41.74 cm; the squared error was 55.90 cm^2) compared to the *Translational Motion Reference System* with 68.85 centimeters (the absolute error was 43.87 cm; the squared error was 58.49 cm^2). The *Mean Head Orientation Reference System* gave a more accurate prediction for each participant. Thus, the *Mean Head Orientation Reference System* was used as the coordinate system for all long-term prediction models and benchmarks. Results are presented in Table 2.

The 50-sample input sequences and prediction labels formed 156,076 input-output pairs in total. The participants traveled a mean distance of 165.28 cm per output length of 2.5 seconds. The average walking speed was 0.72 m/s. For the full model, the training mde was 58.82 cm.

The models using all features (mde = 65.73 cm), only position and gaze (mde = 66.71 cm), and only position and orientation (mde = 67.56 cm) were all very close in performance. The model using only gaze and orientation data (mde = 78.05 cm) and the models that used only one source of data performed worse (mde = 78.19 cm for gaze, mde = 78.38 cm for position data and mde = 81.07 cm for orientation data). A significant difference was achieved when comparing the full model with to the



(a) (b)

Fig. 3: Example paths taken by the user and prediction derives from the model. a: Tree paths where the prediction error (all features) was above the 25 % quantile but below the 75 % quantile. b: Examples in which the prediction failed. The prediction error (all features) was above the 75 % quantile.

model using orientation and gaze ($t(5) = -8.87, p = 0.002$), orientation data only ($t(5) = -7.48, p = 0.002$), position data only ($t(5) = -6.99, p = 0.002$) and gaze data only ($t(5) = -5.78, p = 0.003$). The difference between the errors of the full model and the model using position and gaze data was not statistically significant ($t(5) = -1.56, p = 0.179$). Although the difference between the model using position and orientation data and the model and the full model that also used gaze data reached statistical significance ($t(5) = -3.01, p = 0.036$), it has to be noted that the mde in the full model is only 2.78% smaller. Given the size of this difference, the aforementioned caution in interpreting significance tests is particularly important here.

Regarding the full model, the errors varied substantially. On average, the top 25 % of the prediction errors were over 89.82 cm, including the top 10 % over 127.41 cm. While the lowest 25 % of the prediction errors fell below 32.21cm, including the lowest 10 % below 18.71cm on average (see Figure 3 for examples).

4.3. *Analysis of Feature Dependence on Locomotor Parameters*

We next investigated whether the contribution of different features to the prediction depended on the dynamics of locomotion. Figure 4a shows how the prediction error varies with the distance that the participants walked during the 2.5 seconds of the input sequence, i.e., the history of the walk. Overall, prediction quality was better for longer distances, however, in a medium distance range of around 1.5 meters quality dropped as a slightly larger error showed. This dependence on travel distance was similar for all tested models and the general better performance of the full model, the position and gaze model and the position and orientation model over the other

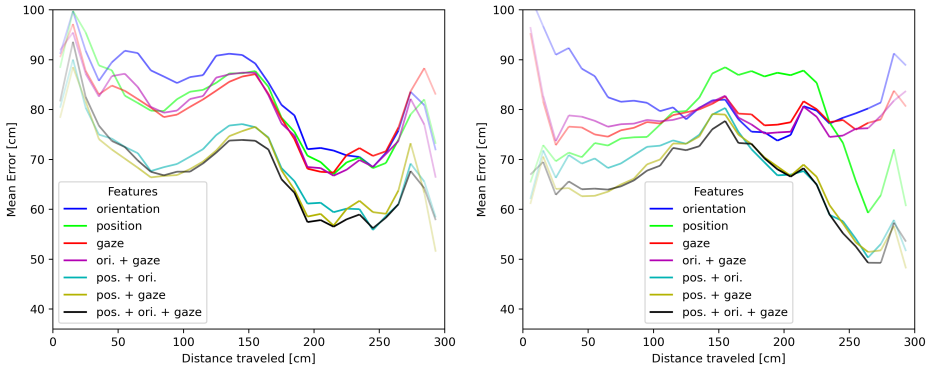
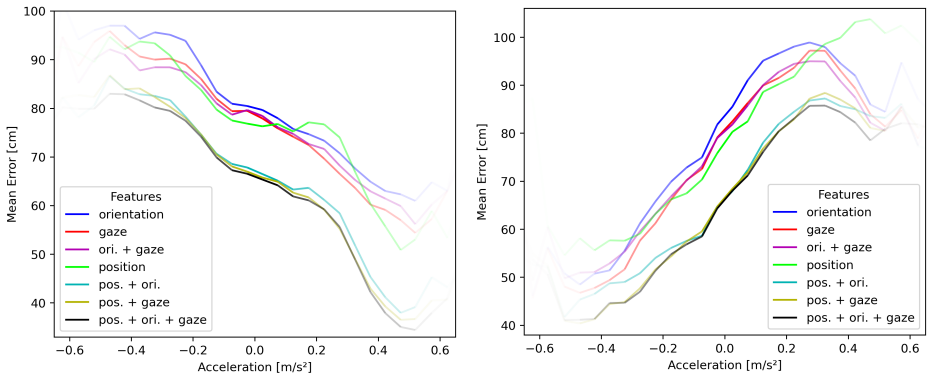


Fig. 4: a: The mdes of models using different sets of features as a function of the distance that the user walked during the 2.5 seconds used as input data. b: The mdes of the different models as a function of the distance that the user walked during the 2.5 seconds used as label data, i.e, the distance that needed to be predicted. Line transparency indicates the number of observations that factored into this data point.



(a) (b)

Fig. 5: The mdes of models using different sets of features as a function of the acceleration at which the user moved during the 2.5 seconds used as input data in a) and during the 2.5 seconds used as label data in b). Line transparency indicates the number of observations that factored into this data point.

models was preserved for all distances.

Figure 4b shows the dependence of the prediction error on the distance that the participants walked during the 2.5 seconds of the output sequence. This graph thus shows how the prediction quality varied with the to-be-predicted distance, i.e., the distance of the walk that needed to be predicted. This Figure also illustrates that performance was better for long distances and a particularly large error occurred for medium distances. Similar to Figure 4a, the full model, the position and gaze model and the position and orientation model maintained comparable performance advantages over the other models at all distances. The model using only positional information showed a large error at intermediate distances, suggesting that in this range the inclusion of other features was particularly necessary. Regarding the value of adding gaze data, we can compare the full model with the model that uses position and orientation. Figure 4a shows that the error is the same for medium and long distances, but for short distances the full model gives a lower prediction error. Between a distance of 50 cm and 60 cm, the difference reached 9.33%.

Figure 5 shows how the prediction error varies with the acceleration contained in the input (a) and output (b) sequences. Error is lowest, i.e. performance is best, for input sequences that contain high accelerations and output sequences that contain high decelerations. Together, this suggests that performance is best if the prediction is taken at a velocity peak. Again, the full model, the position and gaze model and the position and orientation model maintained their performance benefits over the other models at all times. Regarding the value of adding gaze data, figure 5b shows a small difference between the full model and the position and orientation model for high decelerations. When comparing these two models, the error difference reached up to 12.61% between -0.5 m/s^2 and -0.4 m/s^2.

4.4. *Analysis of Model Variants*

We also examined a number of variants of our model architecture to see whether they might produce better performance or whether similar performance might be produced with a simpler model. To examine whether a model with lower computational costs is sufficient for our task, we calculated a GRU model using all features. Between the LSTM and GRU architectures, no significant difference was found for either long-term predictions ($t(5) = -0.73, p = .50$) or short-term predictions ($t(5) = -1.13, p = .31$). Thus, GRU, which is a somewhat simpler model, was quite comparable in performance.

We also tested whether a bidirectional LSTM could outperform the regular LSTM. This was not the case for either time interval. Since a bidirectional LSTM has a more elaborate architecture, it does not add any value to our application, unlike the GRU, which reduces costs.

Next, we evaluated a Bayesian model, which has the additional advantage that it allows to estimate the uncertainty of a single prediction. When testing a Bayesian LSTM model, ten predictions were sampled per input. Although the model performed nominally better than the full model (65.19 cm), the improvement failed to reach statistical significance ($t(5) = 0.81, p = 0.45$).

We also looked at model predictions in a sequence-to-sequence approach. For the sequence-to-sequence approach, 117,254 input-output pairs were obtained. At 77.65cm, the error at the last position was significantly larger compared to a model only predicting the final position ($t(5) = -16.68, p < 0.001$). Therefore, it is only worthwhile to follow this approach if it is necessary to predict the output sequence as a whole.

5. Discussion

We presented an extension of our previous work [50] on multiple trajectory prediction models trained on locomotion data obtained in a free walking VR setup. These models aim to predict the position where a walker intends to go based on the immediate feature history of the walker's position, orientation and gaze. We evaluated the prediction quality of different model variants using different architectures, timescales, coordinate systems and different sets of features. First, we will summarize the prediction results from our models. Then, we will discuss the influence of features and the choice of coordinate system. Lastly, we will discuss possible application scenarios for the prediction method.

5.1. *Model Architecture*

The prediction model with the lowest error was an LSTM model. Trained using all available model features, it was able to provide successful predictions of future positions and outperformed all of our benchmark models. This was especially noticeable in long-term predictions (mde = 66 cm) of positions after 2.5s. For short-term

predictions (mde = 5 mm) of the next 50 ms, the LSTM model outperformed our benchmark models only slightly. However, in both cases the results of the full feature GRU model indicate that a more efficient architecture with lower computational cost might be sufficient. The Bayesian LSTM could not significantly outperform its deterministic counterpart. Thus, although the Bayesian model determined the average over 10 independent runs, these multiple predictions did not improve the estimate. However, if one is willing to accept the higher computational cost, the Bayesian model is useful to obtain a simultaneous estimate of certainty in the prediction.

Using a sequence-to-sequence prediction was less successful. Naturally, the need for 100 subsequent sample points without a missing value for this approach, lowered the amount of available training data in comparison to the other models. Additionally, the amount of data might have been further reduced slightly through short tracking errors. These occurred rarely in some participants, although we did not use the maximum size of the nominal tracking area of the Vive tracking system. Using a higher amount of training data, a more sophisticated loss function, or a more complex network architecture may allow an improved sequence-to-sequence prediction, but were not further analyzed in this study.

5.2. *Coordinate Systems for short-term and long-term predictions*

We compared two types of coordinate systems, one based on mean head orientation, the other based on the current direction of motion. Our results showed that the different coordinate systems were differently suited to the two prediction time periods. The *Mean Head Orientation Reference System* led to better predictions for the long-term prediction, while the *Translational Motion Reference System* achieved lower errors in the short-term LSTM prediction. Although the underlying information was equal in the two reference systems, since both used the same set of base features, some transformations are necessary to transform the data from one coordinate system to the other. A model with many interconnections and many layers might learn such transformations and perform equally well independent of the coordinate system. However, to prevent overfitting, creating an appropriate coordinate system during preprocessing is a more effective approach. Based on our results, it seems beneficial to use a motion-based reference when predicting positions for the next few frames. A head orientation based reference seems better suited when estimating long-term positions. One explanation for these results might be that for short-term prediction the motion direction of the user is basically constant and changes only little. Thus, a reference system based on current motion will provide only small deviations and hence allows efficient predictions. For long-term predictions, motion directions are likely to change as the user turns within the room and a reference system based on the orientation of the user is better suited.

5.3. *Contribution of Gaze, Orientation and Position Feature*

Regarding the set of features of the 2.5 second prediction, the results showed that models using a combination of position and gaze or position and orientation provided predictions of similar quality as the full model using all three features. This suggests that either gaze or orientation data is an especially useful addition to the positional data for the prediction. This fits with previous observations regarding the relationship of head and trunk orientation during locomotion steering [65] and the benefit of gaze data [25], particularly in situations in which participants interact with the virtual environment during locomotion [66]. Notably, our findings indicate that gaze offers the greatest predictive utility when predicting short walking distances (see Figure 4b), or decelerated movements (see Figure 5b). One reason for this result could be that participants used their gaze to plan their foot placement [see 33]. However, it is also possible that gaze data contained valid information regarding stopping or search behavior at slow velocities.

The results in Figure 4 also showed that longer trajectories (beyond 1.5m) based on a faster walking pace led to lower prediction errors. One explanation could be that longer trajectories were less bent and therefore only the walking distance needed to be estimated. To follow-up on this, we estimated path bending by dividing each path into two segments of equal duration and determined the absolute angle between the start and ending positions of each segment (0 degree for a straight path, higher values for more bending). Indeed, for paths longer than 0.5 m, the distance traveled in the labels correlates with bending at $r = -0.442$ on average.

5.4. *Prediction Without Knowledge of the Environment*

The features we used for our prediction models were features of the users' locomotion and orientation of the body and eyes. These are all egocentric features. Our feature set did not contain information about the environment. While one might expect that the addition of environmental features would improve the prediction ability of our models, we purposefully restricted our analysis to the egocentric features since we aimed to produce a system that can predict locomotion in any environment in a general way. The different tasks (searching for a target, walking along a curve and avoiding obstacles) were designed to include multiple typical, natural behaviors. Since our model does not use the layout of the environment, it can be applied to other VR and even non-VR environments (given accurate measurements of the input features), e.g., augmented or extended reality, where environmental information is difficult to obtain.

Due to the choice of tasks in our data set, certain movements are likely represented disproportionately. That might limit the transferability of the model to other movement situations. This needs to be studied in more detail in the future. Another focus of future work could be the addition of moving objects, such as walking avatars, that would likely elicit distinct interactions with eye movements.

5.5. *Possible Applications*

The low computation time of the finished models on current hardware allows their usage in different online applications. For example, short-term prediction of the position of a user in the next couple of frames could be used to enhance techniques that reduce the resolution or level of detail of streamed VR content [e.g. 51]. Long-term locomotion prediction could be useful for immersive environments that allow real walking. In these applications the method could be helpful for early detection of potential collisions with other moving objects in the virtual environment. Furthermore, it could be useful for optimizing redirected walking algorithms such as [e.g. 16]. RDW controllers need to decide at any given time whether and how much to redirect the user. Knowledge of where the user most likely intends to go can be advantageous and speed up RDW controllers [67]. With a prediction error of 65.73 cm, the model is not exact, but an estimate accurate to the centimeter is not necessary for redirected walking. It may be possible to achieve a more accurate prediction in future models by applying a moving average to a time series of predictions while walking.

A further interesting finding was that a model using only gaze data performed quite well (mde 78.19 cm) even without position and movement data. Thus, it seems to be possible to predict a users intended locomotor action directly from observing gaze data. Such a prediction may be useful in human computer interfaces for disabled persons, for example, to allow intuitive control of an assistive robot or navigate an automated wheelchair.

6. Conclusion

We reported on deep learning prediction of a walker's intended future position using a sequence of prior position, orientation, and gaze data. We showed that a model using the LSTM architecture can be used to predict walking paths in VR. Moreover, our results suggest that gaze data provides an advantage for this task, especially regarding short distances in long-term predictions.

7. Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This work was supported by the German Research Foundation (DFG La 952-4-3, La 952-7) and has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951910.

Acknowledgements

The authors would like to thank Nils Winter and Krischan Koerfer for their support.

References

[1] P. Basili, M. Sağlam, T. Kruse, M. Huber, A. Kirsch and S. Glasauer, Strategies of locomotor collision avoidance, *Gait & Posture* **37**(3) 385–390 (2013).

[2] C. G. Keller and D. M. Gavrila, Will the pedestrian cross? a study on pedestrian path prediction, *IEEE Transactions on Intelligent Transportation Systems* **15**(2) 494–506 (2013).

[3] H. S. Koppula and A. Saxena, Anticipating human activities using object affordances for reactive robotic response, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(1) 14–29 (2015).

[4] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater and F. P. Brooks Jr, Walking >walking-in-place >flying, in virtual environments, in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (Los Angeles, CA, USA, 1999), pp. 359–364.

[5] E. Langbehn, P. Lubos and F. Steinicke, Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking, in *Proceedings of the Virtual Reality International Conference-Laval Virtual* (Laval, France, 2018), pp. 1–9.

[6] M. A. Zmuda, J. L. Wonser, E. R. Bachmann and E. Hodgson, Optimizing constrained-environment redirected walking instructions using search techniques, *IEEE Transactions on Visualization and Computer Graphics* **19**(11) 1872–1884 (2013).

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

[8] F. Steinicke, G. Bruder, L. Kohli, J. Jerald and K. Hinrichs, Taxonomy and implementation of redirection techniques for ubiquitous passive haptic feedback, in *2008 International Conference on Cyberworlds* IEEE, (Hangzhou, China, 2008), pp. 217–223.

[9] J. Martinez, M. J. Black and J. Romero, On human motion prediction using recurrent neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI, USA, 2017), pp. 2891–2900.

[10] E. Corona, A. Pumarola, G. Alenya and F. Moreno-Noguer, Context-aware human motion prediction, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA, USA, 2020), pp. 6992–7001.

[11] Y. Tang, L. Ma, W. Liu and W. Zheng, Long-term human motion prediction by modeling motion context and enhancing motion dynamic, *arXiv preprint arXiv:1805.02513* (2018).

[12] H.-S. Moon and J. Seo, Prediction of human trajectory following a haptic robotic guide using recurrent neural networks, in *2019 IEEE World Haptics Conference (WHC)* IEEE, (Tokyo, Japan, 2019), pp. 157–162.

[13] A. Breuer, S. Elflein, T. Joseph, J.-A. Bolte, S. Homoceanu and T. Fingscheidt, Analysis of the effect of various input representations for lstm-based trajectory prediction, in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* IEEE, (Auckland, NZ, 2019), pp. 2728–2735.

[14] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation* **9**(8) 1735–1780 (1997).

[15] Y.-H. Cho, D.-Y. Lee and I.-K. Lee, Path prediction using lstm network for redirected walking, in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* IEEE, (Reutlingen, Germany, 2018), pp. 527–528.

[16] D.-Y. Lee, Y.-H. Cho and I.-K. Lee, Real-time optimal planning for redirected walking using deep q-learning, in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* IEEE, (Osaka, Japan, 2019), pp. 63–71.

[17] S. Razzaque, Z. Kohn and M. C. Whitton, Redirected Walking, in *Eurographics 2001 - Short Presentations* (Eurographics Association, Manchester, UK, 2001).

[18] G. Bruder, F. Steinicke, B. Bolte, P. Wieland, H. Frenz and M. Lappe, Exploiting perceptual limitations and illusions to support walking through virtual environments in confined physical spaces, *Displays* **34**(2) 132–141 (2013).

[19] F. Steinicke, G. Bruder, J. Jerald, H. Frenz and M. Lappe, Estimation of detection thresholds for redirected walking techniques, *IEEE Transactions on Visualization and Computer Graphics* **16**(1) 17–27 (2009).

[20] M. Land and B. Tatler, *Locomotion on foot*, in *Looking and Acting: Vision and eye movements in natural behaviour*, (Oxford University Press, 07 2009), pp. 100–115.

[21] M. F. Land and M. Hayhoe, In what ways do eye movements contribute to everyday activities?, *Vision Research* **41**(25-26) 3559–3565 (2001).

[22] M. Hayhoe and D. Ballard, Eye movements in natural behavior, *Trends in Cognitive Sciences* **9**(4) 188–194 (2005).

[23] A. Belardinelli, M. Y. Stepper and M. V. Butz, It's in the eyes: Planning precise manual actions before execution, *Journal of Vision* **16** 18–18 (01 2016).

[24] J. Gandrud and V. Interrante, Predicting destination using head orientation and gaze direction during locomotion in vr, in *ACM Symposium on Applied Perception, SAP 2016* Association for Computing Machinery, Inc, (Anaheim, CA, USA, 2016), pp. 31–38.

[25] M. Zank and A. Kunz, Eye tracking for locomotion prediction in redirected walking, in *2016 IEEE Symposium on 3D User Interfaces (3DUI)* IEEE, (Greenville, SC, USA, 2016), pp. 49–58.

[26] M. Zank and A. Kunz, Where are you going? using human locomotion models for target estimation, *The Visual Computer* **32**(10) 1323–1335 (2016).

[27] M. A. Hollands, A. E. Patla and J. N. Vickers, "look where you're going!": gaze behaviour associated with maintaining and changing the direction of locomotion, *Experimental Brain Research* **143**(2) 221–230 (2002).

[28] S. Durant and J. M. Zanker, The combined effect of eye movements improve

head centred local motion information during walking, *PLOS ONE* **15**(1) p. e0228345 (2020).

[29] M. A. Hollands, D. E. Marple-Horvat, S. Henkes and A. K. Rowan, Human eye movements during visually guided stepping, *Journal of Motor Behavior* **27**(2) 155–163 (1995).

[30] M. A. Hollands and D. E. Marple-Horvat, Visually guided stepping under conditions of step cycle-related denial of visual information, *Experimental Brain Research* **109**(2) 343–356 (1996).

[31] D. Calow and M. Lappe, Efficient encoding of natural optic flow, *Network Comput. Neural Syst.* **19**(3) 183–212 (2008).

[32] B. M. 't Hart and W. Einhauser, Mind the step: complementary effects of an implicit task on eye and head movements in real-life gaze allocation, *Experimental Brain Research* **223**(2) 233–249 (2012).

[33] J. S. Matthis, J. L. Yates and M. M. Hayhoe, Gaze and the control of foot placement when walking in natural terrain, *Current Biology* **28**(8) 1224–1233 (2018).

[34] D. S. Marigold and A. E. Patla, Visual information from the lower visual field is important for walking across multi-surface terrain, *Experimental Brain Research* **188**(1) 23–31 (2008).

[35] R. Grasso, P. Prévost, Y. P. Ivanenko and A. Berthoz, Eye-head coordination for the steering of locomotion in humans: an anticipatory synergy, *Neuroscience Letters* **253**(2) 115–118 (1998).

[36] T. Imai, S. T. Moore, T. Raphan and B. Cohen, Interaction of the body, head, and eyes during walking and turning, *Experimental Brain Research* **136**(1) 1–18 (2001).

[37] J. Wiener, O. De Condappa and C. Holscher, Do you have to look where you go? gaze behaviour during spatial decision making, in *Proceedings of the Annual Meeting of the Cognitive Science Society* **33**, (Boston, MA, USA, 2011).

[38] D. Kit, L. Katz, B. Sullivan, K. Snyder, D. Ballard and M. Hayhoe, Eye movements, visual search and scene memory, in an immersive virtual environment, *PLOS ONE* **9** 1–11 (04 2014).

[39] B. W. Tatler and S. L. Tatler, The influence of instructions on object memory in a real-world setting, *Journal of Vision* **13** 5–5 (02 2013).

[40] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik and A. Torralba, Eye tracking for everyone, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, 2016), pp. 2176–2184.

[41] J. L. Louedec, T. Guntz, J. L. Crowley and D. Vaufreydaz, Deep learning investigation for chess player attention prediction using eye-tracking and game data, in *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado, 2019), pp. 1–9.

[42] S. C. Koorathota, K. Thakoor, P. Adelman, Y. Mao, X. Liu and P. Sajda,

Sequence models in eye tracking: Predicting pupil diameter during learning, in *ACM Symposium on Eye Tracking Research and Applications* (New York; NY; USA, 2020), pp. 1–3.

[43] L. Theis, I. Korshunova, A. Tejani and F. Huszár, Faster gaze prediction with dense networks and fisher pruning, *arXiv Preprint arXiv:1801.05787* (2018).

[44] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu and S. Gao, Gaze prediction in dynamic 360 immersive videos, in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT, USA, 2018), pp. 5333–5342.

[45] Y. Huang, M. Cai, Z. Li and Y. Sato, Predicting gaze in egocentric video by learning task-dependent attention transition, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 754–769.

[46] H. R. Tavakoli, E. Rahtu, J. Kannala and A. Borji, Digging deeper into egocentric gaze prediction, in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* IEEE, (Waikoloa Village, HI, USA, 2019), pp. 273–282.

[47] M. Cornia, L. Baraldi, G. Serra and R. Cucchiara, Predicting human eye fixations via an lstm-based saliency attentive model, *IEEE Transactions on Image Processing* **27**(10) 5142–5154 (2018).

[48] G. Arechavaleta, J.-P. Laumond, H. Hicheur and A. Berthoz, An optimality principle governing human walking, *IEEE Transactions on Robotics* **24**(1) 5–14 (2008).

[49] P. W. Fink, P. S. Foo and W. H. Warren, Obstacle avoidance during walking in real and virtual environments, *ACM Transactions on Applied Perception (TAP)* **4**(1) 2–es (2007).

[50] G. Bremer, N. Stein and M. Lappe, Predicting future position from natural walking and eye movements with machine learning, in *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)* IEEE, (Taichung, Tawian, 2021), pp. 19–28.

[51] Y. Zhu, G. Zhai and X. Min, The prediction of head and eye movement for 360 degree images, *Signal Processing: Image Communication* **69** 15–25 (2018).

[52] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum and M. Bolas, Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture, *IEEE Transactions on Visualization and Computer Graphics* **18**(4) 555–564 (2012).

[53] R. E. Kalman, A new approach to linear filtering and prediction problems, *Journal of Basic Engineering* (1960).

[54] D. C. Niehorster, L. Li and M. Lappe, The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research, *i-Perception* **8**(3) p. 2041669517708205 (2017).

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *The Journal*

of Machine Learning Research **15**(1) 1929–1958 (2014).

[56] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv Preprint arXiv:1412.6980* (2014).

[57] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, *arXiv Preprint arXiv:1406.1078* (2014).

[58] E. A. Pool, J. F. Kooij and D. M. Gavrila, Context-based cyclist path prediction using recurrent neural networks, in *2019 IEEE Intelligent Vehicles Symposium (IV)* IEEE, (Paris, France, 2019), pp. 824–830.

[59] P. Esposito, Blitz - bayesian layers in torch zoo (a bayesian deep learing library for torch) `https://github.com/piEsposito/blitz-bayesian-deep-learning/`, (2020).

[60] C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, Weight uncertainty in neural network, in *International Conference on Machine Learning* PMLR, (Lille, France, 2015), pp. 1613–1622.

[61] C. Nadeau and Y. Bengio, Inference for the generalization error, *Machine Learning* **52**(3) 239–281 (2003).

[62] R. R. Bouckaert and E. Frank, Evaluating the replicability of significance tests for comparing learning algorithms, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining* Springer, (Sydney, NSW, Australia, 2004), pp. 3–12.

[63] Y. Benjamini and Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society: Series B (Methodological)* **57**(1) 289–300 (1995).

[64] Shapiro and Wilk, An analysis of variance test for normality (complete samples), *Biometrika* **52** 591–611 (12 1965).

[65] G. Courtine and M. Schieppati, Human walking along a curved path. i. body trajectory, segment orientation and the effect of vision, *European Journal of Neuroscience* **18**(1) 177–190 (2003).

[66] N. Stein, G. Bremer and M. Lappe, Eye tracking-based lstm for locomotion prediction in vr, in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces* IEEE, (Christchurch, New Zealand, in press).

[67] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, F. Steinicke and E. S. Rosenberg, 15 years of research on redirected walking in immersive virtual environments, *IEEE Computer Graphics and Applications* **38**(2) 44–56 (2018).