In press in: IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)
Digital Object Identifier: 10.1109/AIVR52153.2021.00013

Predicting Future Position From Natural Walking and Eye Movements with Machine Learning

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

***Abstract*—The prediction of human locomotion behavior is a complex task based on data from the given environment and the user. In this study, we trained multiple machine learning models to investigate if data from contemporary virtual reality hardware enables long- and short-term locomotion predictions. To create our data set, 18 participants walked through a virtual environment with different tasks. The recorded positional, orientation- and eye-tracking data was used to train an LSTM model predicting the future walking target. We distinguished between short-term predictions of 50ms and long-term predictions of 2.5 seconds. Moreover, we evaluated GRUs, sequence-to-sequence prediction, and Bayesian model weights. Our results showed that the best short-term model was the LSTM using positional and orientation data with a mean error of 5.14 mm. The best long-term model was the LSTM using positional, orientation and eye-tracking data with a mean error of 65.73 cm. Gaze data offered the greatest predictive utility for long-term predictions of short distances. Our findings indicate that an LSTM model can be used to predict walking paths in VR. Moreover, our results suggest that eye-tracking data provides an advantage for this task.**

***Keywords*-LSTM, Virtual Reality, Eye Tracking, Locomotion, Path prediction, Machine Learning, Gaze**

I. Introduction

When we see people walk, we can infer their future position 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 by 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]. Additionally, locomotion prediction can 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 rather new approach is the use of artificial neural networks. To process sequential data, a common, if not the most established method in this field is the use of recurrent neural networks (RNNs). RNNs share parameters over a sequence instead of treating every observation differently. 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 the usage of Long Short-Term Memory networks (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 VR users paths can be imperceptibly manipulated to make maximum use of the given physical space [17, 18, 19].

Deep learning has also been used to predict eye-related parameters such as pupil diameter and fixation targets [e.g. 20, 21, 22]. Typically, these analyses were focusing on the analysis of the visual stimuli shown to user and thus either used Convolutional Neural Network (CNN) [e.g. 23] or combinations of CNN and RNN features [e.g. 24, 25, 26]. However, Cornia et al. [27] 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) to train a model, it is also possible to base the analysis on the subjects behavior, which allows applying the same model on other environments that do not share the spatial arrangement of the data collection experiment.

In 2016 Zank & Kunz used eye tracking to develop an algorithm to predict one of two locomotion targets, assuming that gaze behavior precedes the direction of human walking [28]. Indeed, there is a body of evidence supporting this notion [e.g. 29, 30, 31, 32, 33]. Wiener et al. [34] even went a step further and concluded that action preparation requires a change of attention, accompanied by a change of gaze direction, when the decision-relevant information was dissociated from the required direction of movement.

Accordingly, the findings by Zank & Kunz indicated that their predictions based on gaze data were superior in some cases. While rough long-term predictions as well as accurate short-term predictions were possible without the addition of eye-tracking data, it was valuable for long-term predictions in a narrow environment [28]. 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 reduce wasting resources when e.g. streaming high-resolution VR content [see 35]. 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.

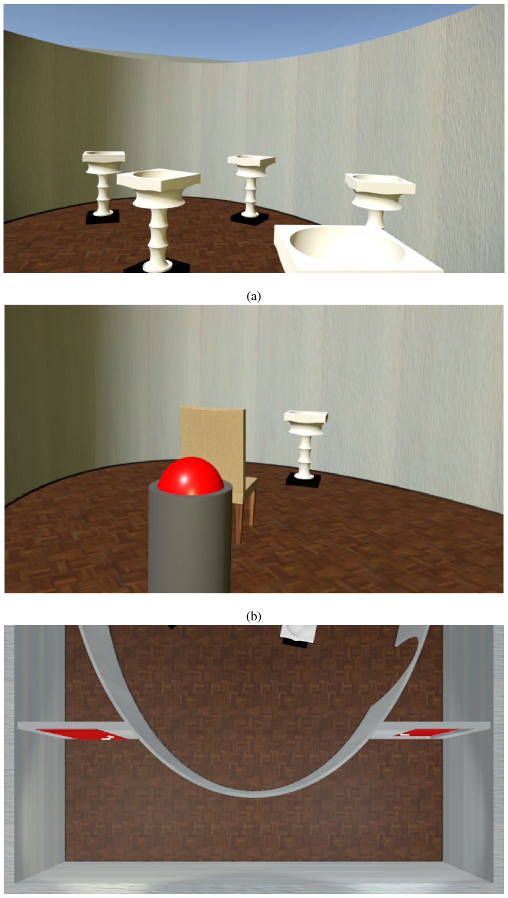
In this study, we created a machine learning locomotion path prediction model using VR position, orientation, and eye-tracking data. We examined the influence of the different features on prediction performance and were especially interested in a comparison of the use of this data for short-term (several frames) vs long-term (several seconds) predictions.

II. 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 subjects were given verbal task instructions instead of defined walking paths. All raw data files are freely available from https://osf.io/b43uv/.

A. *Procedure*

The virtual environment consisted of two rooms linked by a corridor. The rooms contained target objects, which the participant had to search for. In one room the target was placed among six identical looking distractors (see Figure 1a), so that the participant had to perform a search amongst distractors by walking freely between them until she found the target. The other room had four different conditions: obstacle centered, obstacle 30cm to the left, obstacle 30cm 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 the 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 repeated this task four times for 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. The corridor followed a curve with a radius of 5.5 m. Subjects 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 subject. The two rooms were mapped onto the same physical space (impossible spaces scenario) [36]. Whenever the subject 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 [37]. Before testing, subjects were informed about the tasks and were instructed to keep a natural walking speed during the data collection. On average, subjects needed 14 minutes to complete the experiment.


(a)

(b)

(c)

Figure 1: (a) Search task. 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 task. 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) The two rooms were linked by a corridor, in which the user had to walk along a curved path from one room to the other. The corridor is shown from a bird's eye view.

B. *Participants*

18 subjects (8 female) completed the experiment. The subjects' ages 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 Münster. Two authors participated

in the experiment. All other observers were naïve to the purpose of the experiment.

C. Materials

The virtual environment was presented on an HTC Vive Pro Eye 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. The experiment was built with Unity3D and was running on an MSI GE63VR 7RF Raider notebook with an NVIDIA GTX1070 graphics card in a backpack. A Vive tracker was attached to the backpack to measure body orientation independently of the HMD. A Vive controller was used as the input device. Throughout the experiment, positional and orientation data from all trackers, as well as the outputs from the integrated eye tracker, were recorded.

III. Prediction Model

A. 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 50ms 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 of 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 ($\Psi_{t-i}^E, \Theta_{t-i}^E$).

1) Features: All in all 7 features were selected. In addition to the two-dimensional head velocity ($\vec{V}_{t-i}$), yaw and pitch of the HMD ($\Psi_{t-i}^H, \Theta_{t-i}^H$) and gaze direction ($\Psi_{t-i}^E, \Theta_{t-i}^E$) as well as the yaw angle of the body tracker (Ψ_{t-i}^B) were included.

The current two-dimensional velocity $\vec{V}_{t-i}$ was calculated relative to the previous frame. By using velocities, the information is independent of the coordinate system's origin.

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

In this equation, the i represents the respective array index in the time sequence on which the input is based.

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) 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 present 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).

First, we evaluated a coordinate system using the average head orientation of one sequence as a reference angle (*Mean Head Orientation Reference System*).

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

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:

$$\begin{aligned}\psi^H_{t-i} &= \Psi^H_{t-i} - \bar{\Psi}^R_t\\ \theta^H_{t-i} &= \Theta^H_{t-i} - \bar{\Theta}^R_t\\ \psi^B_{t-i} &= \Psi^B_{t-i} - \bar{\Psi}^R_t\\ \psi^E_{t-i} &= \Psi^E_{t-i} + \psi^H_{t-i}\\ \theta^E_{t-i} &= \Theta^E_{t-i} + \theta^H_{t-i}\end{aligned} \tag{4}$$

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:

$$\begin{aligned}v^x_{t-i} &= \cos(-\bar{\Psi}^R_t)V^X_{t-i} - \sin(-\bar{\Psi}^R_t)V^Z_{t-i}\\ v^z_{t-i} &= \sin(-\bar{\Psi}^R_t)V^X_{t-i} + \cos(-\bar{\Psi}^R_t)V^Z_{t-i}\end{aligned} \tag{5}$$

$$\begin{aligned}f^x_t &= \cos(-\bar{\Psi}^R_t)F^X_t - \sin(-\bar{\Psi}^R_t)F^Z_t\\ f^z_t &= \sin(-\bar{\Psi}^R_t)F^X_t + \cos(-\bar{\Psi}^R_t)F^Z_t\end{aligned} \tag{6}$$

In the second approach, the respective direction of movement of the previous step was used as a dynamic reference angle (*Translational Motion Reference System*). 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.

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

Labels and features were expressed as:

$$\begin{aligned}\psi^H_{t-i} &= \Psi^H_{t-i} - \Psi^R_{t-i}\\ \theta^H_{t-i} &= \Theta^H_{t-i}\\ \psi^B_{t-i} &= \Psi^B_{t-i} - \Psi^R_{t-i}\\ \psi^E_{t-i} &= \Psi^E_{t-i} + \psi^H_{t-i}\\ \theta^E_{t-i} &= \Theta^E_{t-i} + \theta^H_{t-i}\end{aligned} \tag{8}$$

$$\begin{aligned}v^x_{t-i} &= \cos(-\Psi^R_{t-i})V^X_{t-i} - \sin(-\Psi^R_{t-i})V^Z_{t-i}\\ v^z_{t-i} &= \sin(-\Psi^R_{t-i})V^X_{t-i} + \cos(-\Psi^R_{t-i})V^Z_{t-i}\end{aligned} \tag{9}$$

$$\begin{aligned}f^x_t &= \cos(-\Psi^R_{t+1})F^X_t - \sin(-\Psi^R_{t+1})F^Z_t\\ f^z_t &= \sin(-\Psi^R_{t+1})F^X_t + \cos(-\Psi^R_{t+1})F^Z_t\end{aligned} \tag{10}$$

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).

B. 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) [38] resulting in the final linear dense layer with two outputs - one for each label coordinate. In total, the model with all features had 51,586 trainable parameters and used adam as the optimizer [39]. 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 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.

First, we created a full model that included all seven features presented in the data preparation. The full model 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 effects of eye-tracking and IMU data, models without these features were added.

In order to obtain a more detailed picture, we also assessed variants of the long-term full 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. [40], 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 [41].

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 [42], which is based on the 'Bayes by Backprop' approach introduced by Blundell et al. [43].

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.

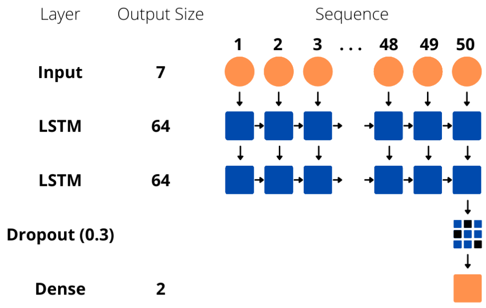


Figure 2: The model Architecture used for both short-term and long-term predictions. 7 features in 50 time steps enter the model. The circles represent this input. The following 4 rows marked with squares form the 4 layers of the model. The final dense layer outputs the prediction result.

C. Evaluation

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 subject 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.

2) Statistical Significance: Using this cross-validation approach, individual prediction errors were calculated for each subject and test set. Moreover, to decide whether a model outperforms a reference model (e.g. the benchmark or a model with fewer features), a significance test provides more information than a mere comparison of average errors.

The results of two cross-validated models are based on the exact same data. Hence, the data is paired. Nadeau and Bengio [44] 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 [44]. It should be mentioned that the results of these significance tests need to be treated with caution. Bouckaert and Frank [45] 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 was applied to the p-values of a single paragraph to avoid underestimation of the p-value due to multiple testing [46]. All tests were two-sided and the assumption of normally distributed data was tested with a Shapiro-Wilk test beforehand [47].

3) Benchmarks: Since this data has 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.

IV. Results

A. 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 mm²) 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 mm²). The former gave a more accurate prediction for every subject. 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.

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 and the model without eye data were almost identical with 5.16 mm and 5.14 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 ($W = 0.75, p = 0.02$). Moreover, testing an effect of eye-tracking data would have been unnecessary since the full model was not better than the model using positional and IMU features. The difference between the model using the two positional features and the model using positional and IMU features failed to reach statistical significance ($t(5) = -1.93, p = 0.11$). All in all, the errors of the LSTM short-term models were quite similar.

Compared to all of the benchmark models, the LSTM models provided better predictions for each of the 6 test sets and each of the 12 subjects. 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 I summarizes all model results.

Table I: 50ms prediction

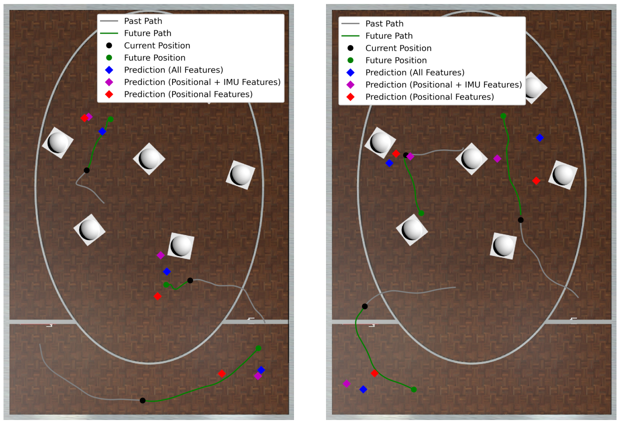
Model			
Architecture	Features	mde	sd
LSTM	positional + IMU	5.14 mm	0.64 mm
LSTM	all	5.16 mm	0.65 mm
LSTM	positional	5.29 mm	0.70 mm
GRU	all	5.33 mm	0.64 mm
Linear Model	all	6.14 mm	0.82 mm
Interpolation	positional	10.45 mm	1.91 mm
Mean	-	16.51 mm	1.53 mm

B. 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 cm²) compared to the *Translational Motion Reference System* with 68.85 centimeters (the absolute error was 43.87 cm; the squared error was 58.49 cm²). The *Mean Head Orientation Reference System* gave a more accurate prediction for each subject. Thus, the *Mean Head Orientation Reference System* was used as the coordinate system for all long-term prediction models and benchmarks.

The 50-sample input sequences and prediction labels formed 156,076 input-output pairs in total. The subjects 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. While the prediction using no eye data came in just behind the full model (mde = 67.56 cm vs. mde = 65.73 cm), the model using only position data falls off at 78.38 cm, which was significantly lower than the full model ($t(5) = -6.99, p = 0.003$) and the model using positional and IMU features ($t(5) = -4.92, p = 0.007$). Although the difference between the model using positional and IMU data and the model also using eye data reached statistical significance ($t(5) = -3.01, p = 0.029$), 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 the 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 exemplary predictions). Further investigation indicated that the gap between the models with and without eye data was not evenly distributed over the length of the predicted path (see Figure 4). At peak, between 50 and 60 cm, the difference reached 9.33% for the prediction of short distances. We also found that beyond a distance of 1.5m, the prediction error decreased in both models.



(a)

(b)

Figure 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.

Table II: 2.5s prediction

Model			
Architecture	Features	mde	sd
LSTM	all	65.73 cm	5.12 cm
GRU	all	66.17 cm	6.01 cm
LSTM	positional + IMU	67.56 cm	5.46 cm
LSTM	positional	78.38 cm	6.77 cm
Linear Model	all	92.52 cm	8.09 cm
Interpolation	positional	131.09 cm	16.16 cm
Mean	-	144.72 cm	14.65 cm

The difference between the best LSTM model and the best benchmark model (linear model) reached statistical significance ($t(5) = -11.08, p < 0.001$).

C. Model Variants

Between the LSTM and GRU architectures, no significant difference could be found for both long-term predictions ($t(5) = -0.73, p = .50$) and short-term predictions ($t(5) = -1.13, p = .31$). Thus, it is quite a comparable model. 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 ($t(5) = -16.68, p < 0.001$). When testing the Bayesian model, 10 predictions were sampled per input. Although the model performed better than the full model (65.19 cm), the improvement failed to reach statistical significance ($t(5) = 0.81, p = 0.45$). Table II summarizes all model results.

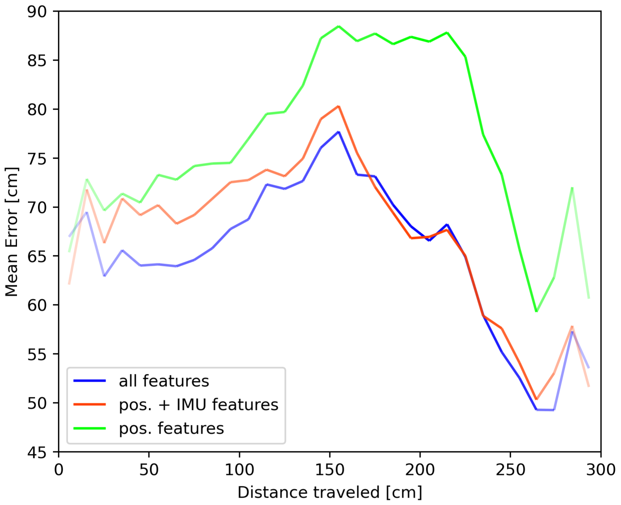


Figure 4: The mdes of models using different sets of features as a function of the distance that the user walked during the 2.5 s used as label data. Line transparency indicates the number of observations that factored into this data point.

V. Discussion

In this study we presented trajectory prediction models trained on free locomotion data obtained in a real-walking VR setup. We compared prediction quality of different models using different timescales, different sets of features, and different coordinate systems. We will first discuss the models and the limitations of the data and then discuss features and coordinate systems.

An LSTM model was able to provide successful prediction of future positions and was able to outperform all of our benchmark models. This was especially noticeable in long-term predictions of position after 2.5s. For short-term predictions of the next 50 ms, the LSTM model outperformed the benchmark models only slightly. However, the results of the full feature GRU model indicate that a more cost-efficient architecture might be sufficient.

The Bayesian model 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. For future applications, it may be possible to reduce the prediction error by applying a moving average to a time series of predictions while walking.

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. 35]. By including locomotion estimation, these methods could also be used for immersive environments that allow real walking. Online long-term prediction could be helpful for early detection of potential collisions and thus in collision avoidance. It could also be useful for optimizing redirected walking algorithms in VR. 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.

Regarding the set of features of the 2.5 second prediction, the results suggest that IMU data is a 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 [48]. Additionally, eye-tracking data provided a small but significant benefit in predicting walking paths. The notion that the addition of eye data can improve predictions is also in accordance with previous findings [28]. Notably, our findings indicate that eye data offers the greatest predictive utility over short walking distances (see Figure 4), or slow movements, respectively. One reason for this result could be that subjects 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. Figure 4 also shows 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 was needed to be estimated. To estimate path bending, we divided 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.

We also compared two types of coordinate systems, one based on mean head orientation, the other based on the current direction of motion. The evaluation 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 information was basically the same 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. Using a model with more interconnections and many layers, capable of such transformations is possible. 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 when estimating long-term positions. One explanation for this difference 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 prediction. 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.

The features we used for our prediction models are features of the users' locomotion and orientation of the body and eyes. These are all egocentric features and do not contain information from 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). However, we can assume that certain movements are represented disproportionately often in our data set, which diminishes the transferability of the model. 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.

VI. Conclusion

We presented a report on deep learning trajectory prediction using position, orientation, and eye-tracking data. It is a cross-validated implementation which uses IMU data and specifically targets VR contexts. We showed how a model using the LSTM architecture can be used to predict walking paths in VR. Moreover, our results suggest that eye-tracking data provides an advantage for this task, especially regarding short distances in long-term predictions.

VII. Declaration of Conflicting Interests

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

VIII. 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.

Acknowledgment

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

References

[1] P. Basili, M. Sağlam, T. Kruse, M. Huber, A. Kirsch, and S. Glasauer, "Strategies of locomotor collision avoidance," *Gait & posture*, vol. 37, no. 3, pp. 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*, vol. 15, no. 2, pp. 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*, vol. 38, no. 1, pp. 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*, 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*, 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*, vol. 19, no. 11, pp. 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, 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, 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*, 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*, 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, 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, 2019, pp. 2728–2735.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 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, 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, 2019, pp. 63–71.

[17] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected Walking," in *Eurographics 2001 - Short Presentations*. Eurographics Association, 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*, vol. 34, no. 2, pp. 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*, vol. 16, no. 1, pp. 17–27, 2009.

[20] 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*, 2016, pp. 2176–2184.

[21] 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*, 2019, pp. 1–9.

[22] 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*, 2020, pp. 1–3.

[23] L. Theis, I. Korshunova, A. Tejani, and F. Huszár, "Faster gaze prediction with dense networks and fisher pruning," *arXiv preprint arXiv:1801.05787*, 2018.

[24] 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*, 2018, pp. 5333–5342.

[25] 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)*, 2018, pp. 754–769.

[26] 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, 2019, pp. 273–282.

[27] 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*, vol. 27, no. 10, pp. 5142–5154, 2018.

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

[29] 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*, vol. 143, no. 2, pp. 221–230, 2002.

[30] H. Brument, I. Podkosova, H. Kaufmann, A. H. Olivier, and F. Argelaguet, "Virtual vs. physical navigation in vr: Study of gaze and body segments temporal reorientation behaviour," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2019, pp. 680–689.

[31] M. Land and B. Tatler, *Locomotion on foot*. Oxford University Press, 07 2009, pp. 100–115.

[32] S. Tuhkanen, J. Pekkanen, P. Rinkkala, C. Mole, R. M. Wilkie, and O. Lappi, "Humans use predictive gaze strategies to target waypoints for steering," *Scientific reports*, vol. 9, no. 1, pp. 1–18, 2019.

[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*, vol. 28, no. 8, pp. 1224–1233, 2018.

[34] 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*, vol. 33, 2011.

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

[36] 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*, vol. 18, no. 4, pp. 555–564, 2012.

[37] R. E. Kalman, "A new approach to linear filtering and

prediction problems," *Journal of Basic Engineering*, 1960.

[38] 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*, vol. 15, no. 1, pp. 1929–1958, 2014.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] 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.

[41] 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, 2019, pp. 824–830.

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

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

[44] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine learning*, vol. 52, no. 3, pp. 239–281, 2003.

[45] 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, 2004, pp. 3–12.

[46] 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)*, vol. 57, no. 1, pp. 289–300, 1995.

[47] Shapiro and Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, 12 1965. [Online]. Available: https://doi.org/10.1093/biomet/52.3-4.591

[48] 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*, vol. 18, no. 1, pp. 177–190, 2003.