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

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 positional, orientation and eye-tracking data with a mean error of 5.14 mm. The best long-term model was the LSTM using positional and orientation data with a mean error of 65.73 cm. Gaze data offered the greatest predictive utility for positional, orientation and eye-tracking data with a mean error of 5.14 mm. 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.

![Figure 1](image1.png)

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 \((\Phi)\), pitch \((\Theta)\) and yaw \((\Psi)\). Both the orientation of the HMD \((\Psi_t^H, \Theta_t^H, \Phi_t^H)\) and the orientation of the body tracker \((\Psi_t^B, \Theta_t^B, \Phi_t^B)\) were recorded.

Lastly, the outputs of the Vive Pro Eye’s integrated eye tracker were obtained as yaw and pitch angles \((\Psi_t^E, \Theta_t^E)\).

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 \((\Psi_t^H, \Theta_t^H)\) and gaze direction \((\Psi_t^E, \Theta_t^E)\) as well as the yaw angle of the body tracker \((\Psi_t^B)\) 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.

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

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).

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

3) Coordinate Systems: Even though \(\vec{F}_t\) and \(\vec{V}_{t-i}\) 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. \(\psi, \theta\)).

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

\[
\bar{\Psi}_t^R = \frac{1}{l} \sum_{i=1}^{l} \Psi_{t-i}^H \quad \bar{\Theta}_t^R = \frac{1}{l} \sum_{i=1}^{l} \Theta_{t-i}^H
\]

(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:

\[
\begin{align*}
\psi_{t-i}^H &= \Psi_{t-i}^H - \bar{\Psi}_{t-i}^H \\
\theta_{t-i}^H &= \Theta_{t-i}^H - \bar{\Theta}_{t-i}^H \\
\psi_{t-i}^B &= \Psi_{t-i}^B - \bar{\Psi}_{t-i}^B \\
\psi_{t-i}^E &= \Psi_{t-i}^E + \psi_{t-i}^H \\
\theta_{t-i}^E &= \theta_{t-i}^E + \theta_{t-i}^H
\end{align*}
\]

(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:

\[
\begin{align*}
v_{t-i}^x &= \cos(-\bar{\Psi}_{t-i}^R)V_{t-i}^X - \sin(-\bar{\Psi}_{t-i}^R)V_{t-i}^Z \\
v_{t-i}^z &= \sin(-\bar{\Psi}_{t-i}^R)V_{t-i}^X + \cos(-\bar{\Psi}_{t-i}^R)V_{t-i}^Z \\
f_{t-i}^x &= \cos(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^X - \sin(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^Z \\
f_{t-i}^z &= \sin(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^X + \cos(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^Z
\end{align*}
\]

(5)

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.

\[\Psi_{t-i}^R = \zeta(V_{t-i-1}, \begin{pmatrix} 0 \\ 1 \end{pmatrix})\]

(7)

Labels and features were expressed as:

\[
\begin{align*}
\psi_{t-i}^H &= \Psi_{t-i}^H - \bar{\Psi}_{t-i}^H \\
\theta_{t-i}^H &= \Theta_{t-i}^H \\
\psi_{t-i}^B &= \Psi_{t-i}^B - \bar{\Psi}_{t-i}^B \\
\psi_{t-i}^E &= \Psi_{t-i}^E + \psi_{t-i}^H \\
\theta_{t-i}^E &= \theta_{t-i}^E + \theta_{t-i}^H
\end{align*}
\]

(8)

\[
\begin{align*}
v_{t-i}^x &= \cos(-\bar{\Psi}_{t-i}^R)V_{t-i}^X - \sin(-\bar{\Psi}_{t-i}^R)V_{t-i}^Z \\
v_{t-i}^z &= \sin(-\bar{\Psi}_{t-i}^R)V_{t-i}^X + \cos(-\bar{\Psi}_{t-i}^R)V_{t-i}^Z \\
f_{t-i}^x &= \cos(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^X - \sin(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^Z \\
f_{t-i}^z &= \sin(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^X + \cos(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^Z
\end{align*}
\]

(9)

\[
\begin{align*}
f_{t-i}^x &= \cos(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^X - \sin(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^Z \\
f_{t-i}^z &= \sin(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^X + \cos(-\bar{\Psi}_{t-i+1}^R)F_{t-i}^Z
\end{align*}
\]

(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).
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.

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 benchmarks. 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 benchmarks. 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 benchmarks. 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 benchmarks. 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.
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

<table>
<thead>
<tr>
<th>Architecture Features</th>
<th>mde</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM positional + IMU</td>
<td>5.14 mm</td>
<td>0.64 mm</td>
</tr>
<tr>
<td>LSTM all</td>
<td>5.16 mm</td>
<td>0.65 mm</td>
</tr>
<tr>
<td>LSTM positional</td>
<td>5.29 mm</td>
<td>0.70 mm</td>
</tr>
<tr>
<td>GRU all</td>
<td>5.33 mm</td>
<td>0.64 mm</td>
</tr>
<tr>
<td>Linear Model all</td>
<td>6.14 mm</td>
<td>0.82 mm</td>
</tr>
<tr>
<td>Interpolation positional</td>
<td>10.45 mm</td>
<td>1.91 mm</td>
</tr>
<tr>
<td>Mean</td>
<td>16.51 mm</td>
<td>1.53 mm</td>
</tr>
</tbody>
</table>

#### 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 ($t(5) = -9.99, p = 0.003$) 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 32.21 cm, including the lowest 10% below 18.71 cm 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 1.5 m, the prediction error decreased in both models.

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 77.65 cm, 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.

### Table II: 2.5s prediction

<table>
<thead>
<tr>
<th>Model</th>
<th>mde</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM all</td>
<td>65.73 cm</td>
<td>5.12 cm</td>
</tr>
<tr>
<td>GRU all</td>
<td>66.17 cm</td>
<td>6.01 cm</td>
</tr>
<tr>
<td>LSTM positional + IMU</td>
<td>67.56 cm</td>
<td>5.46 cm</td>
</tr>
<tr>
<td>LSTM positional</td>
<td>78.38 cm</td>
<td>6.77 cm</td>
</tr>
<tr>
<td>Linear Model all</td>
<td>92.52 cm</td>
<td>8.09 cm</td>
</tr>
<tr>
<td>Interpolation positional</td>
<td>131.09 cm</td>
<td>16.16 cm</td>
</tr>
<tr>
<td>Mean</td>
<td>144.72 cm</td>
<td>14.65 cm</td>
</tr>
</tbody>
</table>
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


