International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2024)
S. Hasegawa, N. Sakata and V. Sundstedt (Editors)

Towards Environment- and Task-Independent Locomotion Prediction for Haptic VR

Shokoofeh Varzandeh [†1] Khrystyna Vasylevska [†‡ 2] Emanuel Vonach[2] and Hannes Kaufmann[2]

[1] Amirkabir University of Technology, Iran [2]TU Wien, Austria

Abstract

The use of robots presenting physical props has significantly enhanced the haptic experience in virtual reality. Autonomous mobile robots made haptic interaction in large walkable virtual environments feasible but brought new challenges. For effective operation, a mobile robot must not only track the user but also predict her future position for the next several seconds to be able to plan and navigate in the common space safely and timely. This paper presents a novel environment- and task-independent concept for locomotion-based prediction of the user position within a chosen range. Our approach supports the dynamic placement of haptic content with minimum restrictions. We validate it based on a real use case by making predictions within a range of 2 m to 4 m or 2 s to 5 s. We also discuss the adaptation to arbitrary space sizes and configurations with minimal real data collection. Finally, we suggest optimal utilization strategies and discuss the limitations of our approach.

CCS Concepts

• ***Human-centered computing*** → *Virtual reality; Interaction techniques;*

1. Introduction

Recent technological developments have led to an exciting combination of previously independent technologies. One of them brings together synthetic visual experiences in virtual reality (VR) and the abilities of robots to create encountered-type haptic devices (ETHD) that make VR content tangible [YHK96, MHSM*21]. This integration enhances the realism of the simulated worlds by providing physical props for interaction, supporting the illusion with haptic stimuli. Recently, this concept was extended to mobile robots, making it applicable to large walkable VR environments [SHZ*20, MVVK23]. However, collocating a mobile robot with a VR user blindfolded by the headset raises safety concerns and requires high system reliability. At the same time, the simulation realism should not suffer, and the haptic objects should already be in place when the user reaches them. Consequently, the core challenges in human-robot interaction in VR revolve around safety and response time [MML21]. The robot should be aware of the user and maintain a safe distance from her, especially during locomotion. That, in turn, might delay the serving of a haptic prop, which increases the robot's response time. That poses a challenge as planning and navigating takes time, especially if the next object for interaction is not known in advance. Many existing ETHDs tackle this by predefining and optimizing the positions of the haptic objects [VGK17] or by utilizing a specific task for the user to create a time gap between the haptic interactions for the robot to move. Support of unrestricted haptic interaction with a number of arbitrarily or dynamically placed objects is still problematic due to these requirements. One way to address this is to anticipate which haptic object or group of objects the user will interact with next. This way, the robot can already navigate to the predicted position before the user arrives. In VR scenes with predetermined interaction locations, such as a museum, an algorithm can be trained to predict in real-time [DMAH24] to provide more time for the robot to respond. Yet, anticipating interactions becomes challenging in a large VR environment with dynamic content placement, like a large architectural studio during an unpredictable creative process.

This work presents a more universal space-independent real-time prediction concept that supports unrestricted content placement. We discuss the specifics of our probabilistic model and evaluate which features work best for it. We also suggest a better data alignment method for our prediction approach. Furthermore, we demonstrate that the retraining can be performed with minimal losses of accuracy using synthetic data, minimizing the need for real user data collection after each change. In addition, we discuss how our concept can be adjusted for the specifics of a given space, users, and other requirements to achieve high variability and scalability.

2. Related Work

Human locomotion prediction involves forecasting a person's future positions, trajectories, or actions based on current and past movement patterns. Many researchers explored the use of Gaussian

† These authors contributed equally.
‡ khrystyna.vasylevska@tuwien.ac.at

DOI: 10.2312/egve.20241356

Processes (GPs) or Gaussian Mixture Models (GMMs) to recognize or predict such movement patterns. While GPs predict future points based on the relation to existing data, GMMs can be used to find how data is clustered. Tay and Laugier [TL08] developed a framework using GMMs and GPs to predict the movement of dynamic objects in familiar scenes. Kim et al. [KLE11] focused on creating continuous dense flow fields from sparsely collected vector sequences. Yoo et al. [YYY*16] aimed to identify prevalent patterns within a scene and their concurrent occurrence propensities using a mixture of topics and GMMs. They clustered observed movement tracks into distinct groups, representing typical patterns that co-occur with a significant likelihood, and predictions were based on the most dominant pattern group. Makansi et al. [MIÇB19] presented a mixture density network architecture, which generates a spectrum of possible future positions at fixed intervals and then fits a mixture of Gaussian or Laplace distributions to these predictions. Carvalho et al. [CVPK19] leveraged large databases of observed trajectories and combined the concepts of localized movement patterns and clustering by representing each cluster with a linear vector field over a space map. All these methodologies focus on generalizing statistical data within a specific environment, but the final results are space-bound and not universal.

In contrast, location-agnostic approaches match observed partial trajectories to a library of prototype paths, which offers the flexibility to be employed in any free space. Hermes et al. [HWSK09] predicted vehicular paths by comparing the observed trajectory to a collection of patterns using a rotation-invariant distance metric. Keller et al. [KHG11] introduced a probabilistic hierarchical trajectory matching approach that employs a probabilistic tree of sampled human movement snippets to locate a matching sub-sequence. Trautman and Krause [TK10] demonstrated the use of GPs for predicting individual trajectories, with an interaction potential that adjusts the trajectory set based on the proximity of people at each moment in time. Later, they integrated goal information into the model [TMMK13], adding the desired destination as a training point within the GP. Xiao et al. [XWF15] categorized sample paths into pre-set motion classes and standardized them by aligning their starting points and extending along a common axis. Although these approaches offer more flexibility regarding the environment, they require a large collection of general movement patterns or need to be tailored for a specific task.

Dynamic Time Warping (DTW) is widely used to analyze the similarity between two movement paths. It can be employed to build robust path prediction models by finding an optimal alignment to historical path variations. Unhelkar et al. [UPSS15] used DTW to build a prediction model for human motion trajectories to navigate mobile robots safely in the same environment. Pérez-D'Arpino and Shah [PS15] anticipated human hand-reaching motions employing DTW for safe cooperation with a robotic arm. In order to reduce the computational complexity of DTW, Choi et al. [CCLJ20] presented a constrained DTW technique only considering alignments in a limited window. However, DTW has significant computational costs, resulting in a trade-off between flexibility and real-time requirements. In contrast, our proposed method simplifies the alignment process, which makes it more robust against noise and deviations at less computational costs, and is suitable for real-time applications.

Alternatively, researchers employ unsupervised learning methods or Convolutional Neural Networks (CNNs) to derive patterns directly from data for prediction. Käfer et al. [KHW*10] introduced a method based on a coupled Hidden Markov Model for concurrent vehicle trajectory estimation at crossroads. Luber et al. [LSSA12] explored the joint interactions between pairs of pedestrians, employing social dynamics to learn motion prototypes based on observed relative motion in public spaces. Their methodology employed an unsupervised clustering technique to predict the most likely paths for two individuals approaching a point of interaction. Su et al. [SZDZ17] put forward an approach harnessing a social-aware Long Short-Term Memory (LSTM) network as a crowd descriptor, which was then integrated with a deep GP to forecast a comprehensive distribution over future pathways for all individuals in a crowd. Nikhil and Tran Morris [NM18] proposed an approach using CNNs to map an input trajectory of a specified length to an entire future path. [MLSL19] Mao et al. treat the human pose as a graph to train a CNN for up to 1 s motion prediction in trajectory space. Chai et al. [CSBA20] adopted a different strategy by using a fixed set of "anchor" trajectories, which are state sequences clustered from training data and represent possible future behavior modes. These anchors serve as inputs to a CNN that infers mid-level scene features and predicts a discrete distribution over the anchors. The model also calculates offsets from the anchor waypoints and uncertainties to produce a Gaussian mixture at each time step. [WMS21] Wang et al. train a neural network with pose data to predict the position of a walking human 0.5 s in advance. [GDS*23] Guo et al. reduce the parameter set for a neural network to only 0.14 million for 1 s human pose prediction. These prediction methods reflect a shift away from strictly sequential models towards frameworks that accommodate the complex and dynamic nature of motion in real-world environments. However, they can lack interpretability of their learned models and adaptation to varying environments, goals, or users might require retraining with massive amounts of training data.

Unlike previous solutions, we strive to create a scalable and robust prediction approach. It supports large virtual and real spaces of different shapes and arbitrary placement of interactive objects. One of the use cases is a mobile robot facilitating a creative process providing the haptic interaction for a freely walking user, like in Mortezapoor et al. [MVVK23]. In such a scenario, haptically interactive objects might vary in number and be relocated at any time. Supporting such an unrestricted yet realistic scenario is still challenging. Related works presented above often employed computationally heavy models, relying on video training, separate analysis of video parts (e.g., trees, cars, pedestrians), or object detection in streaming data. In contrast, our proposed concept is user-oriented and adaptable to the specifics of the task, users, sizes and shapes of haptic objects and spaces. It is based on short trajectories and is more universal and lightweight, requiring fewer features. This makes it suitable for real-time use in encountered-type haptic VR.

3. Prediction Concept

We propose to detach the prediction from the environment and make predictions within a dedicated area around the user. This prediction area should be sized to meet the requirements for the time or

distances at which the predictions should be made. Since we focus on the prediction without knowledge of the environment, we need to account for rapid changes in the user's heading within the VE in all possible directions. This suggests a circular prediction area around the user for the general case, as shown in Figure 1. Should the user's activity be limited by the nature of the task, the circular area can be reduced to a sector-based area. The prediction area should move with the user but take into account the nearby haptic objects. For prediction, we split the area into sectors roughly sized to the haptic objects as shown in Figure 1 b. The center of the arch of the sector is then marked by the prediction target. The number of sectors determines the spatial precision of the prediction. We conceptualize that each person can decide on multiple movement directions, prioritizing based on surroundings and preferences. Therefore, we estimate the likelihood of all prediction targets simultaneously to anticipate the possible changes in behavior as soon as possible. Since the prediction area should react to all the haptic objects in the user's proximity, the resulting trajectories might not always start from the center of the prediction area. Therefore, we introduce a tolerance zone in the prediction area's center (see Figure 1 a). The tolerance zone allows better fitting of the boundary of the prediction area to the haptic objects and facilitates organic locomotion within the prediction area, including the changes in direction. If there are no haptic objects, the prediction area moves with the user and aligns with the haptic objects when they are in close proximity. Once the user's prediction boundary gets near a haptic object, the area's position is gradually adjusted to match one of the targets with the object, and a prediction for possible interaction with the object can be easily made. If multiple objects are nearby, the priority of the alignment is decided by the distance to the object and the user's current heading. Using the GMM probabilistic learning on trajectories within the prediction area, we can estimate the probability of the object for interaction that is within the range. It is also important to decrease the influence of possible signal noise and increase the prediction's overall accuracy. Therefore, our prediction algorithm considers both the current and the recent user's motions.

4. Training

A set of targets $\mathcal{T} = \{T_1, T_2, \ldots, T_{16}\}$ is located on the boundary of the prediction area. The entire set of trajectories to a target T_j is represented $\mathcal{X}_j = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$, with M being the total number of trajectories. Each trajectory $\mathbf{x}_i$ from the set $\mathcal{X}_j$ is described by a sequence of features per time step $k \in \{1, 2, \ldots, K_i\}$: $\mathbf{x}_i = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K\}$, where K is the total number of time steps (frames) in a trajectory, and $\mathbf{f}_k$ is a feature vector of a time step k. The trajectory data comprises the ID of the trajectory and a collection of feature vectors $\mathbf{f}_k$ consisting of the following data: 2D position vector $\mathbf{p}_k \in \mathbb{R}^2$, head yaw rotation $\psi_{\text{head},k} \in [0, 2\pi)$, body yaw rotation $\psi_{\text{body},k} \in [0, 2\pi)$, and 2D velocity vector $\mathbf{v}_k \in \mathbb{R}^2$.

Depending on the dataset, each trajectory might differ in the number of time steps K_i due to framerate variation, differences in user behavior, and average path length. Therefore, we need to align all trajectories before training. In our case, we calculated the average number of time steps K_{mean} and fixed it at a mean number of 372 based on all the trajectories for all the targets in our collected

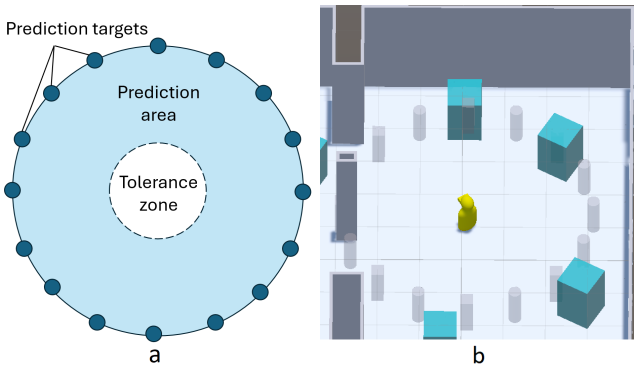

Figure 1: *Prediction area: (a) its structure and (b) visualization within a virtual reconstruction of a real workspace with symbolic haptic objects (cyan) and a user (yellow).*

data. Should the individual trajectory's K_i be shorter or longer, we employ linear interpolation to proportionally resample the trajectory to fit the K_{mean}. Then, we calculate the 2D velocity vector using a first-order derivative estimate with a finite difference equation based on the Taylor extension with the fourth-order five-point backward stencil [Tay16].

In the training phase, we calculate the mean feature vector $\mu_j[k]$ as follows:

$$\mu_j[k] = \frac{1}{M_j} \sum_{i=1}^{M_j} \mathbf{f}_i[k], \tag{1}$$

where M_j is the number of trajectories belonging to target T_j, and $\mathbf{f}_i[k]$ is the feature vector at time step k for trajectory x_i leading to target T_j. Similarly, the covariance matrix of the feature vectors at time step k per target T_j is calculated as:

$$\Sigma_j[k] = \frac{1}{M_j - 1} \sum_{i=1}^{M_j} (\mathbf{f}_i[k] - \mu_j[k])(\mathbf{f}_i[k] - \mu_j[k])^\top, \tag{2}$$

where $\Sigma_j[k]$ is the covariance matrix, $\mu_j[k]$ is the mean of the feature vectors, M_j is the number of trajectories belonging to target T_j, and, $\mathbf{f}_i[k]$ is the feature vector of trajectory x_i. Note that we consider the difference relative to the corresponding target's orientation for the rotation feature in mean and covariance calculation. Therefore, every rotation difference has values between -180 and 180 degrees. This way, we fit a Gaussian distribution for each target at every time step, utilizing the data of the recorded trajectories according to Equation 1 and Equation 2. The model training results in 16 trained GMMs, one model per target.

5. Alignment for Prediction

Our model continuously analyzes the user's tracked motion in real-time to infer her intention. The aim is to determine which target they are most likely going for. For this, we need to find a method to align the current trajectory with stored distributions. The most straightforward approach would be to use the Euclidean distance, where we need to find a time step with minimal distance for each GMM. However, since the prediction area is circular, we took this

into consideration and investigated a second approach to alignment. Thus, we implemented and compared both methods in section 11.

Euclidean Alignment Method. Here, we align streaming position data with a reference time step sequence of μ_j by finding the minimum Euclidean distance in the reference sequence for each point in the streaming data. For each point in the streaming 2D position data $p_{user} = (p_1, p_2)$, we calculate the Euclidean distance to every point in the position feature $p_i(\mu_j[k])$ from the mean reference sequence of each target μ_j. For $k = 1$ *to* K_{mean},

$$\begin{aligned} d_k &= \sqrt{(p_1 - p_1(\mu_j[k]))^2 + (p_2 - p_2(\mu_j[k]))^2}, \\ eucl\_index_j &= \arg\min_k(d_k) \end{aligned} \tag{3}$$

The point k in the reference sequence with the minimum Euclidean distance d_k is used as the alignment index $eucl\_index_j$ for target T_j.

Circular Alignment with Radius-Based Search. We propose a method that relies on the circular nature of the prediction area to align streaming data with a reference sequence. Each new data point in the streaming data is used to calculate the distance between the center of the circle and the user's 2D position $p_{user} = (p_1, p_2)$ resulting in a radius r. For each target, we take the point in the position feature $p_i(\mu_j[k])$ of the mean reference sequence μ_j with a minimum distance d_k to the calculated radius. For $k = 1$ *to* K_{mean},

$$\begin{aligned} d_k &= |r - \sqrt{p_1^2(\mu_j[k]) + p_2^2(\mu_j[k])}|, \\ circ\_index_j &= \arg\min_k(d_k) \end{aligned} \tag{4}$$

The time step k of minimum d_k is then used as the circular alignment index $circ\_index_j$ for target T_j.

6. Probability Inference

After the alignment, we employ GMMs to identify the target with the highest probability of being the next destination. For that, we calculate the log posterior to identify the target that best matches the user's observed trajectory x_o. We utilize a Bayesian approach [DW12] to determine the most probable target T_j based on the observed trajectory $\mathbf{x}_o[1:K_o]$, as shown in Equation 5.

$$P(T_j \mid \mathbf{x_0}[1:K_o]) \propto P(T_j) \cdot P(\mathbf{x}_o[1:K_o] \mid T_j), \tag{5}$$

where $P(T_j)$ is the prior probability of the target T_j. We use a uniform prior for all targets. $P(\mathbf{x}_o[1:K_o] \mid T_j)$ is the likelihood of observing the trajectory $\mathbf{x}_o[1:K_o]$ given the target T_j. The likelihood term can be calculated:

$$P(\mathbf{x}_o[1:K_o] \mid T_j) = \left(\prod_{k=1}^{K_o} \mathcal{N}(\mu_j[k], \Sigma_j[k]) \right)^{1/K_o}. \tag{6}$$

Then we can compute the product from Equation 6 as a logarithm for each target T_j at the time step $k = K_o$ as expressed here:

$$\frac{1}{K_o} \sum_{k=1}^{K_o} \left[-\log(2\pi)^{\frac{N_f}{2}} - \frac{1}{2} \log|\Sigma_j[k]| - \frac{1}{2} \delta[k]^T \Sigma_j^{-1}[k] \delta[k] \right], \tag{7}$$

where $\mathbf{N}_f$ is the number of features used, $\delta[k]$ represents the difference between the observation $\mathbf{x}_o[k]$ and the mean $\mu_j[k]$ at a particular time step k, as determined by the alignment process. This estimation is executed during runtime for each frame of the retrospective data points from previous frames. To predict short-term future behavior, we also consider the recent data points. We implement this approach by applying weighted scaling to the probabilities, dividing by 2^((n - 1 - j) / 10 + 1), where n is the total number of data points, and j is the point's index number in the sequence from oldest to newest. Then, the resulting probabilities are summed up for each target. The target with the highest likelihood is identified as the best target corresponding to the observed trajectory.

7. Integration and Validation

For integration and validation of our proposed approach, we utilized a real use case for large-area haptic interaction with large objects in VR served by a robot. Therefore, the parameters of the prediction area were decided based on the real environment and the targeted prediction time and precision. Consequently, real training and evaluation data were collected for this configuration. We discuss alternative integration scenarios and reuse of the trained algorithm in section 12.

Our test space sized 12 m by 13 m contains obstacles that divide the space into two equal, interconnected rooms with a width of 6 m, as can be seen from the reconstruction Figure 1 b. The robot's response time range is 2-5 s. Therefore, we chose the prediction area with the maximum possible radius $r_{predict} = 3$ m that fits within the room. This allows us to achieve an acceptable prediction accuracy within 2-3 s needed for the robot to arrive. Our sample haptic elements have a 1 m^2 footprint and are spread throughout the workspace. Therefore, we split our prediction area into 16 sectors with 1.2 m spacing between the prediction targets, each covering an angle of 22.5°. That is sufficient since the haptic objects are comparable in size to the user and distributed throughout the space to allow the user free navigation between them. Similarly, we defined the tolerance zone to have a 1 m radius. The movement of the prediction area along the direction of the user's heading or towards the objects in proximity is limited to 0.07 m per frame to minimize the impact on the user's relative trajectory. The fitting happens when the distance between the person and the object is within $r_{predict} - 0.8$ m and $r_{predict} + 0.5$ m, and the distance between the target's center and the haptic object is within 1-2 $r_{predict}$. This condition ensures fitting to multiple objects. To handle multiple nearby objects and expand the prediction window, the circle moves to fit the objects roughly within the user's heading direction. Finally, the prediction algorithm retrospect is set to the last 50 frames (approx. 0.7 s at 75 fps). We implemented the training and inference as regular Unity C# scripts. For the evaluation, we ensured precisely timed recording and accurate replay to reflect the real framerate.

8. Technical Setup and VR Environment

We used a Windows 10 PC with an Intel i9-9900K CPU, NVIDIA RTX 2080Ti GPU, and 32 GB RAM for the evaluation and VR rendering. The user was provided visual input via the HTC Vive Pro head-mounted display (HMD) with a standard wireless module and a power bank. The tracking employed 4 HTC Vive v.2 base stations covering the 6.5 m by 6.5 m tracking area. The head was tracked for position and orientation with the HMD. The user also wore one additional HTC Vive v.2 tracker on the tailbone to pro-

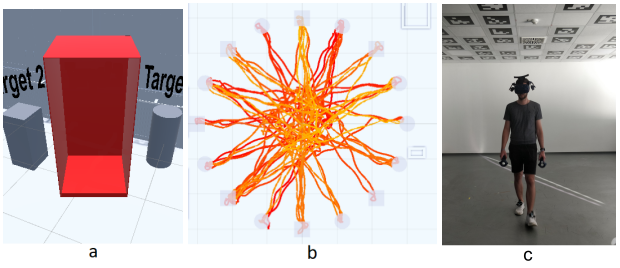

Figure 2: *Data collection: (a) virtual cabin positioned next to the prediction area targets, (b) experimenter's top view on the user's path, (c) a user during the data collection.*

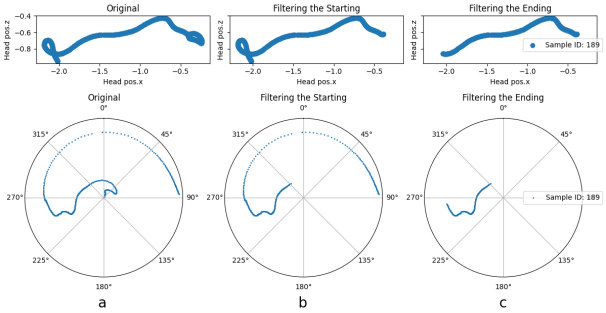

Figure 3: *The data preprocessing: (a) raw data, (b) filtering a hook at the beginning of the trajectory, (c) filtering a hook at the end of the trajectory to obtain the final trajectory data.*

vide body orientation and two more on each foot for collecting additional data (step length and width). We used Unity 3D 2022.1.24f with OpenXR support for VR rendering and motion tracking. The virtual reconstruction of the real workspace with the user inside the prediction area is shown in Figure 1.

9. Real Data Collection

To train our prediction model, we invited 24 volunteers (12 female, 12 male) to collect locomotion data. Participants ranged in age from 19 to 42 years ($Mean = 28.83$, $SD = 5.48$). We used the setup described in section 8 for the data collection. The recording was done for a stationary prediction area within the correctly registered workspace reconstruction. Note that for the training, we aim to collect a range of trajectories from 2 m to 4 m, because trajectories < 2 m are too close to direct hand interaction range to position a robot in time. During actual prediction, the prediction area moves with the user and responds to her actions. Therefore, the collected data also applies to cases with trajectories between different objects due to prediction area realignment. Inspired by Unhelkar et al. [UPSS15], we chose to record the positions and orientations of the HMD and trackers with timestamps at approximately 75 Hz.

Procedure and Task. Each participant was informed of the purpose of the data collection, what data would be recorded, and the possible outcomes of the VR exposure. That was followed by signing the informed consent and filling out the general questionnaire and Kennedy SSQ questionnaire [KLBL93]. Participants were also informed that they could pause or discontinue their participation at any moment. Next, the participants were given the task to find and enter a red cabin (as shown in Figure 2 a), then stay there for 3 s. This triggered the cabin's relocation to a new position that was alternating between a random prediction target position and a random pose within the tolerance zone. The participants were instructed to continue chasing after the cabin in the same manner. To keep the participants motivated, we gamified the task by granting the participants a random piece of a puzzle picture for each visit to the red cabin. Typical resulting trajectories are shown in Figure 2 b.

Data Preprocessing. We recorded a total of 1166 trajectories from all 24 participants for all the targets. Due to the occasional issues with wireless connection, the data had to be preprocessed. Some recorded trajectories were incomplete and thus had to be discarded, and some had interruptions. We identified the parts of the same trajectory by ID. If the distance between the parts was less than 0.1 m, we combined them into a continuous trajectory. Also, the need to search for the next cabin resulted in participants turning at the beginning and the end of trajectories, creating hook-like trajectory ends as in Figure 3 a. As these hooks are not part of the trajectory but rather a task artifact, we filtered these parts of trajectory data. We excluded the parts of the trajectories beginning with the head rotation exceeding 60° from the target direction (see Figure 3 b) and endings with the rotation that exceeded 15° angle away from the target. The filtering did not change the general flow and shape of the trajectories. Figure 3 c shows the final result. After that, the positions were converted to a 2D XZ plane relative to the center of the prediction area, and we calculated velocity. The yaw rotations of the HMD and tailbone tracker are in degrees to the forward vector of the not-rotating prediction area.

10. Synthetic Data Generation

Previously, there was a need to collect new user data for each significant change in the environment or task. However, since our prediction area is environment-independent and is oriented only to targets in proximity, we saw the possibility of minimizing the effort. Based on the filtered real training data, we simulate the user's path and tested whether our synthetic data could be used to train the GMMs to predict for a real user with sufficient reliability. Although, real human motions have multiple complex details and limitations, due to our feature selection and the averaging of the feature data, a simplified modeling is appropriate (as discussed in subsection 11.3). We modeled movement data by introducing stochastic variations to a straightforward path and incorporating intentional semi-randomness into each trajectory. The initial sequence of waypoints is a random selection of the starting position within the tolerance zone and the final position associated with the prediction target (T_j). This path is then broken into segments roughly equal to double the average step length of 45 *cm* in VR, based on our observations and prior findings [LJKM*17]. A new waypoint is calculated for each 90 *cm* segment of the path, deviating from the original direction to one side, mimicking the average distance between the feet of 30 *cm*. This deviation simulates the user's weight shift during walking. The shift's side is randomized for each trajectory. New points are inserted along the path to form a target-oriented tra-

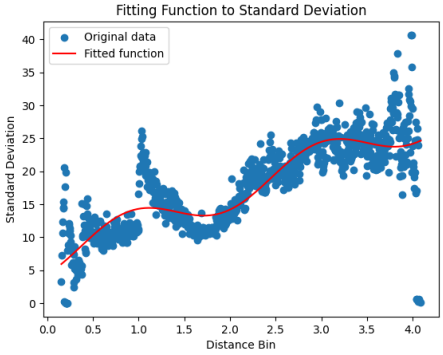

Figure 4: *Standard deviations of head rotation (blue dots) along the path to target approximated with a fitted function (red line).*

jectory roughly resembling human locomotion behavior. The path is then refined regarding the behavior and velocity using the built-in Unity navigation agent. Our settings with double average speed and average acceleration values with active auto-braking compensate for speed reduction due to the high number of waypoints. This results in a gradual and continuous trajectory. For large prediction areas or spaces with many obstacles, non-linear base trajectories might be beneficial, requiring only minor modifications.

Stochastic Modeling of Head Rotation. Using real-world data, we analyzed the head rotation as a function of the difference to the target. The calculated mean and standard deviation for each of 4 *mm* intervals (1000 per maximal trajectory length of 4 *m*) showed the mean value tending to zero. Consequently, we fit a sinusoidal function (Equation 8) with a linear attenuation trend to standard deviation as shown at Figure 4.

$$f(d) = A\sin(\omega d + \phi) + Bd + C, \tag{8}$$

where $A = 2.4299$, $\omega = -3.5986$, $\phi = 11.5808$, $B = 4.9929$, and $C = 7.8498$. This enables stochastic modeling of randomized head rotation reflecting the behavioral uncertainty at the beginning of the trajectory where the goal selection is not finalized. Thus, our simulated user rotates its head relative to the body with a realistic variability for a human-like behavior.

Finally, we take advantage of our circular prediction area to simplify the synthesis of a large trajectories dataset.For that, we deploy Unity's NavAgent to a single target to generate a data pool. Then, we bootstrap it to create a subset for each target and rotate the data accordingly. Thereby, we can adapt the synthetic data to any number of targets for the same prediction area. This approach ensures time-saving and data variation between the targets.

11. Evaluation

Our evaluation investigates the minimum required dataset size for effective training, compares the effects of two different alignment methods on prediction accuracy, and assesses the training with synthetic data. We evaluated our approach with two types of data: real data collected from the real participants and synthetic data that approximates the real behavior. After preprocessing the real data, we obtained 1088 trajectories in total, resulting in 68 trajectories for

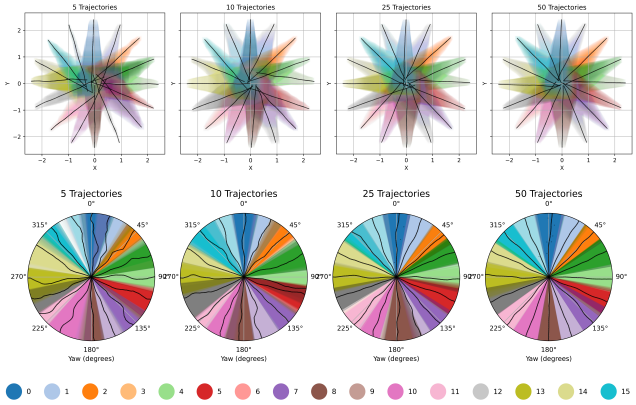

Figure 5: *Influence of the different training dataset sizes (5, 10, 25, 50) per target on the resulting trained means (black lines) and variance (shaded areas) of position (top) and rotation (bottom) vectors.*

each of the 16 targets. From it, we formed a training dataset with 48 trajectories and a testing dataset with 20 trajectories per target. For the synthetic dataset, we generated 2000 trajectories and made a training dataset by bootstrapping 100 trajectories per target.

11.1. Minimum Training Dataset Size

We addressed the question "How big should the training dataset be?" using our synthetic training dataset due to the unlimited data availability and compared the stability of the mean and variance vectors for all time steps. For the comparison, we chose the sample sizes of 5, 10, 25, 50, and 100 trajectories per target and focused on the head position and orientation data. The comparison results for 5 to 50 trajectories are shown in Figure 5. As can be noticed, the shaded variance areas become better separated and homogeneously distributed as the number of training trajectories per target increases. In contrast, small training samples caused these areas to overlap and even create gaps, suggesting that some neighboring targets are more similar than others. Similarly, the means in both rotational and positional data become more stable and distinct with an increased size of the dataset. This leads us to the conclusion that the minimum size of the dataset should be more than 25 unique trajectories per target. Moreover, if the number of targets on the boundary of the prediction area is increased, this number should also be proportionately increased.

11.2. Alignment Comparison

To determine the best alignment method, we compare the Euclidean Alignment with nearest point search and the Circular Alignment with radius-based search on various combinations of features. Since both methods rely on the head position, this feature is present in all combinations. For this part of the evaluation, we train and test exclusively on the real dataset. We evaluate the prediction accuracy (highest probability match to the assigned target) relative to the distance to the target as it is a more stable reference between subjects then time. The results of the alignment methods comparison with

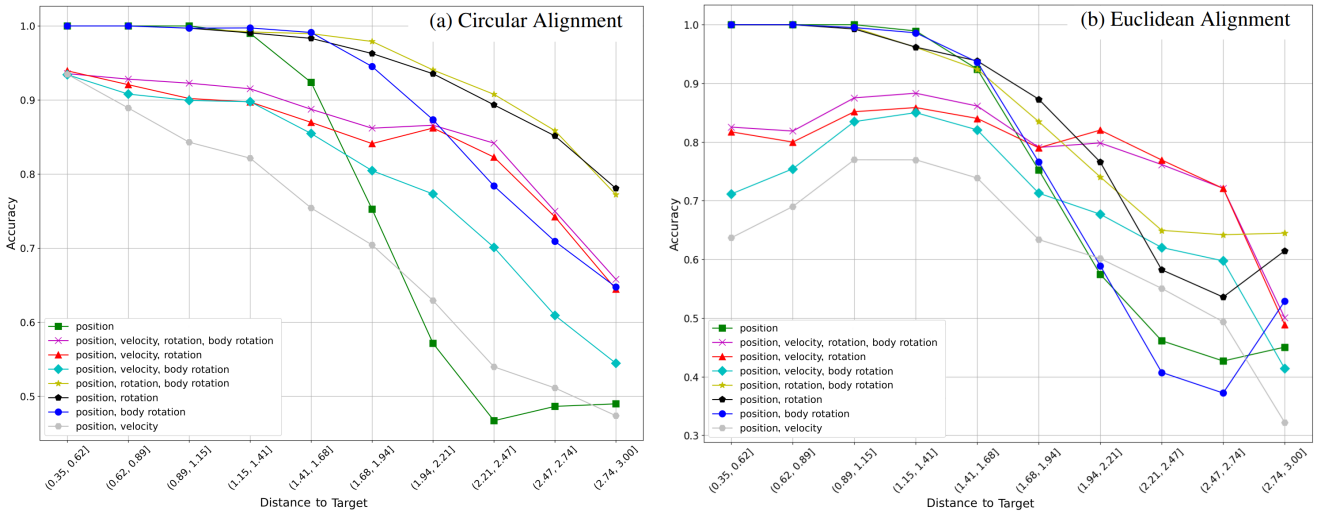

Figure 6: *Prediction accuracy of Circular and Euclidean Alignments with different feature combinations relative to the distance to the target.*

different feature combinations are shown in Figure 6. Overall, we observed that the velocity $\mathbf{v}$ feature does not contribute to the prediction performance. The likely reason is the strong variation in magnitude and acceleration in the data. Therefore, velocity is not suitable for our specific context. Previously, [UPSS15] made a similar observation. Our backward stencil size was also a half smaller smoothing the estimate, but not affecting the outcome.

Circular Alignment. The prediction performs best when using the head position $\mathbf{p}$ and rotation ψ_{head} together with the body rotation ψ_{body} feature combination (see Figure 6 a). It achieves an accuracy of over 75% from the very beginning and steadily increases from there. We see that the $\{\mathbf{p}, \psi_{\text{head}}\}$ and $\{\mathbf{p}, \psi_{\text{head}}, \psi_{\text{body}}\}$ feature sets perform similarly. The slightly better performance for $\{\mathbf{p}, \psi_{\text{head}}, \psi_{\text{body}}\}$ suggests that more stable body rotation helps to reduce the impact of possible natural head rotations on the results. Also, the less accurate results for $\{\mathbf{p}, \psi_{\text{body}}\}$ compared to $\{\mathbf{p}, \psi_{\text{head}}\}$ demonstrate the importance of the head fixation on the target at the beginning of the trajectory.

Euclidean Alignment. While the Euclidean Alignment method also shows some promising results (see Figure 6 b), it underperforms compared to the Circular alignment method, resulting in lower starting accuracy below 65% and overall steeper slopes, reaching the highest prediction precision only 1.4 *m* away from the target. Moreover, there is not a single feature combination that steadily performs well from the beginning of the trajectory to its end. In the beginning, the best performance is achieved by the $\{\mathbf{p}, \mathbf{v}, \psi_{\text{head}}, \psi_{\text{body}}\}$ and $\{\mathbf{p}, \mathbf{v}, \psi_{\text{head}}\}$. However, later, the $\{\mathbf{p}\}$ and $\{\mathbf{p}, \psi_{\text{head}}\}$ perform much better.

11.3. Synthetic Training Viability for Prediction

For this evaluation, we trained the algorithm exclusively on the synthetic data and tested the prediction accuracy with the real testing dataset. Our synthetic dataset for this evaluation contained 48 trajectories per target (a total of 768 trajectories). The testing dataset was the same as in the previous subsection. We employed the Circular Alignment method as it performs best and tested the same feature combinations to see how they compare. The results are presented in Figure 7. In this case, we can see a slight change in the performance of the feature sets. The best prediction results were obtained for the feature vector $\{\mathbf{p}, \psi_{\text{head}}\}$. That might be explained by the stronger coupling in the synthetic data of the body rotation with the position, whereas in the real dataset, the torso rotation has more variance. However, the $\{\mathbf{p}, \psi_{\text{head}}, \psi_{\text{body}}\}$ feature set performs only slightly worse than with the real dataset, achieving an over 70% prediction accuracy within the first meter of the trajectory. The slight decrease in accuracy at the end of the trajectories is likely due to the preemptive turn-around behavior in the testing dataset that was discussed in section 9 *Data Preprocessing*. Since this behavior is a task artifact, it was not modeled in the synthetic dataset.

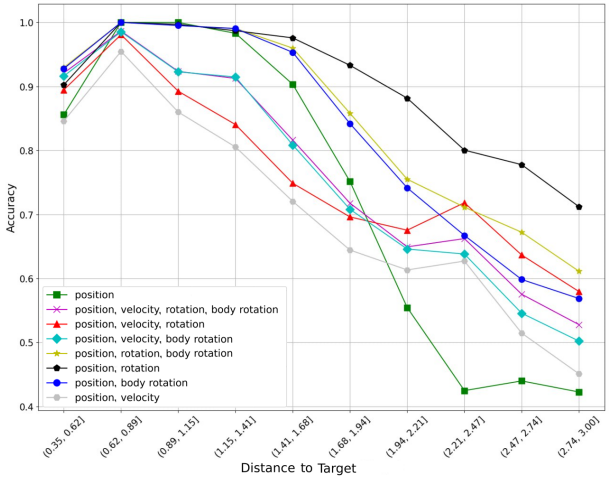

Figure 7: *Prediction accuracy for the synthetic training dataset and real testing dataset with Circular Alignment.*

To better understand the results for the real and the synthetic training datasets, we looked at the two best feature vector candidates and computed the mean accuracy per dataset. For the real training, we achieved 93.4% and 93.8% mean accuracy over the entire distance to a target for $\{\mathbf{p}, \psi_{\text{head}}\}$ and $\{\mathbf{p}, \psi_{\text{head}}, \psi_{\text{body}}\}$, respectively. For the synthetic data, the results were 86.8% and 81.1%, respectively. The results show 7-13% lower performance for synthetic data. These parameters might be improved if the training dataset were a mix of real and synthetic data. Potentially, the synthetic data generation can also be improved if the body rotation is modeled similarly to the head rotation. That, in turn, might also improve the accuracy. Ultimately, the synthetic data can be used to train prediction models like GMMs with a limited feature vector. The synthetic generation might also be beneficial for underrepresented groups of people where the collection of real data is difficult for the participants or the size of the real dataset is too small. This way, the prediction models might become more inclusive, flexible, and reduce the bias for underrepresented groups of users.

12. Discussion

In this paper, we presented a user-oriented prediction approach that is not dependent on the environment and can be used for various tasks with different goals. Unlike the previous solutions, our method does not need large datasets as other GMM approaches like [CVPK19], CNN training [LSSA12], or location agnostic solutions [KHG11] to generalize. Our approach also does not require detailed knowledge of the environment [TL08, UPSS15] or human body pose [WMS21] since we aim to predict the user's intended goal. Although inspired by [UPSS15], we use a lighter set of GMMs that do not require multi-threading instead of the computationally intensive instances of DTW. Consequently, with multi-threading, there is a possibility of running several instances of the predictive algorithm. This can be used to make the prediction more inclusive. For example, one instance can focus on healthy adults, and the other will focus on a user group with different behavioral patterns, such as users in a wheelchair or people with ADHD, for whom the head rotation might not be a good predictor. Additionally, the instances can be focused on different ranges, for example interaction with differently sized objects. In this case, the larger circle will predict the general direction, a large object or a group of smaller objects. If there are small objects at the interaction location, we speculate that the close-range interaction (< 2 m) could be handled with a smaller prediction area. This would be similar to the solution in [PS15] but with lighter GMMs. For instance, we can differentiate between locomotion and hand interaction. Depending on the specifics of the interaction or environment, it is also possible to adapt the prediction area further. For example, to reduce the circular area to a 180° sector or add additional prediction targets. This way, we can avoid the unnecessary computation behind the user or counter the density of the objects of interest.

The short training time for our algorithm suggests that a single instance of the algorithm might be retrained at runtime if several datasets are available. This offers the possibility of individual-based retraining after 15-20 minutes. In this case, our approach for the synthetic data, with the rotation of the trajectory data and bootstrapping, might help create a training-ready dataset. We chose the prediction area's radius based on the prediction time requirements and our environment's size. However, from a practical view, our prediction approach can be reused in other environments thanks to the circular design which is space-independent. It might be adapted or scaled and retrained on the corresponding data to meet other requirements such as prediction time or other ranges.

Naturally, our approach has limitations: The user's position off the center of the prediction area and possible changes in the locomotion direction are countered mainly by the distribution of the trajectories' starts within the tolerance zone and its size. However, as the density of the haptic objects in the proximity increases, there might be cases when the prediction cannot be made in time. Due to the fitting process prioritization for the objects in the heading direction, some objects might end up deep within the prediction area and close to the user, on the sides, or behind her. Should the user change the direction towards one of these objects, there might be up to approximately a second of considerably higher uncertainty of the prediction until the prediction targets and haptic objects realign. There is also a chance she will reach it faster than a reliable estimate can be made. Or if there are neighboring objects, there can be confusion for this short time. Furthermore, the use of synthetic data will always lead to an accuracy loss. However, our average drop of 10% accuracy will decrease with the improvement of the simulation in the future. Finally, there is still a practical interrelation between the size of the prediction area and the size of the space it is deployed in. In particular, that is true when the prediction area is much larger than the space itself. In this case, reusing the trained area is not recommended, and requirements should be reviewed.

13. Conclusion

ETHDs presenting physical props can enhance the realism of haptic VR tremendously. However, it brings new challenges concerning safety and response time, which may require the ability to predict the user's locomotion and interaction targets ahead of time. In this work, we proposed a novel prediction approach for haptic interaction, employing a circular predictive area around the user, which makes our method both more universal and real-time capable. We describe the implementation, training, and performance of our approach, as well as an innovative technique to increase adaptability and scalability by employing synthetic data for training. In our evaluation, we showed that a training set of more than 25 trajectories per target could produce acceptable accuracy in our test scenario. However, a larger and more diverse dataset would perform considerably better. We also presented a Circular Alignment method for trajectories, which proves to be an ideal match for our approach compared to an Euclidean Alignment. We evaluated the ideal feature combination for our algorithm and the viability of using synthetic data for training compared to real user data. With training data based on 48 trajectories per target collected from real users, our algorithm showed a prediction accuracy of almost 80% within the first meter and up to 95% two meters from the target.

Acknowledgements

This work was funded by the Austrian Science Fund, grant F77 (SFB "Advanced Computational Design," SP 5). Special thanks to Alexander Schallhart and Mohammad Ghazanfahri for their help.

References

[CCLJ20] CHOI W., CHO J., LEE S., JUNG Y.: Fast Constrained Dynamic Time Warping for Similarity Measure of Time Series Data. *IEEE Access 8* (2020), 222841–222858. Conference Name: IEEE Access. URL: https://ieeexplore.ieee.org/abstract/document/9290106, doi:10.1109/ACCESS.2020.3043839. 2

[CSBA20] CHAI Y., SAPP B., BANSAL M., ANGUELOV D.: MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. In *Proceedings of the Conference on Robot Learning* (May 2020), PMLR, pp. 86–99. https://proceedings.mlr.press/v100/chai20a.html. 2

[CVPK19] CARVALHO F., VEJDEMO-JOHANSSON M., POKORNY F. T., KRAGIC D.: Long-term Prediction of Motion Trajectories Using Path Homology Clusters. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Nov. 2019), pp. 765–772. https://ieeexplore.ieee.org/document/8968125. doi:10.1109/IROS40897.2019.8968125. 2, 8

[DMAH24] DOHAN M., MU M., AJIT S., HILL G.: Real-walk modelling: deep learning model for user mobility in virtual reality. *Multimedia Systems 30*, 1 (Jan. 2024), 44. URL: https://doi.org/10.1007/s00530-023-01200-z, doi:10.1007/s00530-023-01200-z. 1

[DW12] DONG S., WILLIAMS B.: Learning and Recognition of Hybrid Manipulation Motions in Variable Environments Using Probabilistic Flow Tubes. *International Journal of Social Robotics 4*, 4 (Nov. 2012), 357–368. https://doi.org/10.1007/s12369-012-0155-x. doi:10.1007/s12369-012-0155-x. 4

[GDS*23] GUO W., DU Y., SHEN X., LEPETIT V., ALAMEDA-PINEDA X., MORENO-NOGUER F.: Back to MLP: A Simple Baseline for Human Motion Prediction. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Jan. 2023), pp. 4798–4808. ISSN: 2642-9381. URL: https://ieeexplore.ieee.org/document/10030747, doi:10.1109/WACV56688.2023.00479. 2

[HWSK09] HERMES C., WOHLER C., SCHENK K., KUMMERT F.: Long-term vehicle motion prediction. In *2009 IEEE Intelligent Vehicles Symposium* (June 2009), pp. 652–657. https://ieeexplore.ieee.org/document/5164354. doi:10.1109/IVS.2009.5164354. 2

[KHG11] KELLER C. G., HERMES C., GAVRILA D. M.: Will the Pedestrian Cross? Probabilistic Path Prediction Based on Learned Motion Features. In *Pattern Recognition* (Berlin, Heidelberg, 2011), Mester R., Felsberg M., (Eds.), Springer, pp. 386–395. doi:10.1007/978-3-642-23123-0_39. 2, 8

[KHW*10] KÄFER E., HERMES C., WÖHLER C., RITTER H., KUMMERT F.: Recognition of situation classes at road intersections. In *2010 IEEE International Conference on Robotics and Automation* (May 2010), pp. 3960–3965. https://ieeexplore.ieee.org/document/5509919. doi:10.1109/ROBOT.2010.5509919. 2

[KLBL93] KENNEDY R. S., LANE N. E., BERBAUM K. S., LILIENTHAL M. G.: Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology 3*, 3 (1993), 203–220. 5

[KLE11] KIM K., LEE D., ESSA I.: Gaussian process regression flow for analysis of motion trajectories. In *IEEE International Conference on Computer Vision* (Nov. 2011), pp. 1164–1171. doi:10.1109/ICCV.2011.6126365. 2

[LJKM*17] LAVIOLA JR. J. J., KRUIJFF E., MCMAHAN R. P., BOWMAN D. A., POUPYREV I.: *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2017. 5

[LSSA12] LUBER M., SPINELLO L., SILVA J., ARRAS K.: Socially-aware robot navigation: A learning approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2012), pp. 902–907. doi:10.1109/IROS.2012.6385716. 2, 8

[MHSM*21] MERCADO V. R., HOWARD T., SI-MOHAMMED H., ARGELAGUET F., LÉCUYER A.: Alfred: the Haptic Butler On-Demand Tangibles for Object Manipulation in Virtual Reality using an ETHD. In *2021 IEEE World Haptics Conference (WHC)* (July 2021), pp. 373–378. doi:10.1109/WHC49131.2021.9517250. 1

[MIÇB19] MAKANSI O., ILG E., ÇIÇEK Ö., BROX T.: Overcoming Limitations of Mixture Density Networks: A Sampling and Fitting Framework for Multimodal Future Prediction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), pp. 7137–7146. https://ieeexplore.ieee.org/abstract/document/8953435. doi:10.1109/CVPR.2019.00731. 2

[MLSL19] MAO W., LIU M., SALZMANN M., LI H.: Learning Trajectory Dependencies for Human Motion Prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2019), pp. 9488–9496. ISSN: 2380-7504. URL: https://ieeexplore.ieee.org/document/9009559, doi:10.1109/ICCV.2019.00958. 2

[MML21] MERCADO V. R., MARCHAL M., LÉCUYER A.: "Haptics On-Demand": A Survey on Encountered-Type Haptic Displays. *IEEE Transactions on Haptics 14*, 3 (July 2021), 449–464. doi:10.1109/TOH.2021.3061150. 1

[MVVK23] MORTEZAPOOR S., VASYLEVSKA K., VONACH E., KAUFMANN H.: Cobodeck: A large-scale haptic vr system using a collaborative mobile robot. In *IEEE Conference on Virtual Reality* (2023). 1, 2

[NM18] NIKHIL N., MORRIS B.: Convolutional Neural Networkfor Trajectory Prediction. In *European Conference on Computer Vision (ECCV) Workshops* (2018). 2

[PS15] PEREZ-D'ARPINO C., SHAH J. A.: Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA, USA, May 2015), IEEE, pp. 6175–6182. http://ieeexplore.ieee.org/document/7140066/. doi:10.1109/ICRA.2015.7140066. 2, 8

[SHZ*20] SUZUKI R., HEDAYATI H., ZHENG C., BOHN J. L., SZAFIR D., DO E. Y.-L., GROSS M. D., LEITHINGER D.: RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2020), CHI '20, Association for Computing Machinery, pp. 1–11. URL: https://doi.org/10.1145/3313831.3376523, doi:10.1145/3313831.3376523. 1

[SZDZ17] SU H., ZHU J., DONG Y., ZHANG B.: Forecast the Plausible Paths in Crowd Scenes. In *International Joint Conference on Artificial Intelligence* (Aug. 2017), pp. 2772–2778. doi:10.24963/ijcai.2017/386. 2

[Tay16] TAYLOR C. R.: Finite difference coefficients calculator. https://web.media.mit.edu/~crtaylor/calculator.html, 2016. 3

[TK10] TRAUTMAN P., KRAUSE A.: Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2010), pp. 797–803. https://ieeexplore.ieee.org/document/5654369. doi:10.1109/IROS.2010.5654369. 2

[TL08] TAY M. K. C., LAUGIER C.: Modelling Smooth Paths Using Gaussian Processes. In *Field and Service Robotics: Results of the 6th International Conference*, Laugier C., Siegwart R., (Eds.). Springer, Berlin, Heidelberg, 2008, pp. 381–390. https://doi.org/10.1007/978-3-540-75404-6_36. doi:10.1007/978-3-540-75404-6_36. 2, 8

[TMMK13] TRAUTMAN P., MA J., MURRAY R., KRAUSE A.: Robot navigation in dense human crowds: The case for cooperation. In *IEEE International Conference on Robotics and Automation* (May 2013), pp. 2153–2160. doi:10.1109/ICRA.2013.6630866. 2

[UPSS15] UNHELKAR V. V., PÉREZ-D'ARPINO C., STIRLING L., SHAH J. A.: Human-robot co-navigation using anticipatory indicators of human walking motion. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (May 2015), pp. 6183–6190. doi:10.1109/ICRA.2015.7140067. 2, 5, 7, 8

[VGK17] VONACH E., GATTERER C., KAUFMANN H.: VRRobot: Robot Actuated Props in an Infinite Virtual Environment. In *Proceedings of IEEE Virtual Reality 2017* (Los Angeles, CA, USA, 2017), IEEE, pp. 74–83. URL: http://ieeexplore.ieee.org/document/7892233/, doi:10.1109/VR.2017.7892233. 1

[WMS21] WANG A., MAKINO Y., SHINODA H.: Machine Learning-based Human-Following System: Following the Predicted Position of a Walking Human. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (May 2021), pp. 4502–4508. ISSN: 2577-087X. URL: https://ieeexplore.ieee.org/abstract/document/9561691, doi:10.1109/ICRA48506.2021.9561691. 2, 8

[XWF15] XIAO S., WANG Z., FOLKESSON J.: Unsupervised robot learning to predict person motion. In *IEEE International Conference on Robotics and Automation* (USA, June 2015), vol. 2015, IEEE. doi:10.1109/ICRA.2015.7139254. 2

[YHK96] YOKOKOHJI Y., HOLLIS R. L., KANADE T.: What You Can See Is What You Can Feel - Development of a Visual/Haptic Interface to Virtual Environment. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium* (1996), pp. 46–53. 1

[YYY*16] YOO Y., YUN K., YUN S., HONG J., JEONG H., CHOI J. Y.: Visual Path Prediction in Complex Scenes with Crowded Moving Objects. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, June 2016), IEEE, pp. 2668–2677. http://ieeexplore.ieee.org/document/7780661/. doi:10.1109/CVPR.2016.292. 2