Comparison of Target Prediction in VR and MR using Inverse Reinforcement Learning

Mukund Mitra
Indian Institute of Science
Bangalore, India
mukundmitra@iisc.ac.in

Preetam Pati
Indian Institute of Information Technology, Kalyani
Kolkata, India
preetam6teen@gmail.com

Vinay Krishna Sharma
Indian Institute of Science
Bangalore, India
vinaysharma@iisc.ac.in

Subin Raj
Indian Institute of Science
Bangalore, India
subinp@iisc.ac.in

Partha Pratim Chakrabarti
Indian Institute of Technology, Kharagpur
Bangalore, India
ppchak@cse.iitkgp.ac.in

Pradipta Biswas
Indian Institute of Science
Bangalore, India
pradipta@iisc.ac.in

ABSTRACT

Numerous research were undertaken to predict pointing targets in Graphical User Interfaces (GUI). This paper extends target prediction for Extended reality (XR) platforms through Sampling-based Maximum Entropy Inverse Reinforcement Learning (SMEIRL). The SMEIRL algorithm learns the underlying reward distribution for the pointing task. Results show that SMEIRL achieves better accuracy in both VR and MR (for example 32.60% accuracy in VR and 34.48% accuracy in MR at 30% of pointing task) compared to Artificial Neural Network (ANN) and Quadratic Extrapolation (QE) during early stage of pointing task. For later stage, QE performs better (for example 93.51% accuracy in VR and 95.58% accuracy in MR at 70% of the pointing task) than SMEIRL and ANN. All the three algorithms, SMEIRL, ANN and QE reported higher target prediction accuracy in MR than in VR.

CCS CONCEPTS

• **Human-centered computing** → *Pointing*;

KEYWORDS

Target Prediction, Virtual and Mixed Reality, Neural Networks, Inverse Reinforcement Learning

ACM Reference Format:

Mukund Mitra, Preetam Pati, Vinay Krishna Sharma, Subin Raj, Partha Pratim Chakrabarti, and Pradipta Biswas. 2023. Comparison of Target Prediction in VR and MR using Inverse Reinforcement Learning. In *28th International Conference on Intelligent User Interfaces (IUI '23 Companion), March 27–31, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3581754.3584130

IUI '23 Companion, March 27–31, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0107-8/23/03.
https://doi.org/10.1145/3581754.3584130

1 INTRODUCTION

Target prediction is widely used in automotive [2] and assistive technologies [10] to help users select target in a GUI. It reduces pointing time in XR systems [6] [9] when users miss physical touch due to the absence of haptic feedback in the system. Earlier research on target prediction used velocity extrapolation [3], classical kinematic [8] and template matching approaches [11]. With the advancement of machine learning techniques and powerful head mounted display, researchers are implementing machine learning models for target prediction in VR [6] and MR [13] [5]. However, there is not much reported research which implements and compares target prediction accuracy in VR and MR for the same pointing task. Hence, the paper investigates the following research questions: *Do change in immersive media effect target prediction accuracy for a pointing task? If yes, can we leverage this change to increase the target prediction accuracy?* In this paper, we leveraged the fact that Inverse reinforcement Learning (IRL) [16] can be used to model and reason human behaviour by learning the underlying reward distribution for a pointing task [15]. We implemented Sampling-based Maximum Entropy Inverse Reinforcement Learning (SMEIRL) [14] which performs better prediction and is less computationally expensive [14] than other IRL algorithms, for target prediction. We compared target prediction accuracy of the IRL based model in VR and MR with Quadratic Extrapolation [8] and Artificial Neural Networks [4].

2 DATA COLLECTION

The experiment extends the standard ISO 9241 pointing task [7]. The task consists of sequentially selecting a central red ball and peripheral white balls. The size and distance of white ball was randomly selected from three sets of sizes (1.5,2.0,2.5 cm diameter), and distance (5,10,15 cm) from the centre red ball. A total of 30 iterations (center red ball to target white ball selections) to be completed for a successful session for each participant using VR and MR interfaces. Data was collected from 13 participants with 26.4 years average age.

Data collection in VR: Unity3D is used to design a virtual reality environment. A real scale model of the room was created using unity assets and textures were added to give a realistic background

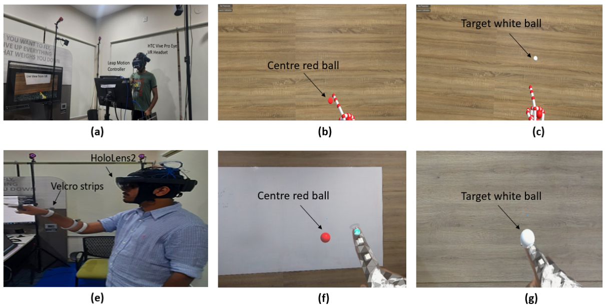

Figure 1: (a) Experimental setup using VR interface, (b) and (c) HTC Vive Pro view of the pointing task in VR. (d) Experimental setup using MR interface, (e) and (f) HoloLens2 view of the pointing task in MR.

for the task. The VR interface ran on HTC Vive Pro Eye virtual reality headset. The hand interaction in VR interface was achieved by using commercially available Leap Motion Controller from UltraLeap. Leap motion was mounted vertically on the VR headset facing forwards. This allowed for accurate and seamless tracking on hands. Once the hands are tracked, their accurate digital models are rendered in VR scene providing realistic hand interactions in VR. The participants used these digital hand models in VR to interact with the center and target balls during the experiment. The experimental setup and HTC Vive Pro view is shown in Figure 1(a), (b) and (c) respectively.

Data collection in MR: Microsoft HoloLens2 to create the MR environment. HoloLens2 has eye tracking and hand tracking capability. MR environment made with the help of the Mixed Reality Toolkit (MRTK) and Unity game engine. MRTK hand tracking and eye tracking profile allows the user to interact with holograms. The experimental setup and HoloLens2 view is shown in Figure 1(c), (d) and (e) respectively.

To accurately track and record hand movement of the participants we have used the OptiTracker Motion Capture Software Motive 1.10.1 along with four Flex 13 cameras for both VR and MR interface. Leap Motion was only used for rendering digital hand model in VR. Tracking was done using motion capture system.

3 TARGET PREDICTION ALGORITHMS

Sampling-based Maximum Entropy Inverse Reinforcement Learning: Since, we are learning the reward distribution of the pointing task done by humans, we say the dataset collect by users as expert dataset, and the corresponding pointing trajectories as expert trajectories. Let s denote the state of human hand and τ denote a trajectory containing sequence of states, i.e., $\tau = [s_0, s_1, ...s_{N-1}]$ where N is the length of the trajectory. Given a set of expert trajectories $\Xi_D = \{\tau_i\}$ with $i = 1, 2, ..., M$, the maximum entropy IRL [16] recovers the underlying reward function which maximises the likelihood of the trajectories. A reward function with a selected feature space $\mathbf{f}(.)$ defined over the trajectory τ is given by: $R(s_n, \omega) = \omega^T f(s_n)$. In the proposed setup, the feature space $\mathbf{f}(.)$ is designed by assigning higher values to states visited by the expert trajectories and the vector ω specifies the weight of feature contributing to the reward function R. The likelihood of expert demonstrations is given by:

$$P(\Xi_D|\,\omega) = \prod_{i=1}^{M} \frac{e^{R(\tau_i,\omega)}}{\int_{\tau_i \in \Xi_D} e^{R(\tau_i,\omega)} d\tau} = \frac{1}{Z} e^{R(\tau_i,\omega)} \tag{1}$$

where Z is given by $Z = \int_{\tau \in \Xi_D} e^{R(\tau,\omega)} d\tau$. The optimal ω^* which maximizes the averaged log-likelihood of the expert demonstrations is given by:

$$\omega^* = \underset{\omega}{argmax} \frac{1}{M} logP(\Xi_D|\omega) = \underset{\omega}{argmax} \frac{1}{M} \sum_{i=1}^{M} \{logP(\tau_i|\omega)\} \tag{2}$$

In sampling based methods, Z for each set of expert trajectories is approximated via summation over all the sample trajectories $\psi_m = \left\{s_0^{'}, s_1^{'}, ..., s_{N-1}^{'}\right\}$ in the sample set $\{\psi_m\}, m = 1, 2, ..., K$, given by: $Z \approx \sum_{m=1}^{K} e^{R(\psi_m,\omega)}$, where $s^{'}$ is a state visited by sample trajectory. For generating sample trajectories, we fit linear, quadratic and cubic polynomials on the training data. We see that quadratic curve has least average error (8.25mm) compared to linear and cubic. Thus we use quadratic sample trajectories. The objective function for the optimization problem given in (2) becomes:

$$L(\omega) = \frac{1}{M} \sum_{i=1}^{M} logP(\tau_i|\omega) = \frac{1}{M} \sum_{i=1}^{M} \left\{ R(\tau_i, \omega) - log \sum_{m=1}^{K} e^{R(\psi_m^i,\omega)} \right\} \tag{3}$$

Where L is the objective function. ψ_m^i is the set of all sampled trajectories having the same start and target position as τ_i. The gradient of the objective function is obtained by partial derivative of $L(\omega)$ with respect to ω is given by:

$$\nabla_\omega L = \frac{1}{M} \sum_{i=1}^{M} \left\{ f(\tau_i) - \tilde{f}\left(\psi_m^i, \omega\right) \right\} \tag{4}$$

where, $\tilde{f}(\psi_m, \omega)$ is given by:

$$\tilde{f}(\psi_m, \omega) = \sum_{m=1}^{K} \frac{e^{R(\psi_m,\omega)}}{\sum_{m=1}^{K} e^{R(\psi_m,\omega)}} f(\psi_m, \omega) \tag{5}$$

and $\tilde{f}(\psi_m, \omega)$ is the expected sample feature count of states over all sample trajectories. $f(\psi_m, \omega)$ is given by:

$$f(\psi_m, \omega) = \sum_{s_n^{'} \epsilon \psi_m} f\left(s_n^{'}, \omega\right) \tag{6}$$

The above optimization problem is solved using gradient descent [12] until convergence. For target prediction, all the 30 target white balls were placed together. Rewards are obtained for a given a partial pointing trajectory. Rewards are obtained for the partial trajectories of same length (as that of given pointing trajectory) derived from learned reward distribution, corresponding to each target. The target corresponding to the least difference in total reward is the predicted target. Video of target prediction is available at Link.

Artificial Neural Network (ANN): We used an encoder-decoder LSTM artificial neural network with hidden layer dimension 20, adam optimizer and mean square error loss. For training, batch size was set to 65 and number of epochs to 20. The X,Y,Z coordinates of the index finger of the partial trajectory were the input and corresponding shifted time stamp coordinates were the output of the network. Pytorch 1.13.0 was used for the implementation.

Quadratic Extrapolation (QE): We implemented quadratic extrapolation [8] to fit a quadratic function to the total length of the partial trajectory and extrapolate it to predict target position.

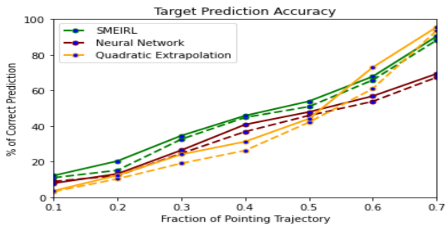

Figure 2: Variation of percentage of correct target prediction with fraction of input pointing trajectory. Dashed lines shows accuracy in VR and solid lines represent accuracy in MR

4 RESULT AND DISCUSSION

We compared our approach of target prediction using SMEIRL in VR and MR with QE and ANN. The accuracy of target prediction is defined as:

$$accuracy = \frac{Number\ of\ correct\ target\ predictions}{Total\ number\ of\ predictions} \tag{7}$$

The target prediction accuracy is shown in Figure 2. Following relevant observations could be made:

- At early stage (30%) of pointing task, SMEIRL performs better with 32.60% accuracy in VR and 34.48% accuracy in MR than ANN and QE. At later stage of pointing task (70%), QE performs better with 93.51% accuracy in VR and 95.58% accuracy in MR than SMEIRL and ANN. From Figure 2, we see that SMEIRL shows higher prediction accuracy till 55% of the pointing task. QE performs better than SMEIRL from 55 − 70% of the pointing task. From Table-1, we see that target prediction accuracy is higher in MR than VR using all the three methods. For example, at 30% of the pointing task, SMEIRL reports an accuracy of 34.48% in MR and 32.60% in VR, ANN reports an accuracy of 26.41% in MR and 24.52% in VR, and QE reports an accuracy of 24.10% in MR and 19.06% in VR.
- The proposed SMEIRL model achieves best target prediction accuracy of 90.37% in MR and 88.21% in VR given 70% of pointing trajectory. This is better than Dalsgaard's model [6] which achieves best prediction accuracy of 87% for pointing task in VR using Support Vector Machine (SVM) and best set of pointing features.
- Ziebart's model [15] reports 90% target classification error (or 10% accuracy) at 30% of the pointing task and Biswas's model [4] reports an accuracy of 22% and 13% for head and eye-gaze based prediction at 30% of the pointing task respectively, in 2D. SMEIRL performs better by achieving an accuracy of 32.60% in VR and 34.48% in MR at 30% of the pointing task. For the same pointing task used in this work, in 2D, Bashar [1] reported best target prediction of 62.1% using mean reverting diffusion model with linear kalman filter based smoothing for able-bodied users, whereas SMEIRL achieves best target prediction accuracy of 88.21% in VR and 90.37% in MR at 70% of the pointing task.

Table 1: Comparison of target prediction accuracy in VR and MR

	VR		MR	
Method	Early stage (30%)	Late stage (70%)	Early stage (30%)	Late stage (70%)
SMEIRL	32.60	88.21	34.48	90.37
ANN	24.52	67.39	26.41	69.24
QE	19.06	93.51	24.10	95.58

5 CONCLUSION

We implemented SMEIRL algorithm to predict targets using the learned underlying reward distribution of the pointing task. We compared target prediction accuracy using SMEIRL with ANN and QE for the pointing task in VR and MR. We found that SMEIRL achieves better target prediction accuracy than ANN and QE for initial 55% of the pointing task and QE performs better thereafter. The target prediction accuracy is higher in MR compared to VR using SMEIRL, ANN and QE. Future work includes testing the performance of the algorithm for unseen environments, exploring various factors responsible for change in target prediction accuracy and leveraging those factor to improve target prediction.

REFERENCES

[1] Bashar I Ahmad, Patrick M Langdon, Pete Bunch, and Simon J Godsill. 2014. Probabilistic intentionality prediction for target selection based on partial cursor tracks. In *International Conference on Universal Access in Human-Computer Interaction*. Springer, 427–438.

[2] Bashar I Ahmad, James K Murphy, Patrick M Langdon, Simon J Godsill, Robert Hardy, and Lee Skrypchuk. 2015. Intent inference for hand pointing gesture-based interactions in vehicles. *IEEE transactions on cybernetics* 46, 4 (2015), 878–889.

[3] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino. 2005. Predictive interaction using the delphian desktop. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. 133–141.

[4] Pradipta Biswas and Patrick Langdon. 2014. Multimodal target prediction model. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 1543–1548.

[5] Yujin Choi, Wookho Son, and Yoon Sang Kim. 2021. A Study on Interaction Prediction for Reducing Interaction Latency in Remote Mixed Reality Collaboration. *Applied Sciences* 11, 22 (2021), 10693.

[6] Tor-Salve Dalsgaard, Jarrod Knibbe, and Joanna Bergström. 2021. Modeling Pointing for 3D Target Selection in VR. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. 1–10.

[7] Sarah A Douglas, Arthur E Kirkpatrick, and I Scott MacKenzie. 1999. Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 215–222.

[8] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. 2007. Endpoint prediction using motion kinematics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 637–646.

[9] George B Mo, John J Dudley, and Per Ola Kristensson. 2021. Gesture Knitter: A Hand Gesture Design Tool for Head-Mounted Mixed Reality Applications. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[10] Kevin C Olds, Sara Sibenaller, Rory A Cooper, Dan Ding, and Cameron Riviere. 2008. Target prediction for icon clicking by athetoid persons. In *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2043–2048.

[11] Phillip T Pasqual and Jacob O Wobbrock. 2014. Mouse pointing endpoint prediction using kinematic template matching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 743–752.

[12] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747* (2016).

[13] Shuai Wang, Ruifeng Guo, Hongliang Wang, Yuanjing Ma, and Zixiao Zong. 2018. Manufacture assembly fault detection method based on deep learning and mixed reality. In *2018 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 808–813.

[14] Zheng Wu, Liting Sun, Wei Zhan, Chenyu Yang, and Masayoshi Tomizuka. 2020. Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving. *IEEE Robotics and Automation Letters* 5, 4 (2020), 5355–5362.

[15] Brian Ziebart, Anind Dey, and J Andrew Bagnell. 2012. Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. 1–10.

[16] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.