International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2024)
T. Tanabe and V. Yem (Editors)

Poster

Advanced Motion Prediction for Virtual Reality Gaming: a CNN-Based Approach

Hubert Jegierski[1], Maciej Jegierski[1], Adrian Łapczyński[1], Paweł Babiuch[1],
Mirosław Płaza[2], Paweł Pięta[†][2] Grzegorz Łukawski[2], Stanisław Deniziak[2], Jacek Opałka[2], Artur Jasiński[2],
Magdalena Igras-Cybulska[3] and Paweł Węgrzyn[4]
[1]EpicVR sp. z o.o., Cracow, Poland
[2]Kielce University of Technology, Department of Information Systems, Kielce, Poland
[3]AGH University of Science and Technology, Institute of Electronics, Cracow, Poland
[4]Jagiellonian University, Department of Games Technology, Cracow, Poland

---

Abstract

A novel motion prediction model (MPM) for virtual reality (VR) video games was developed, consisting of a motion recognition model (MRM) and a next movement prediction model (NMPM), both using convolutional neural networks (CNNs). Motion capture was performed with HTC Vive Pro and Meta Quest 2. Two custom datasets were created to train the MRM and NMPM. Our method achieved a top-1 accuracy of 77% and a top-2 accuracy of 90%, even with motion data sequences sharing similar initial stages but diverging in subsequent movements.

CCS Concepts

• ***Computing methodologies*** → ***Virtual reality; Neural networks;*** • ***Hardware*** → ***Sensor devices and platforms;***

---

1. Introduction

Predicting player activity in VR environments is challenging due to the complexity and variability of human motion, influenced by individual physical qualities [Y*19], motion speed, and interaction context [S*24]. High-precision predictions enable games to respond more fluidly to player actions, resulting in smoother interactions, reduced latency, and a more immersive experience, particularly in embodied VR [SN18].

This paper introduces a novel MPM designed for VR gaming, utilizing CNNs for accurate motion recognition and next movement prediction. Unlike traditional MoCap systems that require extensive hardware (e.g., Leap Motion in [V*19]), our model leverages VR components (goggles and controllers), making it more accessible and scalable. Our work offers insights into the use of deep learning applications in VR gaming and emphasizes potential future developments. This research is part of a larger project supporting multiplayer VR game development [B*24]. Detailed results on single gesture recognition are available in [P*24], and a comprehensive study on developers' needs is summarized in [IC*24].

2. Material and Methods

2.1. Data

Motion data were collected from 19 video game players using HTC Vive Pro and Meta Quest 2 VR headsets, resulting in 5,649 valid recordings across 56 motion classes: head, hand/arm, whole body, and animations. Data were captured at 80Hz and stored as JSON files with positional and rotational data from the head (via goggles) and hands (via controllers). A Unity-based application ensured accurate tracking. The recordings were preprocessed into a motion recognition dataset (MRD), as illustrated in Figure 1.

Additionally, a motion prediction dataset (MPD) was created from 1,442 valid recordings in 8 scenarios, each consisting of 3–15 movements from predefined classes, and transformed as shown in Figure 1. The MRD contains single motions per recording, while the MPD includes sequences, with 27% of moves having one possible next move, 24% with two, and 49% with three or more.

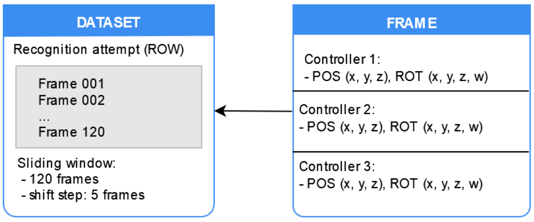

Figure 1: *Preprocessing process: POS – position, ROT – rotation.*

2.2. Methods

Our MPM consists of two components: the MRM and NMPM (Figure 2). Real-time input from a VR headset is taken by the MRM, saved in a buffer for temporal analysis, and processed to identify the current motion. The recognized motion is passed to the NMPM,

---

[†] Corresponding author, e-mail: p.pieta@tu.kielce.pl

DOI: 10.2312/egve.20241379

which uses a history of recent movements to predict the next motion. An end-to-end architecture was not chosen for the MPM, because our research project involved at first the MRM, thus the substantial number of labeled recordings gathered in the MRD.

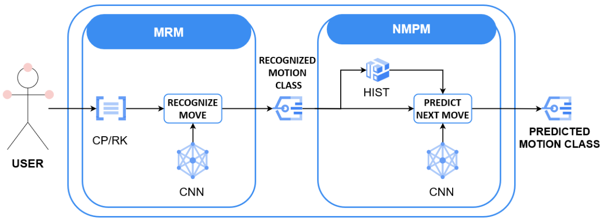

Figure 2: *MPM components: CP/RK – current position / rotation of the key body parts, HIST – history of 5 recent movements.*

The MRM is a CNN (Table 1) with six convolutional layers (32, 64, 128, 256, 512, and 1024 filters), each with a window size of 3 and ReLU activation function (AF). A max pooling layer (pool size 2) is followed by five fully connected (FC) layers, with 4096, 2048, 1024, and 128 neurons using ReLU AF, while the final layer has 56 neurons and uses softmax AF for class output.

The NMPM, also a CNN (Table 1), predicts the next move using the previous five moves. It follows a similar architecture with two convolutional layers (32 and 64 filters, window size 3, ReLU AF), a max pooling layer, and three FC layers (64, 32, and 56 neurons), with the final layer using softmax AF for class output.

Table 1: *MPM architecture: BN – BatchNormalization, DO – Dropout, LN – LayerNormalization, MP1D – MaxPooling1D.*

Model	Layers	Kernel	Stride	Filters	Layer	AF
MRM	Conv1D & BN	3 × 3	2	32	-	ReLU
	Conv1D & BN	3 × 3	2	64	-	ReLU
	Conv1D & BN	3 × 3	2	128	-	ReLU
	Conv1D & BN	3 × 3	2	256	-	ReLU
	Conv1D & BN	3 × 3	2	512	-	ReLU
	Conv1D & BN	3 × 3	2	1024	-	ReLU
	MP1D & DO	2 × 2	1	-	-	-
	Flatten	-	-	-	-	-
	Dense & DO	-	-	-	4096	ReLU
	Dense & DO	-	-	-	2048	ReLU
	Dense & DO	-	-	-	1024	ReLU
	Dense & DO	-	-	-	128	ReLU
	Dense	-	-	-	56	Softmax
NMPM	Conv1D & LN	3 × 3	1	32	-	ReLU
	Conv1D & LN	3 × 3	1	64	-	ReLU
	MP1D	2 × 2	1	-	-	-
	Flatten	-	-	-	-	-
	Dense	-	-	-	64	ReLU
	Dense	-	-	-	32	ReLU
	Dense	-	-	-	56	Softmax

3. Experiments

For MRM training, Adam optimizer was used with a learning rate of 10^{-4} and a weight decay of 0.004 over 10 epochs, with a batch size of 32. For NMPM training, Nadam optimizer was applied with a learning rate of 10^{-4}, running for 100 epochs. The dataset was divided into training (60%) and test (40%) sets. We used top-N accuracy as the evaluation metric, where a prediction is correct if the true value is among the N most probable responses. Top-N accuracy was assessed for N=1, 2, 3, and 5, calculated as:

$$\text{top-N accuracy} = \frac{1}{K} \sum_{i=1}^{K} \sum_{j=1}^{N} \mathbb{I}(y_{i,j} = y_{true}) \tag{1}$$

To address data imbalance, weighted precision, recall, and F1 scores [P*24] were also calculated.

4. Results

Table 2: *Obtained results: T-NA – Top-N Accuracy, REC – Recall, PR – Precision, F1 – F1 Score.*

Dataset	T-1A	T-2A	T-3A	T-5A	REC	PR	F1
Training	0.85	0.92	0.96	0.98	0.85	0.87	0.83
Test	0.77	0.90	0.94	0.97	0.77	0.78	0.75

Table 2 summarizes the results of experiments. We employed the top-N accuracy metric to account for sequences with identical initial stages but different subsequent moves, allowing us to evaluate multiple likely outcomes. This approach ensures a fair assessment of the model's performance given the data's complexity.

The model achieved strong precision and recall across both datasets, with slightly lower but still solid performance on the test set. In particular, the F1 score on the test data is 0.75, reflecting a good balance between precision and recall.

5. Discussion and Conclusion

The results highlight the robustness of the developed MPM, achieving a top-1 accuracy of 77% on the test dataset despite the inherent ambiguity of the data, where multiple consecutive moves can be valid as mentioned in Section 2.1. The top-2 accuracy further supports this, reaching 90% in predicting the next move.

Beyond gaming, accurate motion prediction can enhance safety and comfort in VR experiences by anticipating player movements, thus reducing risks of collisions with real-world objects. This capability also optimizes system performance, ensuring smooth operation during complex player actions. Additional applications include exergames [C*21] and rehabilitation [M*23], where precise monitoring of motion sequences is essential.

References

[B*24] Babiuch P., et al.: MUN: an AI-powered multiplayer networking solution for VR games. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (2024), pp. 1204–1205. 1

[C*21] Caserman P., et al.: Full-body motion recognition in immersive-virtual-reality-based exergame. *IEEE Transactions on Games 14*, 2 (2021), 243–252. 2

[IC*24] Igras-Cybulska M., et al.: Supporting Unity developers with an AI-powered asset: Insights from an exploratory user study on multiplayer game development. In *Proceedings of the 1st International Workshop on Designing and Building Hybrid Human–AI Systems (SYNERGY 2024)* (2024). 1

[M*23] Maskeliūnas R., et al.: BiomacVR: A virtual reality-based system for precise human posture and motion analysis in rehabilitation exercises using depth sensors. *Electronics 12*, 2 (2023), 339. 2

[P*24] Piętа P., et al.: Automated classification of virtual reality user motions using a motion atlas and machine learning approach. *IEEE Access 12* (2024), 94584–94609. 1, 2

[S*24] Skrzek T., et al.: Automatic recognition of gesture qualities in speech delivery in VR. In *2024 IEEE VR Conference* (2024), IEEE, pp. 624–627. 1

[SN18] Smith H. J., Neff M.: Communication behavior in embodied virtual reality. In *Proceedings of the 2018 ACM CHI Conference* (2018), pp. 1–12. 1

[V*19] Vaitkevičius A., et al.: Recognition of american sign language gestures in a virtual reality using leap motion. *Applied Sciences 9*, 3 (2019), 445. 1

[Y*19] Yang L., et al.: Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware 1*, 1 (2019), 84–112. 1