OpenXR-Based Hand Motion Recording for VR Applications using Unity and VR Headset

Marco Giammetti[1,*,†], Andrea Antonio Cantone[1,†] and Pietro Battistoni[1,†]

[1] *Department of Computer Science, University of Salerno, 84084 Fisciano (SA), Italy*

Abstract

This paper presents VR Hands Recording, a Unity-based system designed to simplify the recording of hand motion animations in virtual reality environments. Built on the OpenXR standard, the system ensures broad compatibility with various hand-tracking-enabled headsets, including Meta Quest and Pico devices. Users can intuitively record hand gestures through a preconfigured Unity scene and export animations in FBX format for seamless integration into 3D graphics software such as Blender. Compared to existing solutions, VR Hands Recording offers a flexible, cost-effective, and cross-platform alternative suitable for creative production, human-computer interaction research, and XR prototyping. The system has been evaluated both quantitatively and qualitatively, demonstrating high usability, precision, and ease of integration into post-production pipelines.

Keywords

Hand Tracking, Virtual Reality, OpenXR, Unity Animation Pipeline

1. Introduction

The VR Hands Recording project was developed with the aim of simplifying the process of capturing hand movement animations in virtual reality environments. By leveraging Unity in combination with the hand tracking of the compatible devices and native recording packages, the system enables intuitive and accurate recording of user hand gestures. These animations can then be exported in widely supported formats, such as FBX, for seamless integration into 3D graphics software like Blender. Hand gesture recording plays a key role in co-creating virtual environments for therapeutic applications [1][2].

The importance of natural hand-based interaction in immersive VR has been widely recognized, as it enhances user presence, embodiment, and engagement across various application domains [3][4]. In this context, providing developers and researchers with accessible tools to efficiently capture hand motions is critical to enabling realistic interaction design and content production workflows [5][6].

Built on the OpenXR standard, the solution offers broad compatibility with a wide range of VR headsets supporting hand tracking, ensuring both flexibility and scalability. It provides a ready-to-use pipeline, particularly valuable for creative teams, multimedia production, human-computer interaction research, and AI-driven applications [7, 8]. Notably, this system has already been used to generate hand animation content for MEMORI SRL, as showcased in a publicly released video on LinkedIn[1].

2. Related work

Several solutions have been proposed for recording hand movements in virtual reality, each offering different trade-offs in terms of compatibility, licensing, and integration flexibility. Among the most

Sense-XR 2025 - The Feeling of Virtual, XR, Haptics, and 3D Graphics, a workshop co-located with CHItaly 2025 (Salerno, Italy, 6–10 October 2025)

[*] Corresponding author.

[†] These authors contributed equally.

thetmo@gmail.com (M. Giammetti); acantone@unisa.it (A. A. Cantone); pbattistoni@unisa.it (P. Battistoni)
xrlab.it (M. Giammetti)
0009-0003-1631-6739 (M. Giammetti); 0009-0001-0270-5106 (A. A. Cantone); 0000-0002-7477-2439 (P. Battistoni)

[1] https://www.linkedin.com/posts/memorisrl_aisuruminiguide-intelligenzaartificiale-ai-activity-7351129108208136192-_eya

notable are the following.

2.1. Meta-Quest-Hand-Tracking-Motion-Recording

This open-source project [2], available on GitHub, allows users to record hand tracking data using Meta Quest headsets (Quest 2, Quest 3, and Quest Pro). The system supports FBX export and is designed specifically for Unity environments.

However, the main limitation of this solution lies in its dependency on Meta's proprietary SDK and lack of support for non-Meta devices. Furthermore, it is not based on the OpenXR standard, reducing its portability across XR ecosystems.

2.2. Glycon3D

Glycon3D [3] is a commercial hand and body motion capture solution compatible with multiple platforms, including Meta Quest, SteamVR, and HTC Vive. It supports FBX and BVH export formats and offers a professional-grade recording environment.

While highly capable and flexible, Glycon3D requires a commercial license (starting at $99), which may not be accessible for small teams, educational use, or prototyping scenarios.

3. System Architecture

The VR Hands Recording system is designed to provide a seamless and modular framework for capturing hand movements in virtual reality. It relies on widely adopted tools and standards to ensure ease of use, cross-device compatibility, and integration with post-production pipelines.

3.1. Software and Frameworks

The system is implemented using the following software components:

- Unity: version 2022.3.49f1 (LTS), serving as the development and execution environment.
- XR Framework: OpenXR (via the Mixed Reality Toolkit), ensuring broad compatibility across XR devices.
- Unity Packages. **Recorder**: used for capturing animations directly within the Unity Editor. **FBX Exporter**: enables export of recorded animations to FBX format, compatible with 3D graphics software. **UnityGLTF**: enables export FBX file for a more wide compatible GLB/GLTF file https://github.com/KhronosGroup/UnityGLTF

This software stack allows users to record animations with minimal setup while maintaining flexibility for later customization.

3.2. Supported Hardware

The system supports a variety of headsets equipped with hand tracking capabilities, including:

- Meta Quest 2
- Meta Quest 3 / 3S
- Meta Quest Pro
- Pico (any model with hand tracking support)

All headsets must be connected to a PC (e.g., via Meta Quest Link or equivalent methods if used on another headset) to run the Unity-based recording environment.

This architecture allows the system to operate across multiple hardware configurations while maintaining consistent performance and accuracy in gesture capture.

[2] https://github.com/richdrummer33/Meta-Quest-Hand-Tracking-Motion-Recording

[3] https://www.glycon3d.com

4. System Workflow and Technical Design

This section describes the general workflow available on the github repository[4] and the architectural choices behind the VR Hands Recording system. The goal is to provide a lightweight and flexible tool for capturing hand motion data in virtual reality environments, while ensuring compatibility with industry-standard 3D tools and diverse hardware platforms. We first present the operational workflow designed for end users, followed by the rationale behind the selection of technologies. We then detail the post-processing pipeline, and conclude by discussing technical challenges encountered during development and the corresponding solutions.

4.1. System Workflow

The system was designed to minimize user intervention while maintaining flexibility. Upon launching the Unity scene, the user is presented with a preconfigured XR rig with hand tracking and a visual mirror for live feedback (see Fig. 1). The recording process involves selecting the rig in the Recorder panel, pressing Play, performing hand movements using a compatible headset, and stopping the recording to export the animation in FBX format.

At project startup (see Fig. 2), the Unity environment presents the user with a preconfigured scene that includes :

- an XR rig with hand tracking enabled;
- a virtual mirror to visualize hand movements in real time;
- the Recorder panel for managing the capture process.

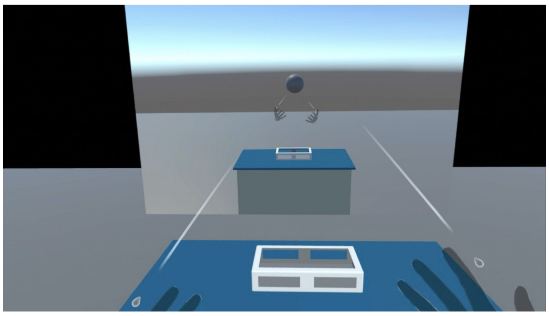


Figure 1: A preconfigured Unity XR rig with real-time hand tracking and a virtual mirror for user feedback.

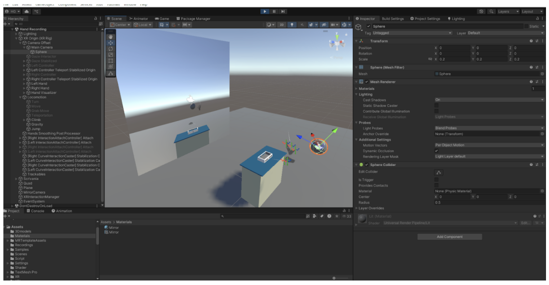


Figure 2: Overview of the Unity scene at startup, including the XR rig, virtual mirror, and Recorder panel for managing the capture process (see Figure 3)

To start recording:

- Select the FBX option in the Recorder Panel (see Figure 4).

[4] https://github.com/theTMO/VR-Hands-recording

- select the XR rig in the Recorder panel (see Figure 5).
- Press Play to start recording.
- Move your hands while wearing the headset.
- Press Stop to complete the capture.
- The generated file will be exported in `.fbx` format using the FBX Exporter package.

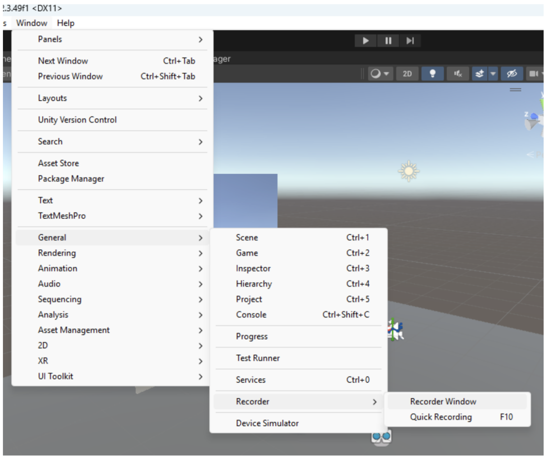
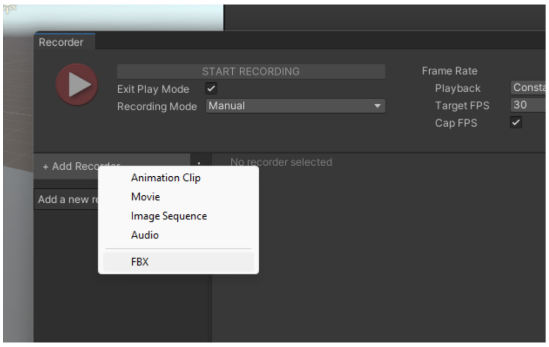


Figure 3: The Recorder panel with the XR rig selected.


Recorder

START RECORDING

Exit Play Mode ☑

Recording Mode Manual

Frame Rate

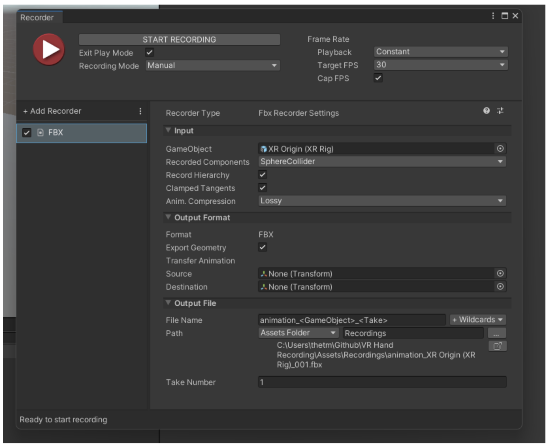
Playback Consta

Target FPS 30

Cap FPS ☑

No recorder selected

Figure 4: This option is not available unless the FBX Exporter is installed.


Recorder

START RECORDING

Exit Play Mode ☑

Recording Mode Manual

Frame Rate

Playback Constant

Target FPS 30

Cap FPS ☑

+ Add Recorder

☑ FBX

Recorder Type Fbx Recorder Settings

Input

GameObject XR Origin (XR Rig)

Recorded Components SphereCollider

Record Hierarchy ☑

Clamped Tangents ☑

Anim. Compression Lossy

Output Format

Format FBX

Export Geometry ☑

Transfer Animation

Source None (Transform)

Destination None (Transform)

Output File

File Name animation_<GameObject>_<Take> + Wildcards

Path Assets Folder Recordings

C:\Users\thetm\Github\VR Hand Recording\Assets\Recordings\animation_XR Origin (XR Rig)_001.fbx

Take Number 1

Ready to start recording

Figure 5: Recorder window configuration with the XR rig prefab selected.

To ensure compatibility with software such as Blender, Maya, and 3ds Max, the use of *UnityGLTF* is recommended:

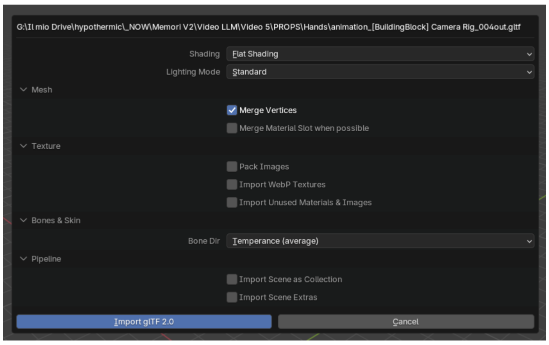
- it converts Unity `.fbx` files to formats such as `.GLB`, `.GLTF`;
- it preserves animations, materials, and skeletal hierarchy;
- it is free and available at: https://github.com/KhronosGroup/UnityGLTF

4.2. How to import in Blender

The following settings are recommended when importing the animation into Blender:

- **Shading:** Use Normal Data
- **Lighting Mode:** Standard
- **Merge Material Slot:** Enabled
- **Pack Images:** Enabled
- **Bone Dir:** Fortune (may look better, less accurate)
- **Import Scene as Collection:** Enabled
- **Select Imported Objects:** Enabled
- **Import Scene Extras:** Enabled

These parameters help prevent issues related to scale, rotation, or data loss (see Fig. 6).



G:\Il mio Drive\hypothermic_NOW\Memori V2\Video LLM\Video 5\PROPS\Hands\animation_[BuildingBlock] Camera Rig_004out.gltf

Shading: Flat Shading

Lighting Mode: Standard

Mesh

☑ Merge Vertices

□ Merge Material Slot when possible

Texture

□ Pack Images

□ Import WebP Textures

□ Import Unused Materials & Images

Bones & Skin

Bone Dir: Temperance (average)

Pipeline

□ Import Scene as Collection

□ Import Scene Extras

Figure 6: FBX export settings in the Unity Recorder panel, including selected parameters.

4.3. Technical Implementation and Design Choices

Unity was selected for its broad compatibility with major VR headsets on the market, particularly those based on the Qualcomm XR2 chipset (such as Meta Quest 2/3 and Pico Neo 3/4). It also provides native integration with recording and exporting plugins.

OpenXR is an open standard supported by all major VR hardware manufacturers. This ensures compatibility not only with Meta Quest, but also with other headsets that offer hand tracking support.

FBX was chosen as the export format due to its versatility and wide compatibility with most 3D animation software (including Blender, Maya, Unity, and Unreal). The subsequent conversion to glTF/glb reduces file size and ensures modern, web-friendly interoperability.

4.4. Challenges and Solutions

Minor technical challenges were encountered during development. In particular, unnecessary controllers and skeleton structures were sometimes included in the exported animation files. This was resolved by simplifying the Unity rig and disabling non-essential components prior to export.

Additionally, hand tracking quality was found to vary depending on the headset used (see Fig. 7). Devices such as the Meta Quest Pro and Pico 4 demonstrated high tracking accuracy, thanks to their support for the most recent version of the OpenXR Hand Tracking Interaction.

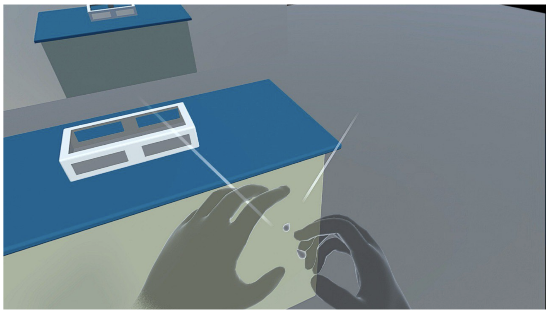


Figure 7: OpenXR hands provide a standardized representation of hand tracking, ensuring cross-device compatibility.

4.5. Post-Processing and Integration

Exported hand motion data in FBX format are post-processed using Noesis or imported into Blender using a specific configuration to preserve skeletal hierarchy and animation quality. Recommended import settings include enabling bone merging, normal data shading, and scene extras. This ensures visual consistency and reduces errors in scale and rotation.

4.6. Challenges and Solutions

During the development of the system, several minor issues were encountered:

- **Unnecessary controllers or skeletons in exported files.** This was addressed by simplifying the Unity rig and disabling non-essential components before exporting the animation.
- **Noisy hand tracking.** The quality of hand tracking heavily depends on the headset used. Devices such as the Meta Quest Pro and Pico 4 produced highly accurate results thanks to support for the latest version of the OpenXR Hand Tracking Interaction Profile [5].

5. Evaluation

5.1. Quantitative Evaluation

Tracking Accuracy. The system's accuracy depends on both the headset used and the environmental conditions. With the latest implementation of OpenXR (version 1.0.27+), the system is capable of tracking up to 26 joints per hand with millimeter-level precision, supporting pinching, grasping, and coherent 3D positioning [6].
Recording Time. Recording is immediate and real-time. Users can start and stop the recording process using the Unity Recorder panel with a single click.
Export/Import Performance.

- **FBX file generation:** 1–3 seconds, depending on animation length and hardware performance.
- **Conversion to .glb:** Instantaneous using Noesis or Blender.

5.2. Qualitative Evaluation

User Testing. The system was evaluated by digital creatives, designers, and XR developers. Feedback highlighted the tool's ease of use, smooth integration with Blender, and its versatility for generating animated content. Its effectiveness is further supported by previous XR-based training applications employing similar hardware and frameworks [9].

Usability. The familiar Unity interface, combined with the in-scene virtual mirror, provides users with direct visual feedback of their hand movements, effectively reducing the learning curve.

[5] https://registry.khronos.org/OpenXR/specs/1.0/man/html/XrHandTrackerEXT.html
[6] https://registry.khronos.org/OpenXR/specs/1.0/man/html/XrHandTrackerEXT.html

Table 1
Comparison of hand motion capture tools

Feature	VR Hands Recording	Glycon3D	Meta GitHub Project
Cost	Free	Commercial ($99)	Free (open-source)
Device Compatibility	Universal (OpenXR)	Quest, SteamVR	Meta Quest only
Output Format	FBX	FBX, BVH	FBX
Rig Flexibility	High	Medium	Limited to presets
Post-Processing	Blender-ready	Intermediate	Requires rig adaptation

5.3. Differences and comparisons

Table 2
Differencces and comparison of hand motion recording tools for VR

Project	**Framework**	**Device Compatibility**	**License**
Meta-Quest-Hand-Tracking	Meta SDK	Meta Quest only	Open-source
Glycon3D	Multi-framework	Meta, SteamVR, Vive	Commercial ($)
VR Hands Recording (this)	**OpenXR**	**Meta Quest, Pico (hand tracking)**	**Proprietary / Dev Tool**

The system is intended for:

- Creators of animated content for web, virtual reality, and interactive storytelling
- XR game designers and immersive application developers
- Researchers in the field of human-computer interaction
- Educational or artistic teams aiming to prototype realistic hand movements

6. Conclusion

VR Hands Recording provides an accessible, flexible, and cost-effective solution for capturing hand movement animations in virtual reality environments. Built on Unity and the OpenXR framework, the system supports a wide range of headsets, including Meta Quest and Pico devices, offering broad compatibility and future-proof integration.

Its intuitive workflow allows users to quickly record hand gestures, export them in FBX format, and import them into popular 3D graphics tools such as Blender for further editing and rendering. Compared to existing solutions, VR Hands Recording stands out for its:

- Cross-platform compatibility enabled by OpenXR;
- Ease of use through pre-configured Unity scenes;
- Direct FBX export with minimal post-processing;

Free availability, making it suitable for small teams, educational settings, and research environments.

This system is particularly well-suited for creative industries, multimedia production, human-computer interaction studies, and AI-driven XR applications suggesting future directions such as co-creation of therapeutic XR spaces via AI integration [1]. Future developments may include real-time streaming of motion data, integration with full-body avatars, and support for standalone (untethered) workflows [10].

7. Declaration on Generative AI

During the preparation of this work, the author(s) used GPT in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] A. D. Greca, I. Amaro, P. Barra, E. Rosapepe, G. Tortora, Enhancing therapeutic engagement in mental health through virtual reality and generative ai: a co-creation approach to trust building, in: 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2024, pp. 6805–6811. doi:10.1109/BIBM62325.2024.10822177.

[2] S. Gupta, S. Bagga, D. K. Sharma, Hand gesture recognition for human computer interaction and its applications in virtual reality, Advanced Computational Intelligence Techniques for Virtual Reality in Healthcare (2019). URL: https://api.semanticscholar.org/CorpusID:213652695.

[3] G. Buckingham, Hand tracking for immersive virtual reality: Opportunities and challenges, Frontiers in Virtual Reality Volume 2 - 2021 (2021). doi:10.3389/frvir.2021.728461.

[4] M. Casillo, L. Cecere, F. Colace, A. Lorusso, D. Santaniello, C. Valentino, A framework to improve cultural experience through metaverse and recommender systems, in: 2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2025, pp. 582–588. doi:10.1109/VRW66409.2025.00123.

[5] G. B. Mo, J. J. Dudley, P. O. Kristensson, 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, CHI '21, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3411764.3445766.

[6] N. Plant, R. Gibson, C. G. Diaz, B. Martelli, M. Zbyszyński, R. Fiebrink, M. Gillies, C. Hilton, P. Perry, Movement interaction design for immersive media using interactive machine learning, in: Proceedings of the 7th International Conference on Movement and Computing, MOCO '20, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3401956.3404252.

[7] A. Tortora, I. Amaro, A. Della Greca, P. Barra, Exploring the role of generative artificial intelligence in virtual reality: Opportunities and future perspectives, in: J. Y. C. Chen, G. Fragomeni (Eds.), Virtual, Augmented and Mixed Reality, Springer Nature Switzerland, Cham, 2025, pp. 125–142. doi:10.1007/978-3-031-93700-2_9.

[8] S. Aghel Manesh, T. Zhang, Y. Onishi, K. Hara, S. Bateman, J. Li, A. Tang, How people prompt generative ai to create interactive vr scenes, in: Proceedings of the 2024 ACM Designing Interactive Systems Conference, DIS '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 2319–2340. doi:10.1145/3643834.3661547.

[9] L. Rizzo, D. Rossi, M. Giammetti, P. Barra, Simulation system for dental bracket positioning, in: 2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2025, pp. 604–610. doi:10.1109/VRW66409.2025.00126.

[10] P. Barra, M. Giammetti, A. Tortora, A. Della Greca, Redefining interaction in a digital twin laboratory with mixed reality, in: H. Degen, S. Ntoa (Eds.), Artificial Intelligence in HCI, Springer Nature Switzerland, Cham, 2024, pp. 295–307. doi:10.1007/978-3-031-60611-3_21.