

# A New Software Toolset for Recording and Viewing Body Tracking Data

Julian Fietkau

julian.fietkau@unibw.de

University of the Bundeswehr Munich  
Munich, Germany

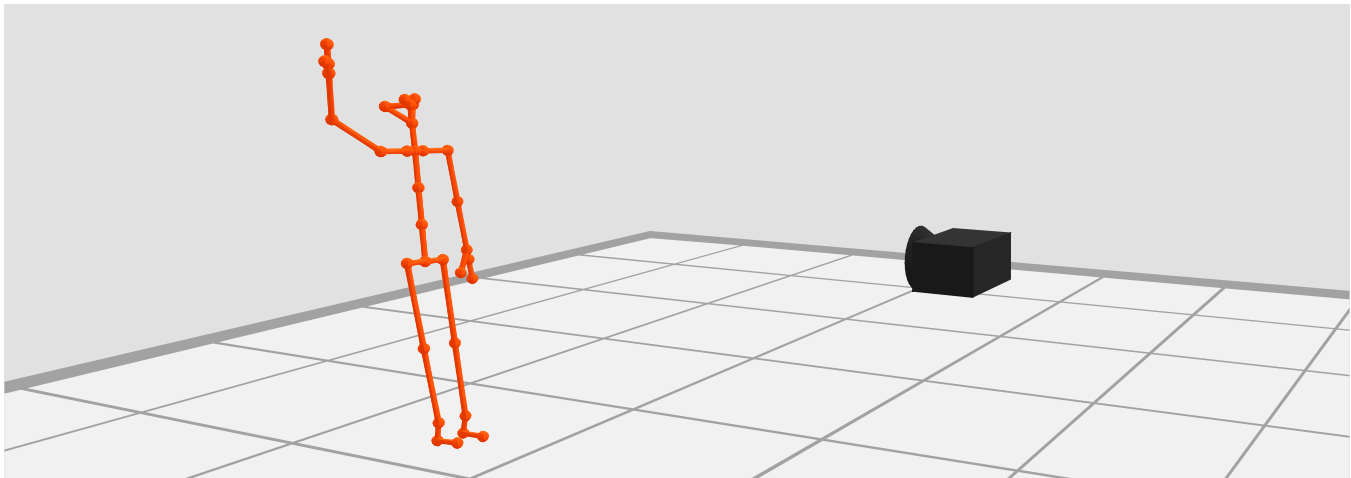


Figure 1: A scene view exported from the *PoseViz* software for visualizing body tracking data, showing a person waving at a 3D sensor.

## ABSTRACT

While 3D body tracking data has been used in empirical HCI studies for many years now, the tools to interact with it tend to be either vendor-specific proprietary monoliths or single-use tools built for one specific experiment and then discarded. In this paper, we present our new toolset for cross-vendor body tracking data recording, storing, and visualization/playback. Our goal is to evolve it into an open data format along with software tools capable of producing and consuming body tracking recordings in said new format, and we hope to find interested collaborators for this endeavour.

## KEYWORDS

body tracking, pose estimation, data visualization, visualization software

## 1 INTRODUCTION

*Body tracking* (also called *pose estimation*) has been in scientific and industrial use for over a decade. It describes processes that extract the position or orientation of people and possibly their individual

limbs out of data from sensors (typically cameras), making it a sub-field of image processing. Due to increasing computational viability and decreasing costs of appropriate sensor hardware spearheaded by early *Kinect* depth camera models, it has enjoyed a modest but consistent relevance in empirical deployment studies in human-computer interaction (e.g. [3, 4]).

We will be using the word *body tracking* to describe the process of extracting the aforementioned spatiotemporal data on people and their movement from sensor hardware such as cameras. *Body tracking data* is the holistic term to describe the resulting data points, generally including the positions of people in the sensor's detection area at a specific point in time, as well as the positions and orientations of their limbs at some specific degree of resolution and precision that depends on the hardware and software setup. In the literature, body tracking data is also called *pose data* [6] or *skeleton data* [5], referring to the relatively coarse resolution of body points represented in the data.

Body tracking data offers some advantages over full video recordings. It is well-suited for gesture or movement analysis, because spatiotemporal coordinates of audience members are readily accessible. It also offers anonymity and can be stored and analyzed without needing to account for personally identifiable information.

We most commonly see body tracking data collected, analyzed, and made use of in real time, i.e., the system detects a person, executes some kind of reaction (possibly interactive and visible to the person, like in interactive exhibits or body tracking games, or possibly silent and unnoticeable, like in crowd tracking cameras

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Veröffentlicht durch die Gesellschaft für Informatik e.V. in P. Fröhlich & V. Cobus (Hrsg.): *Mensch und Computer 2023 – Workshopband*, 03.-06. September 2023, Rapperswil (SG)

© 2023 Copyright held by the owner/author(s).  
<https://doi.org/10.18420/muc2023-mci-ws13-334>

that count passers-by), and then immediately discards the full body tracking data. This is practical for most experiments, but it also has disadvantages:

- It makes it difficult to compare different sensor hardware or software setups to determine which one is best suited for a specific spatial context without implementing the full data analysis.
- The process to go from sensor data to system reaction is relatively opaque without interim visualizations. It is not generally possible to mentally visualize body tracking data based on its component 3D coordinates, so some kind of real-time pose visualization must additionally be implemented for testing and debugging purposes.

While a few implementations for recording and storing the full body tracking data exist (see section 3), they are vendor-specific and their storage formats are not open.

With this paper, we advocate for a novel open format for recording, storing, and replaying body tracking data. We describe the file/stream format we have developed, showcase our prototypical recording software capable of recording body tracking data from several different cameras and tool suites, and present the *PoseViz* visualization software, a visualization and playback tool for stored or real-time streamed body tracking data.

## 2 TOOLSET OVERVIEW

To have practical use, a body tracking tool suite has to include at least a way to record and store body tracking data, and a way to visualize or play back previously recorded data. In order to facilitate these two functions, it also requires a data storage format that both portions of the system have agreed on. This article outlines how we provide all three components. See Figure 2 for a visual overview of how the toolset components interact.

First off, section 3 presents our body tracking data storage format, which has been designed with ease of programmatic interaction in mind. In section 4 we describe our *Tracking Server*, the software that integrates various sensor APIs and converts captured body tracking data into the PoseViz file format. Section 5 shows the PoseViz software, which can play back stored (or live-streamed) body tracking data.

## 3 THE POSEVIZ FILE FORMAT

We begin with a brief look at the landscape of existing file formats. As of this publication, there are no other vendor-neutral storage formats for body tracking data. In its *Kinect Studio* software, Microsoft provides a facility to record body tracking data using Kinect sensors, but the format is proprietary and can only be played back using Kinect Studio on Windows operating systems. The OpenNI project as well as Stereolabs (the vendor for the ZED series of depth sensors) both have their own recording file formats, but they are geared towards full RGB video data, not body tracking data. There are specialized motion capture formats, *Biovision Hierarchy* being a popular example, but their format specifications are not open and their underlying assumptions on technical aspects like frame rate consistency do not necessarily translate to the body tracking data storage use case.

To solve this issue, we have developed a new file format for body tracking data with the following quality criteria in mind:

- It needs to be vendor-neutral with support for a variety of different sensors, tracking APIs and body models.
- It should be simple to read and parse programmatically. This way, implementers have an easy start even if there is no existing parser in their language of choice.
- It must support rich metadata and context information to allow body tracking recordings to be stored alongside contextual data on its spatial surroundings, on the hardware and software setup that was used, etc.
- It needs to be extensible and annotatable to allow (a) usage with a variety of different sensors and their data output, and (b) post-hoc enrichment and annotation of frame data with added information derived from postprocessing or external information sources.

Our resulting design for a file format is closely inspired by the *Wavefront OBJ* format for 3D vertex data. It is a text-based format that can be viewed and generally understood in a simple text editor. See Figure 3 for an example of what a PoseViz body tracking recording looks like.

It contains a header section that supports metadata in a standardized format. The body of the file consists of a sequence of timestamped *frames*, each signifying a moment in time. Each frame can contain one or more *person* records. Each person record has a mandatory ID (intended for following the same person across frames within one recording) and X/Y/Z position. Optionally, the person record can contain whatever data the sensor provides, most commonly including a numbered list of key body points with their X/Y/Z coordinates. The frame and person record may also be extended with additional fields derived from post-hoc data interpretation and enrichment. As an example, the ZED 2 sensor does not provide an *engagement* value like the Kinect does, but a similar value could be calculated per frame based on the key point data and reintegrated into the recording for later visualization [1].

The PoseViz file format has been adjusted and evolved over the year that we have been recording body tracking data in our deployment setup [2]. It is expected to evolve further as compatibility for more sensors, body models and use cases is added. We seek collaborators from the research community who would be interested in co-steering this process.

## 4 RECORDING AND STORING BODY TRACKING DATA

With the file format chosen, there needs to be a software that accesses sensor data in real time and converts it into PoseViz data files. In our toolset, this function is performed by the *Tracking Server*, named for its purpose to provide tracking data to consuming applications. Its current implementation is a Python program that can interface with various sensor APIs. It serves two central use cases:

- (1) Manual recording: start and stop a body tracking recording via button presses, save the result to a file. This mode is intended for supervised laboratory experiments.
- (2) Automatic recording: start a recording every time a person enters the field of view of the sensor and stop when the

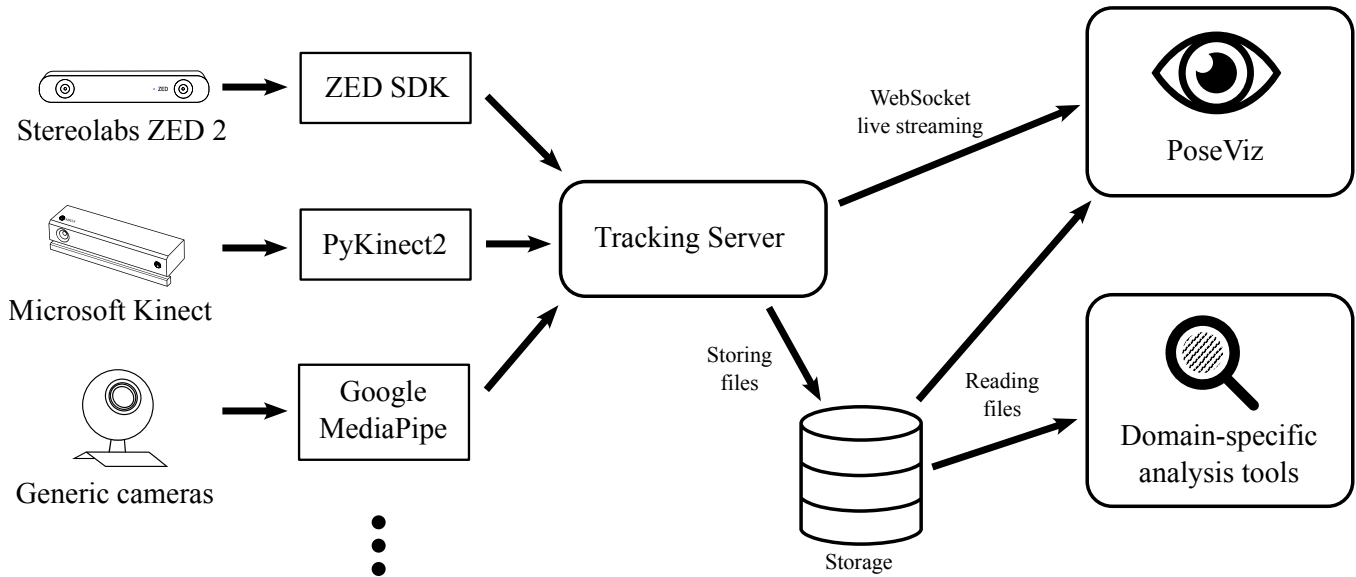


Figure 2: Toolset overview showing the interactions between system components.

```

ts 2022-11-24T11:39:18.796
ct 0.18679 0.34504 0.12434
co -0.30539 -0.03130 -0.01309 0.95162
f 0
p 0 -0.19495 1.34052 3.32204
cf 0.76
ast IDLE
gro -0.01758 0.97214 0.01931 -0.23298
v 0.00000 0.00000 0.00000
k 0 -0.18878 1.29331 3.28806
k 1 -0.18537 1.15794 3.28186
k 2 -0.18196 1.02257 3.27566
k 3 -0.17897 0.88718 3.26986
k 4 -0.14888 0.88851 3.25432
k 5 -0.03154 0.89370 3.19375
(...)
    
```

Figure 3: Excerpt from a PoseViz body tracking data file showing the beginning of a recording including a timestamp, camera translation and camera rotation, followed by the first frame (time offset zero) containing one person. Partial body tracking data is shown here including the tracking confidence, action state, global root orientation, current velocity, and the first few body key points.

last person leaves it. Each recording gets stored as a separate time-stamped file. This mode is intended for long-term deployments.

The Tracking Server is capable of persisting its recordings to the file system for later asynchronous access, or it can provide a *WebSocket* stream to which a PoseViz client can connect across a local network or the internet to view body tracking sensor data in real time. As for sensor interfaces, it can currently fetch body

tracking data from Stereolabs ZED 2 and ZED 2i cameras via the ZED SDK (other models from the same vendor are untested) or from generic video camera feeds using Google’s *MediaPipe* framework and its *BlazePose* component. An interface for Kinect sensors using *PyKinect2* is in the process of being developed.

In our current deployment setup, we have the Tracking Server running in automatic recording mode as a background process. Between July 2022 and June 2023, it has generated approximately 40 GB worth of body tracking recordings across our two semi-public ZED 2 sensor deployments at University of the Bundeswehr Munich.

The Tracking Server is not yet publically released.

## 5 VISUALIZATION AND PLAYBACK IN POSEVIZ

During our first experiments with capturing body tracking data, we noticed very quickly that the capturing process cannot be meaningfully evaluated without a corresponding visualization component to check recorded data for plausibility. The *PoseViz* software (not affiliated with the Python module of the same title by István Sáráncsi) is the result of extending our body tracking visualization prototype into a relatively full-featured visualization tool that gives access to a variety of useful visualizations.

We planned PoseViz as a platform-neutral tool, intended to run on all relevant desktop operating systems and preferably also on mobile devices. The modern web platform offers enough rendering capabilities to make this feasible. Consequently, PoseViz was implemented as a JavaScript application with 3D rendering code using the *three.js* library. The software runs entirely client-side and requires no server component except for static file delivery.

On account of being designed to replay body tracking recordings, the PoseViz graphical user interface is based on video player applications with a combined play/pause button, a progress bar

showing the timeline of the current file, and a timestamp showing the current position on the timeline as well as the total duration (see Figure 4). The file can be played at its actual speed using the play button, or it can be skimmed by dragging the progress indicator.

PoseViz can be used to open previously recorded PoseViz files, or it can open a WebSocket stream provided by a Tracking Server to view real-time body tracking data. The current viewport can be exported as a PNG or SVG file at any time.

Users can individually enable or disable several render components, including joints (body key points), bones (connections between joints), each person's overall position as a pin (with or without rotation), the sensor at its true position as well as its field of view (provided it is known), as well as 2D walking trajectories and estimated gaze directions. We are working on a feature to display a 3D model of the spatial context of a specific sensor deployment.

The default camera is a free 3D view that can be rotated around the sensor position. In addition, the camera can be switched to the sensor view (position and orientation fixed to what the sensor could perceive) or to one of three orthographic 2D projections.

These capabilities are geared towards initial explorations of body tracking data. Researchers can use this tool to check their recordings for quality, identify sensor weaknesses, or look through recordings for interesting moments.

For most research questions surrounding body tracking, more specific analysis tools will need to be developed to inquire about specific points of interests. For example, if a specific gesture needs to be identified or statistical measures are to be taken across a number of recordings, this is outside the scope of PoseViz and a bespoke analysis process is needed. However, post-processed data may be added to PoseViz files and visualized in the PoseViz viewport – for example, we have done this for post-hoc interpreted engagement estimations (displayed through color shifts in PoseViz).

PoseViz can be used in any modern web browser.<sup>1</sup>

<sup>1</sup><https://poseviz.com/>

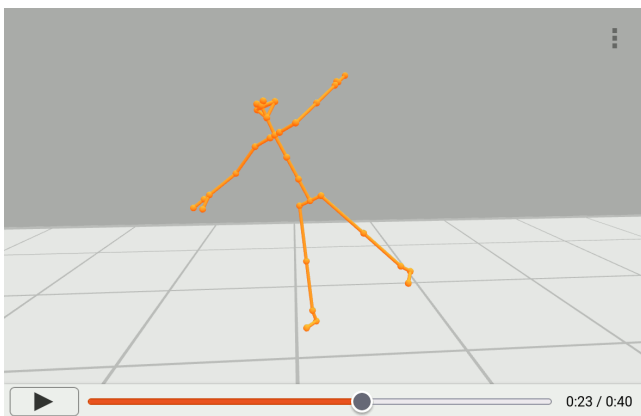


Figure 4: Screenshot of the PoseViz playback software, showing an example body tracking data recording as well as the player UI at the bottom of the screen (play button, progress bar, time stamp) and the settings menu signified by the three dots in the upper right.

## 6 CONCLUSION

In this article we have described the PoseViz file format for body tracking data as well as our Tracking Server for recording body tracking events and the PoseViz visualization software for playing recorded body tracking data. Each of these components can only be tested in conjunction with one another, which is why they have to evolve side by side.

This toolset is currently in use for the *HoPE* project (see Acknowledgements) and is seeing continued improvement in this context. We feel that it has reached a stage of maturity where external collaborators could feasibly make use of it in their own research contexts. It is still far from being a commercial-level drop-in solution, but making use of this infrastructure (and contributing to its development) may save substantial resources compared to implementing a full custom toolset. Potential collaborators are advised to contact the author.

The intended next step for the toolset is an expert evaluation. Researchers who have previously worked with body tracking data will be interviewed about their needs for visualization tools, and they will have an opportunity to test the current version of PoseViz and offer feedback for future improvements.

## ACKNOWLEDGEMENTS

Thank you to Jan Schwarzer, Tobias Plischke, James Beutler, and Maximilian Römpler for their feedback and contributions regarding PoseViz and body tracking data recording in general.

This research project, titled “Investigation of the honeypot effect on (semi-)public interactive ambient displays in long-term field studies,” is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 451069094.

## REFERENCES

- [1] Coleen Cabalo, Lars Gatzemeyer, and Lukas Mathes. 2023. Evaluating the engagement of users from public displays. In *Mensch und Computer 2023 – Workshopband*, Peter Fröhlich and Vanessa Cobus (Eds.). Gesellschaft für Informatik e.V., Bonn, Germany, 7 pages. <https://doi.org/10.18420/muc2023-mci-ws13-282>
- [2] Michael Koch, Julian Fietkau, and Laura Stojko. 2023. Setting up a long-term evaluation environment for interactive semi-public information displays. In *Mensch und Computer 2023 – Workshopband*, Peter Fröhlich and Vanessa Cobus (Eds.). Gesellschaft für Informatik e.V., Bonn, Germany, 5 pages. <https://doi.org/10.18420/muc2023-mci-ws13-356>
- [3] Ville Mäkelä, Tomi Heimonen, and Markku Turunen. 2018. Semi-Automated, Large-Scale Evaluation of Public Displays. *International Journal of Human-Computer Interaction* 34, 6 (2018), 491–505. <https://doi.org/10.1080/10447318.2017.1367905>
- [4] Jan Schwarzer, Susanne Draheim, and Kai von Luck. 2022. Spatial and Temporal Audience Behavior of Scrum Practitioners Around Semi-Public Ambient Displays. *International Journal of Human-Computer Interaction* (2022), 19 pages. <https://doi.org/10.1080/10447318.2022.2099238>
- [5] Sijie Song, Culling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2017. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. *Proceedings of the AAAI Conference on Artificial Intelligence* 31, 1 (2017), 4263–4270. <https://doi.org/10.1609/aaai.v31i1.11212>
- [6] Kathan Vyas, Le Jiang, Shuangjun Liu, and Sarah Ostadabbas. 2021. An Efficient 3D Synthetic Model Generation Pipeline for Human Pose Data Augmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 1542–1552. <https://doi.org/10.1109/CVPRW53098.2021.00170>