The Journal of Open Source Software

Caliscope: GUI Based Multicamera Calibration and Motion Tracking

Donald Prible[1]

1 The University of Texas at Austin, United States

DOI: 10.21105/joss.07155

Software

- Review
- Repository
- Archive

Editor: Beatriz Costa Gomes

Reviewers:

- @Timozen
- @davidpagnon

Submitted: 19 August 2024
Published: 15 October 2024

Summary

3D motion capture is an indispensable tool employed in multiple research domains such as rehabilitation, sports science, and robotics. This technique necessitates a thorough calibration of each camera's intrinsic properties, the alignment of all cameras in a shared spatial context, and accurate 2D point tracking. With these calculations, it is possible to triangulate landmark locations and reconstruct movement dynamics. Historically, this process has relied on costly proprietary tools, though the emergence of open-source pose estimation tools and the availability of high-quality consumer-grade cameras have paved the way for innovative approaches.

Caliscope is a Python package designed to automate the process of camera calibration and landmark triangulation using pre-recorded video, enabling a range of camera hardware setups and tracking solutions to be employed. User interaction with the software is facilitated through a graphical user interface (GUI), which provides visual feedback to assist in troubleshooting and quality control.

Statement of need

Commercially available motion capture systems composed of proprietary hardware and software can be prohibitively expensive. A 10-camera Vicon system, for example, can cost approximately $50,000 (Cinematography Database, 2021). This presents a hurdle for researchers, particularly those who study clinical populations with movement disorders that make travel to a research lab challenging. Transporting data collection tools to where these clinical populations already are could expand the pool of potential research participants, but presents risks and limitations given the cost of the equipment. Rehabilitation researchers would benefit from an open-source tool that decouples data acquisition from data processing in motion capture systems. This enables the development of lower cost hardware configurations that may be more easily deployed to multiple data collection sites, expanding the scale and ease of data collection.

The emergence of machine learning tools for automated 2D landmark tracking has facilitated a variety of open-source motion capture projects. DeepLabCut (Mathis et al., 2018), for example, allows for the rapid training of custom pose estimation models, applicable to a range of animal subjects, including humans. It offers 3D stereotriangulation using two cameras.

Expanding beyond two cameras enhances landmark localization and mitigates landmark occlusion but adds significant complexity to the calibration process. While OpenCV (Bradsky, 2000) enables straightforward single and stereocamera calibration, calibrating more than two cameras requires the use of bundle adjustment (Triggs et al., 2000), leveraging optimization tools such as SciPy (Mayorov, 2015; Virtanen et al., 2020).

Anipose (Karashchuk et al., 2021), which integrates with DeepLabCut, automates the bundle adjustment process for multicamera calibration as well as the triangulation of 3D landmark positions from more than 2 views. To facilitate the calibration process, Anipose simplifies the underlying camera intrinsic parameters, using one parameter each for the pinhole camera and distortion. A process of iterative bundle adjustment is used to remove the impact of outlier results in the calibration.

Caliscope carries forward the triangulation code of Anipose, but provides dedicated estimation of camera intrinsics according to the standard camera model employed by OpenCV (Bradsky, 2000). It therefore estimates 4 parameters for the pinhole camera model, and 5 parameters for the lens distortion. This is in contrast to the 2 parameters estimated by Anipose (focal length and a single distortion paramters). With more accurate and precise intrinsic parameters, the bundle adjustment computes quickly without the need for an iterative approach.

FreeMocap (Cherian et al., 2024) is another notable open-source motion capture project. It uses the original calibration process developed by Anipose. Unlike Anipose, FreeMocap uses Google's Mediapipe (Lugaresi et al., 2019) for pose tracking, which avoids the more complicated setup required by DeepLabCut. Caliscope also uses this approach for ease of execution, though defines the `Tracker` as an abstract base class that is implemented with multiple versions of Mediapipe tracking (i.e. pose, hands, and holistic). This flexible framework is intededed to facilitate the integration of additional landmark tracking tools in the future, such as MMPose (MMPose Contributors, 2020).

Pose2Sim (Pagnon et al., 2022) is another tool that merges both camera calibration and markerless pose estimation. Similar to Caliscope, Pose2Sim employs a camera model composed of 4 pinhole camera parameters and 5 distortion parameters. Additionally, both projects export their output to the `.trc` file format to facilitate integration with the biomechanical modelling software OpenSim. While Pose2Sim does not perform bundle adjustment to refine the extrinsic camera estimates, it does provide a number of features that are valuable components of a motion tracking workflow which are not present in Caliscope and would be useful in future motion tracking tools. These include the ability to calibrate camera extrinsics based on scene elements, the capacity to distinguish between multiple subjects in view at once, and more sophisticated triangulation methods that can incorporate the confidence of a model's prediction of a pose landmark.

Acknowledgements

The author would like to extend gratitude to Lili Karushcheck, PhD whose project Anipose provided the initial inspiration for Caliscope as well as to Jon Matthis, PhD who is the lead developer of FreeMocap. Dr. Matthis provided valuable early-stage guidance related to general code considerations and open-source project management. Additional thanks are due to Ryan Govostes for coding contributions and feedback related to the intrinsic calibration GUI development.

References

Bradsky, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, *120*, 122–125.

Cherian, A., Queen, P., Trent, W., Endurance, I., & Matthis, J. S. (2024). *FreeMoCap: A free, open source markerless motion capture system*. https://doi.org/10.5281/zenodo.7233714

Cinematography Database. (2021). *Indie Vicon System Cost and Solo Operator Guide*. https://www.youtube.com/watch?v=YgT6XY6ldj8

Karashchuk, L., Rupp, K. L., Dickinson, E. S., Walling-Bell, S., Sanders, E., Azim, E., Brunton, B. W., & Tuthill, J. C. (2021). Anipose: A toolkit for robust markerless 3D pose estimation. *Cell Reports*, *36*(13). https://doi.org/10.1016/j.celrep.2021.109730

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). *MediaPipe: A Framework for Building Perception Pipelines*. https://doi.org/10.48550/arXiv.1906.08172

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, *21*(9), 1281–1289. https://doi.org/10.1038/s41593-018-0209-y

Mayorov, N. (2015). Large-scale bundle adjustment in scipy. In *SciPy Cookbook*. https://scipy-cookbook.readthedocs.io/items/bundle_adjustment.html

MMPose Contributors. (2020). *OpenMMLab Pose Estimation Toolbox and Benchmark*.

Pagnon, D., Domalain, M., & Reveret, L. (2022). Pose2Sim: An open-source Python package for multiview markerless kinematics. *Journal of Open Source Software*, *7*(77), 4362. https://doi.org/10.21105/joss.04362

Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle Adjustment — A Modern Synthesis. In B. Triggs, A. Zisserman, & R. Szeliski (Eds.), *Vision Algorithms: Theory and Practice* (pp. 298–372). Springer. https://doi.org/10.1007/3-540-44480-7_21

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Van Der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2