The Journal of Open Source Software

Sports2D: Compute 2D human pose and angles from a video or a webcam

David Pagnon [1¶] **and HunMin Kim** [2]

1 Centre for the Analysis of Motion, Entertainment Research & Applications (CAMERA), University of Bath, Claverton Down, Bath, BA2 7AY, United Kingdom **2** Inha University, Yonghyeon Campus, 100 Inha-ro, Michuhol-gu, Incheon 22212, South Korea ¶ Corresponding author

DOI: 10.21105/joss.06849

Software

- Review
- Repository
- Archive

Editor: Kevin M. Moerman

Reviewers:

- @tuliofalmeida
- @nicos1993

Submitted: 17 February 2024
Published: 24 September 2024

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

`Sports2D` provides a user-friendly solution for automatic and real-time analysis of multi-person human movement from a video or a webcam. This Python package uses 2D markerless pose estimation to detect joint coordinates from videos, and then computes 2D joint and segment angles.

The output incorporates annotated videos and image sequences overlaid with joint locations, joint angles, and segment angles, for each of the detected persons. For further analysis, this information is also stored in files that are editable with MS Excel® or any other spreadsheet editor (.trc for locations, .mot for angles, according to the OpenSim standard (Delp et al., 2007; Seth, 2018)).

`Sports2D` may be useful for clinicians as a decision supports system (CDSS) (Bright et al., 2012), as well as for gait analysis (Whittle, 2014) or ergonomic design (Patrizi et al., 2016). Sports coaches can also use it to quantify key performance indicators (KPIs) (O'Donoghue, 2008; Pagnon, Domalain, Robert, et al., 2022), or to better understand, correct, or compare athletes' movement patterns. Finally, it can be used by researchers as a simple tool for 2D biomechanical analysis on the fly. One of the multiple use cases would be to evaluate ACL injury risks from deceleration drills (Di Paolo et al., 2021).

Statement of need

Machine learning has recently accelerated the development and availability of markerless kinematics (Colyer et al., 2018; Zheng et al., 2023), which allows for the collection of kinematic data without the use of physical markers or manual annotation.

A large part of these tools focus on 2D analysis, such as `OpenPose` (Cao et al., 2019), `BlazePose` (Bazarevsky et al., 2020), or `DeepLabCut` (Mathis et al., 2018). More recently, `RTMPose` (Jiang et al., 2023) offered a faster, more accurate, and more flexible alternative to the previous solutions. Still, although they bear the advantage of being open-source, none of these options are easily accessible to people who do not have a programming background, and the output is not directly usable for further kinematic investigation. Yet, clinical acceptance of new technologies is known to be influenced not only by their price value and their performance, but also by their perceived ease-of-use, the social influence around the customer, and other parameters described by the Unified Theory of Acceptance and Use of Technology (UTAUT2) (Venkatesh et al., 2012).

Figure 1: Example results from a demonstration video.

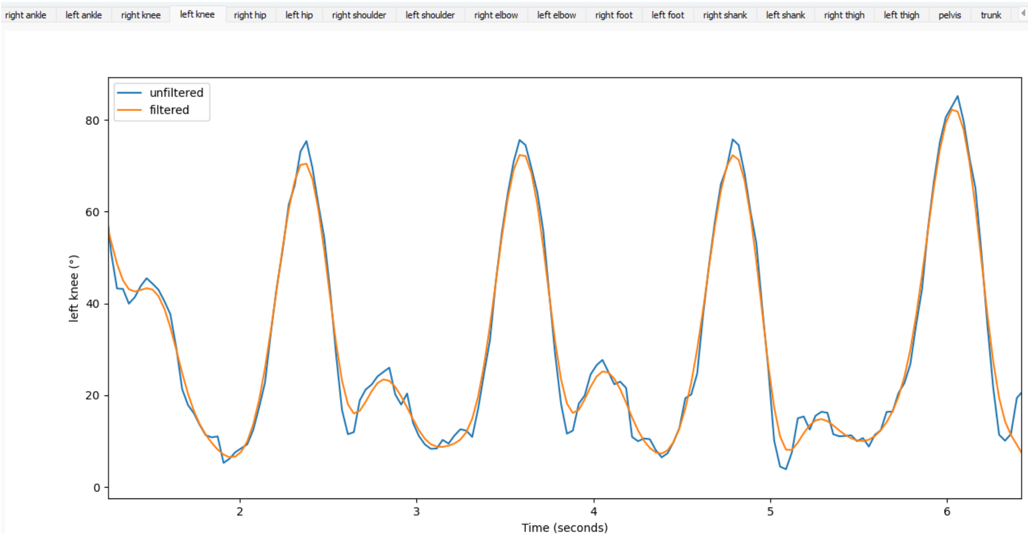

Figure 2: Example joint angle output.

In fact, there is a clear trade-off between accuracy and ease-of-use. Some open-source tools focus on the accuracy of a 3D analysis by using multiple cameras, such as Pose2Sim (Pagnon, Domalain, & Reveret, 2022) or OpenCap (Uhlrich et al., 2022). These, however, require either a certain level of programming skills, a particular hardware setup, or to send data to a server that does not comply with the European rules of data protection (GDPR). Some other tools choose to put more emphasis on user-friendliness, and point out that 2D analysis is often sufficient when the analyzed motion mostly lies in the sagittal or frontal plane. Sit2Stand (Boswell et al., 2023) and CP GaitLab (Kidziński et al., 2020) provide such tools, although they are focused on very specific tasks. Kinovea (Kinovea), on the other hand, is a widely used software for sports performance analysis, which provides multiple additional features. However, it relies on tracking manual labels. This can be time-consuming when analyzing numerous videos, and it may also be lacking robustness when the tracked points are lost. It is also only available on Windows, and requires the user to transfer files prior to analysis.

`Sports2D` is an alternative solution that aims at filling this gap: it is free and open-source, straightforward to install and to run, can be run on any platform, can be run locally for data protection, and it automatically provides 2D joint and segment angles without the need for manual annotation. It is also robust and flexible, works in real-time and can be used to process one video, several videos at once, or webcam stream. Multi-person analysis is available, and the output is directly usable for further statistical analysis.

Workflow

Installation and usage

`Sports2d` is installed under Python via `pip install sports2d`. If a valid CUDA installation is found, Sports2D uses the GPU, otherwise it uses the CPU with OpenVino acceleration.

A detailed installation and usage guide can be found on the repository: https://github.com/david-pagnon/Sports2D.

Sports2D method details

Sports2D:

1. Reads stream from a webcam, from one video, or from a list of videos. It selects an optional specified time range to process.
2. Sets up the RTMLib pose tracker with specified parameters. It can be run in lightweight, balanced, or performance mode, and for faster inference, keypoints can be tracked for a certain number of frames instead of detected. Any RTMPose model can be used.
3. Tracks people so that their IDs are consistent across frames. A person is associated to another in the next frame when they are at a small distance. IDs remain consistent even if the person disappears for a few frames. This carefully crafted `sports2d` tracker runs at a comparable speed as the RTMlib one but is much more robust. The user can still choose the RTMLib method if they need it by using the `tracking_mode` argument.
4. Retrieves the keypoints with high enough confidence, and only keeps the persons with enough average high-confidence.
5. Computes the selected joint and segment angles, and flips them on the left/right side if the respective foot is pointing to the left/right. The user can select which angles they want to compute, display, and save.
6. Draws bounding boxes around each person and writes their IDs
   Draws the skeleton and the keypoints, with a green to red color scale to account for their confidence
   Draws joint and segment angles on the body, and writes the values either near the joint/segment, or on the upper-left of the image with a progress bar
7. Interpolates missing pose and angle sequences if gaps are not too large. Filters them with the selected filter (among `Butterworth`, `Gaussian`, `LOESS`, or `Median`) and their parameters
8. Optionally shows processed images, saves them, or saves them as a video
   Optionally plots pose and angle data before and after processing for comparison
   Optionally saves poses for each person as a TRC file, and angles as a MOT file

**The Demo video that Sports2D is tested on is voluntarily challenging, in order to demonstrate the robustness of the process after sorting, interpolation and filtering. It contains:

- One person walking in the sagittal plane
- One person in the frontal plane. This person then performs a flip while being backlit, both of which are challenging for the pose detection algorithm
- One tiny person flickering in the background who needs to be ignored

Joint and segment angle estimation:

Specific joint and segment angles can be chosen. They are consistent regardless of the direction the participant is facing: the participant is considered to look to the left when their toes are to the left of their heels, and to the right otherwise. Resulting angles can be filtered in the same way as point coordinates, and they can also be plotted.

Joint angle conventions are as follows (Figure 3):

- Ankle dorsiflexion: Between heel and big toe, and ankle and knee.
  -90° when the foot is aligned with the shank.
- Knee flexion: Between hip, knee, and ankle.
  0° when the shank is aligned with the thigh.
- Hip flexion: Between knee, hip, and shoulder.
  0° when the trunk is aligned with the thigh.
- Shoulder flexion: Between hip, shoulder, and elbow.
  180° when the arm is aligned with the trunk.
- Elbow flexion: Between wrist, elbow, and shoulder.
  0° when the forearm is aligned with the arm.

Segment angles are measured anticlockwise between the horizontal and the segment lines:

- Foot: Between heel and big toe.
- Shank: Between knee and ankle.
- Thigh: Between hip and knee.
- Pelvis: Between left and right hip
- Trunk: Between hip midpoint and shoulder midpoint
- Shoulders: Between left and right shoulder
- Head: Between neck and top of the head
- Arm: Between shoulder and elbow.
- Forearm: Between elbow and wrist.

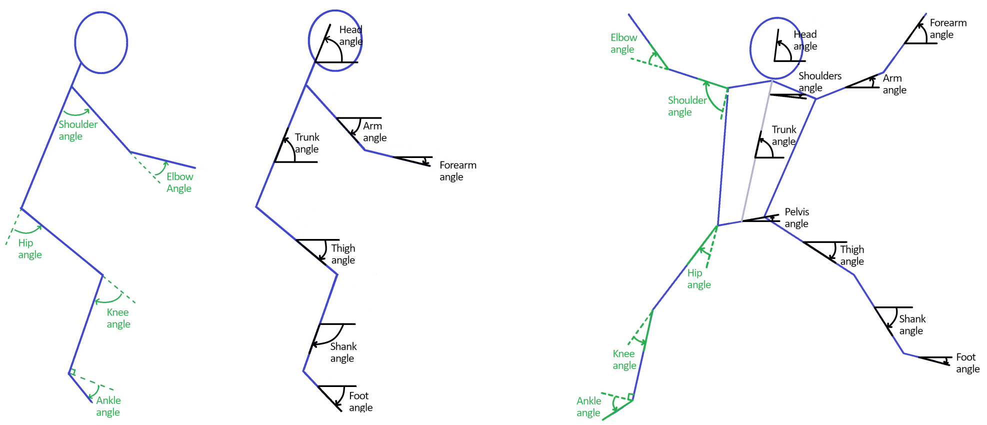

Figure 3: Joint angle conventions

Limitations

The user of `Sports2D` should be aware of the following limitations:

- Results are acceptable only if the participants move in the 2D plane, either in the frontal plane or in the sagittal one. If you need research-grade markerless joint kinematics, consider using several cameras, and constraining angles to a biomechanically accurate model. See `Pose2Sim` (Pagnon, Domalain, & Reveret, 2022) for example.
- Angle estimation is only as good as the pose estimation algorithm, i.e., it is not perfect (Wade et al., 2022), especially if motion blur is significant such as on some broadcast

videos.

Acknowledgements

I would like to acknowledge Rob Olivar, a sports coach who enlightened me about the need for such a tool.
I also acknowledge the work of the dedicated people involved in the many major open-source software programs and packages used by `Sports2D`, such as `Python`, `RTMPPose`, `OpenCV` (Bradski, 2000), among others.

References

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking. *arXiv Preprint arXiv:2006.10204*. https://doi.org/10.48550/arXiv.2006.10204

Boswell, M. A., Kidziński, Ł., Hicks, J. L., Uhlrich, S. D., Falisse, A., & Delp, S. L. (2023). Smartphone videos of the sit-to-stand test predict osteoarthritis and health outcomes in a nationwide study. *Npj Digital Medicine*, *6*(1), 32. https://doi.org/10.1038/s41746-023-00775-1

Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal of Software Tools*.

Bright, T. J., Wong, A., Dhurjati, R., Bristow, E., Bastian, L., Coeytaux, R. R., Samsa, G., Hasselblad, V., Williams, J. W., Musty, M. D., & others. (2012). Effect of clinical decision-support systems: A systematic review. *Annals of Internal Medicine*, *157*(1), 29–43. https://doi.org/10.7326/0003-4819-157-1-201207030-00450

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., & Sheikh, Y. (2019). OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(1), 172–186. https://doi.org/10.1109/TPAMI.2019.2929257

Colyer, S. L., Evans, M., Cosker, D. P., & Salo, A. I. (2018). A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system. *Sports Medicine-Open*, *4*(1), 1–15. https://doi.org/10.1186/s40798-018-0139-y

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., & Thelen, D. G. (2007). OpenSim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, *54*(11), 1940–1950. https://doi.org/10.1109/TBME.2007.901024

Di Paolo, S., Zaffagnini, S., Tosarelli, F., Aggio, F., Bragonzoni, L., Grassi, A., & Della Villa, F. (2021). A 2D qualitative movement assessment of a deceleration task detects football players with high knee joint loading. *Knee Surgery, Sports Traumatology, Arthroscopy*, *29*, 4032–4040. https://doi.org/10.1007/s00167-021-06709-2

Jiang, T., Lu, P., Zhang, L., Ma, N., Han, R., Lyu, C., Li, Y., & Chen, K. (2023). RTMPose: Real-time multi-person pose estimation based on MMPose. *arXiv*. https://doi.org/10.48550/arXiv.2303.07399

Kidziński, Ł., Yang, B., Hicks, J. L., Rajagopal, A., Delp, S. L., & Schwartz, M. H. (2020). Deep neural networks enable quantitative movement analysis using single-camera videos. *Nature Communications*, *11*(1), 4054. https://doi.org/10.1038/s41467-020-17807-z

Kinovea. *Kinovea - a microscope for your videos*. https://www.kinovea.org/features.html; GitHub. https://www.kinovea.org/features.html

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

O'Donoghue, P. (2008). Principal components analysis in the selection of key performance indicators in sport. *International Journal of Performance Analysis in Sport*, *8*(3), 145–155. https://doi.org/10.1080/24748668.2008.11868456

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

Pagnon, D., Domalain, M., Robert, T., Lahkar, B.-K., Moussa, I., Saulière, G., Goyallon, T., & Reveret, L. (2022). A 3D markerless protocol with action cameras – Key performance indicators in boxing. *2022 Congress of the European College of Sport Science (ECSS)*. https://hal.archives-ouvertes.fr/hal-03790926

Patrizi, A., Pennestrì, E., & Valentini, P. P. (2016). Comparison between low-cost marker-less and high-end marker-based motion capture systems for the computer-aided assessment of working ergonomics. *Ergonomics*, *59*(1), 155–162. https://doi.org/10.1080/00140139.2015.1057238

Seth, J. L. A. U., Ajay AND Hicks. (2018). OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLOS Computational Biology*, *14*(7), 1–20. https://doi.org/10.1371/journal.pcbi.1006223

Uhlrich, S. D., Falisse, A., Kidziński, Ł., Muccini, J., Ko, M., Chaudhari, A. S., Hicks, J. L., & Delp, S. L. (2022). *OpenCap: 3D human movement dynamics from smartphone videos*. 2022.07.07.499061. https://doi.org/10.1101/2022.07.07.499061

Venkatesh, V., Thong, J. Y., & Xu, X. (2012). Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *MIS Quarterly*, 157–178. https://doi.org/10.2307/41410412

Wade, L., Needham, L., McGuigan, P., & Bilzon, J. (2022). Applications and limitations of current markerless motion capture methods for clinical gait biomechanics. *PeerJ*, *10*, e12995. https://doi.org/10.7717/peerj.12995

Whittle, M. W. (2014). *Gait analysis: An introduction*. Butterworth-Heinemann.

Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., Kehtarnavaz, N., & Shah, M. (2023). Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, *56*(1), 1–37. https://doi.org/10.1145/3603618