THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêtée ministériel : 25 mai 2016

Présentée par

David PAGNON

Thèse dirigée par **Lionel REVERET**
et codirigéee par **Mathieu DOMALAIN**

préparée au sein du **Laboratoire Jean Kuntzmann**
dans **l'École Doctorale l'École Doctorale Mathématiques, Sciences et technologies de l'Information, Informatique**

"Design, evaluation, and application of a workflow for biomechanically consistent markerless kinematics in sports"

"Conception, évaluation, et application d'une méthode biomécaniquement cohérente de cinématique sans marqueurs en sport"

Thèse soutenue publiquement le **"Date de soutenance"**,
devant le jury composé de :

Président
Laboratoire, Président

Rapporteur
Laboratoire, Rapporteur

Examinateur
Laboratoire, Examinateur

Lionel REVERET
INRIA Grenoble, Directeur de thèse

Mathieu DOMALAIN
Institut Pprime, Co-Encadrant de thèse

Invité
Laboratoire, Invité

"To all of you who care about more important stuff than what follows."

Acknowledgements

Should I start this by declaring that these PhD years have been alternatively depressing and engaging, exhausting and stimulating, infuriating and enthralling? This is trite, and true for everyone, PhD student or not. Covid pandemic or not. Child birth or not. Struggles in close friends' and relatives' lifes or not. But there it is. Now that it is stated, let me go straight to my acknowledgements.

Above anyone else, I want to thank my mother. She not only had to deal with the difficult task of raising me and putting up with my constant flow of questions, but also with welcoming the four smaller sisters that came after me. As a widow. With debts to pay off, and very little money coming in. Moving every two years, until we settled in for a small appartment in a neighborhood that some would call a ghetto, although we prefered calling it home. And yet, there was always food on the table. Even better, we had no idea how poor we were, because she literally sacrificed her life for ours, and her passions for our interests. This is quintescential Christlike love. We all had the incredible opportunity of doing at least one physical, and one artistic activity, on top of pursuing university level studies. We also learned how to live happily with very little, which I'm starting to realize is a sort of superpower. Most importantly, she made children that all love each other. Now that I'm a father too, I can measure how high she set the bar, and I can only hope to be half as good as her. I can't award her the Legion of Honor she deserves, but at least here is a little bit of recognition! Thank you from all of us, maman.

I also have a deep thought for my father, who tragically passed away when I was still a little child. He did have to struggle with some issues that would eventually cause his death, but I believe he fought until the very end. He is actually the one who taught me a nice lesson of persistence, surely without even trying. A friend and I were racing up a hill, while my father timed us. I lost. We raced again, I lost again. I tried more, and sure enough, I lost every single race. I went to my dad and complained: "I'm tired papa, can we stop?" "Are you tired, really? Very good, it means that you're on your way to make progress!" I paused, and let it sink in for a few moments. And without a word, I went back running. That's how I learned that getting better goes with accepting to suffer a little. Later on, I also realized that out of any bad experience, be it death, you can take away something positive, something that will halp you grow. Against all odds, I even made a first professional carrier in sports. I am very grateful for both my parents: I am who I am, with all my quirks and all that's to be loved or to be hated, thanks to them.

So many more people to thank! I'm just getting started, sorry to inflict you this. But let's start with the sisters. Esther comes just after me, she married an awesome guy from Congo, and is currently raising two wonderful little girls. She is the closest to what my mom was with us (and still is), making anyone feel home at any time, always on the move, taking care of her family during the day and working at nights, juggling countless tasks and thinking it is all just natural. Then comes Déborah, although she didn't come alone since Joëlla followed 10 minutes later. But believe it or not, she is slightly more than a twin. She has a high sense of justice and a desire to be helpful, which made her switch from the arts history field to the health one, so as to be more true to herself. Joëlla also is incredible. She fights every day her health issues, could not finish high school but still managed to get a bachelor degree, and she now is a professionnal violinist, whose empathy perspires through all her plays. I'm on a roll now, and I don't think you'll be suprised if I tell you that my last sister, Noémie, is decent enough. She also became a professionnal violinist, she runs every day, and she is currently studying psychology. She also spends a lot of energy mediating arguments between people she loves. A family I'm proud of, not only because of their obvisous skills, but because of their virtues.

I want to thank my grand-parents, whose house was the ground base for all of my aunts, uncles, and cousins, who met there during each and every vacation. They made us discover the delightful joy of being cold, wet and exhausted during rainy hikes, to finally end up above a splendid sea of cloud from which protruded just a few sharp peaks, over which Alpine choughs maneuvered with their vigorous flight. They are the true pillars of our extended family. The cycle of life being what it is, they became older and can't hike anymore. I am now very happy to see the whole family striving to take care of them, as much as they have been taken care of. I can sadly not name every single other member of my family, humans or animals, but they are all a crucial part of myself.

I do need to spend some time for the love of my life, Mikaela. We met in Lebanon, she is American, she cares about France as little as I care about the USA, and yet she accepted to come here for me, in the armpit of the old and stinky world. She had the courage to take over my mother's difficult job to bear with my incessant questions. She actually has a lot of answers, since the extend of her knowledge is so wide and well-rounded. She is also an awesome writer, and a qualified editer who plays a large role in making my productions publishable. She is much more than she believes of herself: exceedingly faithful, remarkably generous, paradoxically very introverted but willing to home all the persons in need we come across, and unfortunately suffering of how little her power is to make the world a better place. She also comes with a very nice family in law, and of course, she is the mother of my child Cédric! A stunning baby who spends an excessive amount of energy smiling at every one, all day long (aside from sometimes, when he screams his head off.) He might give me a hard time whenever I get started writing my thesis, but he does it in a very cute way. And he always embodies a very good way for us to get away with our shared legendary absent-mindedness. I'm looking forward to the time I'll be old enough for him to change my diapers.

Life wouldn't be life without friends, old and new ones, whether I see them several times a week or once every two or three blue moons. Friends of the family, friends from church, friends from parkour, friends from the performing world, friends I have no idea how I got to know them. Not to brag, but they are too numerous to name them all.

Finally, let's remember that this is a PhD thesis that I'm writing, and that there is no thesis without a lab, without supervisors, without fellow PhD students, post-docs, interns, researchers, administrative workers, cleaning operatives, and all who are involved in making work enjoyable (sic.) I want to thank them all. Lionel, my director, who saved me from the happy hell of starving performing arts to give me the chance to throw myself in another highly precariously fun situation. Mathieu, my co-supervisor, who was quite present and helpful, always ready to give me quick and valuable feedback, despite he lived in the other end of the country. One expert in computer vision, the other in biomechanics: the perfect fit for the objectives of my doctorate. Thibault, my faithful office colleague, that I often left alone with the sole presence of cold-blooded computer hardware while I worked remotely. Other colleagues from other places such as the INSEP, the LBMC, the Pprime institute, etc. Thank you all!

To sum it up, I owe this work to my family, my friends, my colleagues, and I'm guillible enough to believe I owe it to God above all. I am happy I have overcome it, not only alone but with all of the aforenamed people!

On these words, I suppose I can now start with what I'm here for.

Abstract

Abstract.

Titre, Abstract, Mots clés

Potentiellement une seule section pour Abstract / Résumé, potentiellement en 2 colonnes (cf template Rennes)

Résumé

Résumé.

Contents

General introduction

General introduction.

Intérêt markerless dans le sport

Problèmes de détection de features dans image, calibration et triangulation, scaling et cinématique inverse, et où mon travail s'inscrit (bridge between 2D feature detection in computer vision, and physically consistent 3D biomechanics for sports)

Présentation détaillée de chaque chapitre

Schéma résumé: acquisition, calibration, pose estimation, triangulation&filtrage, scaling, inverse kinematics + applications

1
State of the art

Motion capture (MoCap) in sports is traditionally performed with marker-based (opto-electronic) systems. However, this presents some drawbacks. As a consequence, alternatives are being investigated, among which those offered by Inertial Measurement Units (IMUs) or dept-field (RGB-D) cameras. Markerless analysis from videos sources represents one of the most promising prospects, which has been possible thanks to progress in machine learning. From 2D pose estimation to 3D joint angle determination, this is a new field which opens up new possibilities for motion analysis in a sports context.

This chapter is an up-to-date and more detailed version of the introduction of the previously published paper: "Pose2Sim: An End-to-End Workflow for 3D Markerless Sports Kinematics—Part 1: Robustness" [Pagnon2021].

Contents

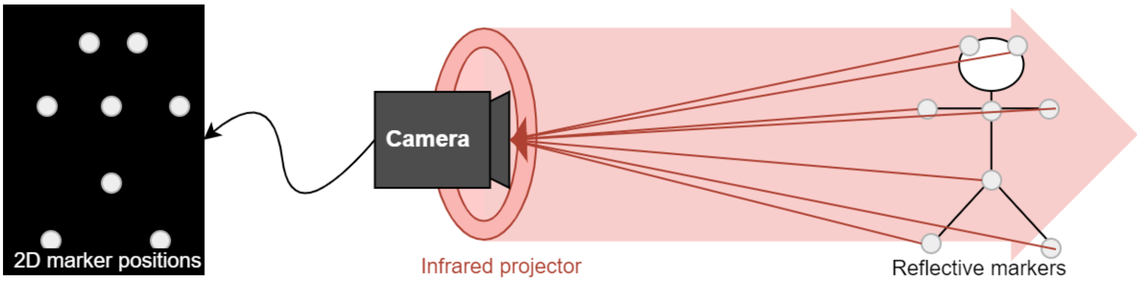
1.1 Overall context of kinematics in sports

1.1.1 General context

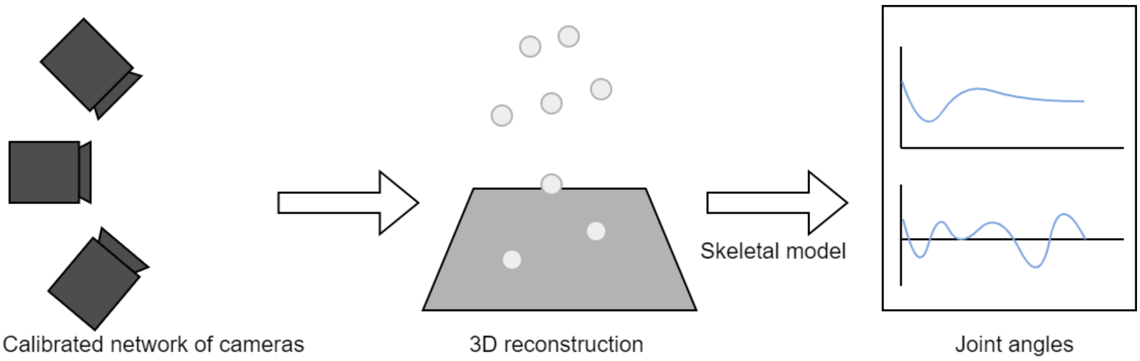
As coaching athletes implies observing and understanding their movements, motion capture (MoCap) is essential in sports. It helps improving movement efficiency, preventing injuries, or predicting performances. For the last few decades, marker-based systems have been considered the best choice for the analysis of human movement, when regarding the trade-off between ease of use and accuracy. However, these methods have proven to be much more challenging in a sports context than in a laboratory setting, and to be generally inappropriate [Mündermann2006, Colyer2018]. As a consequence, other methods have been investigated (see Table 1.1).

1.1.2 Marker-based systems

Marker-based systems use a network of opto-electronic cameras. Each of these cameras are surrounded by a crown of infrared LEDs, which projects light toward the subject, who is equipped with reflective markers. Ideally, only the light reflected from these markers is captured by the cameras. The camera usually pre-processes the image to make it binary, and only outputs the coordinates of the detected marker (Figure 1.1a).



(a) An opto-electronic camera is traditionnally surrounded by a crown of infrared LEDs, projecting light toward the subject. The subject wears markers, which reflect light back to the camera. Marker positions are then known in the camera plane.



(b) Once calibrated, a network of these cameras allows for 3D reconstruction of marker positions. Marker coordinates are then used to infer the posture of the subject.

Figure 1.1: Principles of marker-based motion capture. (Figure 1.1a) presents the functioning of an opto-electronic camera. (Figure 1.1b) shows how a network of calibrated motion capture cameras helps obtaining joint angles.

If calibrated, using a network of these cameras allows for triangulating the 2D coordinates. Calibration involves knowing the cameras' intrinsic properties (such as focal length, optical center, distortion) as well as their extrinsic properties (their position and orientation as regards to the global coordinate system.) See Chapter 2.2 on 3D reconstruction for more details. The reconstructed 3D marker positions are then used to optimize the posture of a physically consistent skeleton, scaled to each individual subject. In particular, this allows for obtaining 3D joint angles at each point in time, commonly referred to as inverse kinematics (IK.)

Yet, reflective marker-based camera systems are complex to set up, are time-consuming, and are very expensive. They also require specific lightning conditions, and involve cumbersome cabling. Moreover, markers may fall off the body of the participant due to sharp accelerations or sweat. They can hinder the natural movement of athletes, which is likely to affect their warm-up, focus, and safety. While the accuracy of landmark location is claimed to be sub-millimetric in marker-based methods [Topley2020], marker placement is tedious, intrusive, prone to positioning variability from the operator [Tsushima2003], and subject to skin movement artifacts, especially on soft tissues. Della Croce et al. found out that inter-operator variations in marker placements range from 13 to 25 mm, which can propagate up to 10° in joint angle prediction [Gorton2009, della Croce1999]. For example, tissue artifacts account for up to a 2.5 cm marker displacement at the thigh, which can cause as much as a 3° error in knee joint angles tissues [Benoit2015, Cappozzo1995]. Joint positions must be calculated explicitly in marker-based methods, which introduces more variability: these errors range up to 5 cm, which can contribute up to 3° of error in lower limb joint angles [Leboeuf2019]. Nevertheless, since marker-based methods benefit from decades of research, they are still considered as the reference method for motion capture.

1.1.3 IMU and RGB-D systems

Consequently, other approaches based on alternative technologies have been investigated over the past years. For instance, wearable Inertial Measurement Units (IMUs) can be placed on an athlete's limbs. IMUs are generally made of an accelerometer, a gyroscope, and a magnetometer. The accelerometer measures the linear acceleration, the gyroscope measures the rotational speed, and the magnetometer measures the orientation of the earth magnetic field. Fusing and integrating these signals allows for the determination of their 3D orientations. The orientation of the athlete's limbs can then be used in combination with a skeletal model to infer their posture (Figure 1.2).

IMUs offer the advantages of getting away from all camera-related issues. They are inexpensive, they do not involve any complex setup and calibration, the field of view is larger, they are not sensitive to self- and gear-occlusions, they can be operated outside of a controlled environment, and they can work in real-time [Johnston2019, Chambers2015]. They still have the drawback of being an external equipment to wear, involving high technical skills from the operator, and are sensitive to ferromagnetic disturbances. Above all, they are exposed to drift over time and need to be calibrated every few minutes. Joint angle accuracy is relatively good in the flexion/extension plane, but less so in other rotational planes where errors are greater than 5° for most motions [Zhang2013, Rekant2022]. Moreover, they are not suitable for joint positions assessment, since these are obtained through multiple integrations of the original signal [Ahmad2013].

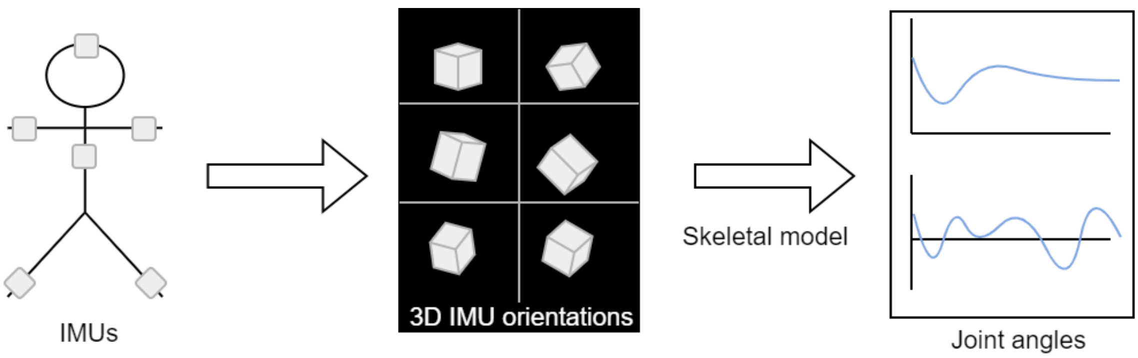


Figure 1.2: IMUs are placed on the subject's limbs. The orientation of the limbs is then used to infer the posture of the subject.

Another approach involves depth-field cameras (RGB-D). Older models projected infrared *structured* light (i.e., a pattern) onto the scene. The relative deformation of the pattern reflected from the scene was then used to estimate depth. Newer models project infrared *modulated* light onto the scene. The time of flight of the light reflected from the scene is then used to estimate depth. Results are commonly considered to be 2.5D, since only the depth of the front facing plane of view is measured. Gait analysis results are natively poor, but after an optimization by a neural network, [Guo2022] manage to get root-mean-square errors under 7° for knee flexion/extension angle at the most challenging part of the gait cycle, although 3D joint angle errors usually stay under 2-3°. However, it may not perform as well on other motions on which the neural network has not been trained. A network of a few RGB-D cameras can give access to full 3D [Carraro2017, Choppin2013, Colombel2020]. Nevertheless, these cameras hardly function in direct sunlight nor at a distance over 5 meters, and they work at lower frame rates (generally under 30 Hz) [Han2013, Pagliari2015].

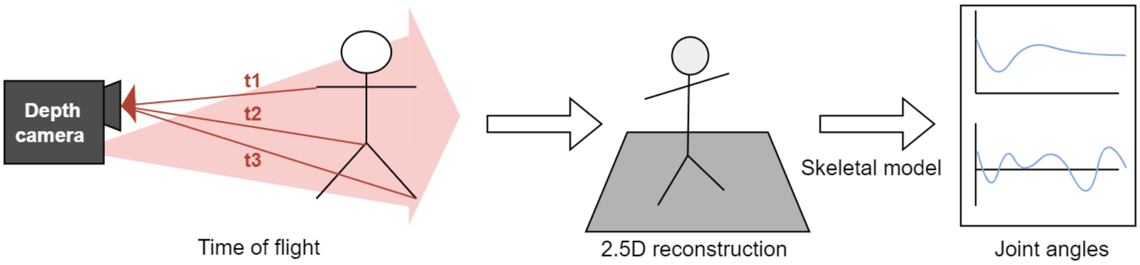


Figure 1.3: A depth-field camera (RGB-D) projects infrared modulated light onto the subject's body. The time it takes for the light to be reflected to the camera sensor (time of flight) depends on distance, and gives access to the depth of the scene. Older RGB-D cameras use structured light rather than time of flight calculations to infer depth.

1.1.4 Markerless systems

A recent breakthrough has come from computer vision, and the advent of 2D pose estimation from image sources, which quickly became more efficient and accurate. The explosion of deep-learning based methods from camera videos, for which the research has skyrocketed around 2016 [Wang2021], is related to the increase in storage capacities and huge improvements in GPU computing. A search on the ScienceDirect database for "deep learning 3D human pose estimation" produced fewer than 100 papers per year until 2015, and the number is now reaching over 1,000, fitting an exponential curve (Figure 1.4).

It has rekindled interest from the biomechanics community towards image-based motion analysis, which is where it all started with the invention of chronophotography in the 19th century by Marey in France, and Muybridge in the USA [Baker2007]. Currently, two approaches coexist in human and animal motion analysis: the first one mostly focuses on joint positions, and is lead by the computer vision and the deep-learning communities; while the second one is interested in joint angles, such as the biomechanics community uses to obtain physically coherent kinematics individualized to each subject. One of the main current challenges is to bridge the gap between these two worlds, and to take advantage of deep-learning technologies for kinematic analysis [Cronin2021, Seethapathi2019].

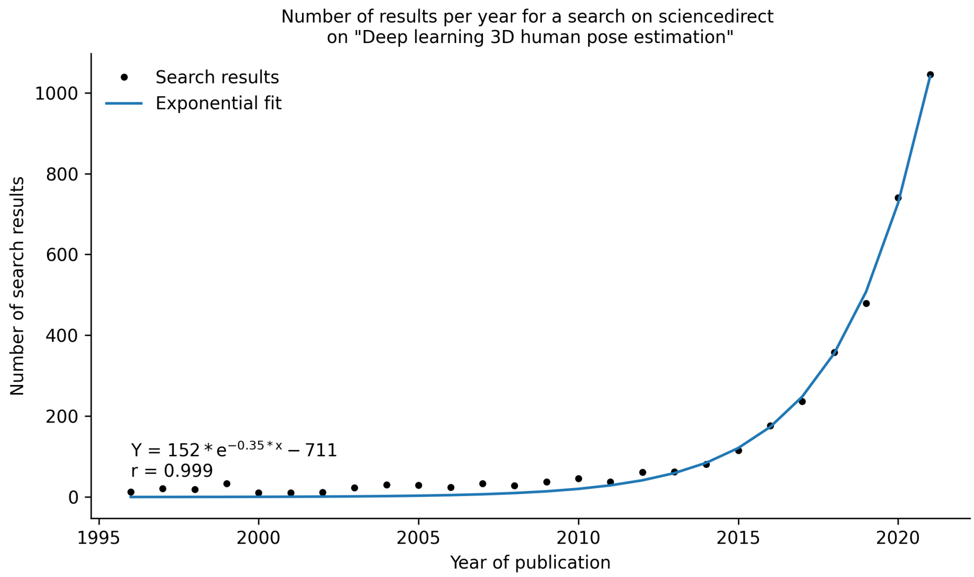


Figure 1.4: The search for "deep learning 3D human pose estimation" (dots) fits an exponential curve (line). The search produced less than 100 results until 2015, and is now well over a 1,000 per year.

1.2 2D markerless analysis

1.2.1 2D pose estimation

The most well-known off-the-shelf 2D human pose estimation solutions are OpenPose [Cao2019] (Figure 1.5), and to a lesser extent AlphaPose [Fang2017]. While both show similar results, OpenPose has the advantage of being a bottom-up approach, whose computation time does not increase with the number of persons detected [Cao2019]. A bottom-up approach first detects all available joint keypoints, and then associates them to the right persons; while a top-bottom approach first detects bounding boxes around each person, and then finds joint keypoints inside of them. OpenPose is the only multi-person 2D pose estimation solution that provides foot keypoints, which are essential for sports motion analysis.

Other approaches have shown even better results on evaluation datasets (see review [Chen2020]), but they are generally slower and not as widespread. The technology, however, is still maturing and some light-weight systems such as BlazePose [Bazarevsky2020], UULPN [Wang2022b], or YOLOv7 [Wang2022a] are being proposed, which can operate in real time on a mobile phone; however, they respectively support single-person detection only, are not accurate enough for quantitative motion analysis, or haven't been embraced by the community yet. Some work has also been done on temporal consistency across frames with OpenPifPaf, which makes the system much faster, and helps it perform better on low-resolution regime or with occlusions such as in crowds [Kreiss2022].

Two other 2D pose estimation toolboxes are DeepLabCut [Mathis2018,Lauer2022] and SLEAP [Pereira2022], which were initially intended for markerless animal pose estimation. They have the advantage that they can be custom trained for the detection of any human or not human keypoint with a relatively small dataset.

All the tools presented in this section are open-source. See Chapter 2.1.3 on Machine learning for 2D pose detection for more technical details.

Figure 1.5: 2D pose estimation by OpenPose. Image courtesy of [Cao2019].

1.2.2 2D kinematics from 2D pose estimation

Some authors bridge 2D pose estimation to more biomechanically inspired variables, such as in gait kinematics analysis. Kidzinski et al. present a toolbox for quantifying gait pathology that runs in a Google Colab [Kidziński2020]. Stenum et al. evaluate gait kinematics calculated from OpenPose input concurrently with a marker-based method. Mean absolute error of hip, knee and ankle sagittal angles were 4.0°, 5.6° and 7.4° [Stenum2021]. Liao et al. have not released their code, but they use OpenPose outputs to train a model invariant to view [Liao2020]. Viswakumar et al. perform direct calculation of the knee angle from an average phone camera processed by OpenPose [Viswakumar2019]. They show that OpenPose is robust to challenging clothing such as large Indian pants, as well as to extreme lightning conditions. Other sports activities have been investigated, such as lower body kinematics of vertical jump [Drazan2021] or underwater running [Cronin2019]. Both works train their own model with DeepLabCut. Serrancoli et al. fuse OpenPose and force sensors to retrieve joint dynamics in a pedaling task [Serrancolí2020]. Although it doesn't specifically use deep-learning approaches, another noteworthy tool for 2D sports movement analysis is Kinovea [Fernández-González2020]. It allows to manually label keypoints on a frame, and track them in time in order to obtain point trajectories or angle data.

1.3 3D markerless analysis

1.3.1 3D pose estimation

There are a lot of different approaches for markerless 3D human pose estimation, and listing them all is beyond our scope (see review [Wang2021]). Some more ancient ones are not based on deep-learning and require specific lightning and background conditions, such as visual-hull reconstruction [Ceseracciu2014]. Some directly lift 3D from a single 2D camera (see review [Liu2022c]), with different purposes: one estimates the positions of a set of keypoints around the joint instead of determining only the joint center keypoint, so that axial rotation along the limb is solved [Fisch2020]; SMPL and its sequels retrieve not only joint positions and orientations, but also body shape parameters [Loper2015]; while XNect primarily focuses on real time [Mehta2020]. A few approaches even strive to estimate 3D dynamics and contact forces from a 2D video input [Li2019, Rempe2021, Louis2022]. Some incorporate kinematic priors into their neural networks in order to take advantage of human knowledge [Xu2020]. Surprisingly, this does not seem to be done in multi-view approaches. Rempe et al. solve occlusions from a 2D input [Rempe2020], but this remains a probabilistic guess that may be unsuccessful in case of unconventional positions of hidden limbs, whereas using more cameras would have given more trustworthy results.

Some research attempts to solve 3D pose estimation from a network of uncalibrated cameras, i.e., cameras whose extrinsic parameters (translation and rotation with respect to the coordinate system), intrinsic parameters (focal length, pixel size, etc.), and distortion coefficients are not known (See Chapter 2.2 on 3D reconstruction for more details.) It either uses 2D pose estimations of each view as visual cues to calibrate on [Takahashi2018, Xu2021, Liu2022a], or an adversarial network that predicts views of other cameras, compares them to real views, and adjusts its calibration accordingly [Ershadi-Nasab2021]. Dong et al. recover 3D human motion from unsynchronized and uncalibrated videos of a repeatable movement found on internet videos (such as a tennis serve performed by a celebrity) [Dong2020]. Using uncalibrated videos is still a very experimental trend, that would require more research before being used in biomechanics.

We choose to focus on the methods that estimate 3D pose by triangulating 2D pose estimations from a network of multiple calibrated cameras. The classical evaluation metric is the MPJPE (Mean Per Joint Position Error), which is the average Euclidian distance between the estimated joint coordinate and its ground truth. Most methods take OpenPose as an input for triangulation, and more specifically the body_25 model. Labuguen et al. evaluate 3D joint positions of a pop dancer with a simple Direct Linear Transform triangulation (DLT [Hartley1997, Miller1980]) from 4 cameras [Labuguen2020]. Apart from the upper body for which error goes up to almost 700 mm, the average joint position error is about 100 mm. Nakano et al. examine three motor tasks (walking, countermovement jumping, and ball throwing), captured with 5 cameras and triangulated with the same methods, with a subsequent Butterworth filter [Nakano2019]. 47% of the errors are under 20 mm, 80% under 30 mm, and 10% are above 40 mm. The largest errors are mostly caused by OpenPose wrongly tracking a joint, for example by swapping the left and the right limb, that causes large errors up to 700 mm. This may be fixed either by using a better 2D pose estimator, or by using more cameras to reduce the impact of an error on a camera, or else by considering the temporal continuity in movement. Needham et al. use 9 cameras and find that ankle MPJPEs are within the margin of error of marker-based technologies (1–15 mm), whereas knee and hip MPJPEs are greater (30–50 mm). These errors are systematic and likely due to "ground-truth" images being mislabeled in the training dataset [Needham2021]. They also run the comparison with AlphaPose and with DeepLabCut. While AlphPose's results are similar to OpenPose's; DeepLabCut errors are substantially higher.

Slembrouck at al. go a step further and tackle the issue of limb swapping and of multiple person detection [Slembrouck2020]. In case of multiple person detection, one needs to make sure they associate the person detected on one camera to the same person detected on other ones. Slembrouck et al. manage to associate persons across cameras by examining all the available triangulations for the neck and mid-hip joints: the persons are the same when the distance between the triangulated point and the line defined by the detected 2D point and the camera center is below a certain threshold. They only focus on lower limb. Their first trial features a person running while being filmed by seven cameras, whereas their second one involves a person doing stationary movements such as squats while filmed by 3 cameras. After filtering, the average positional error in the first case is about 40 mm, and it is roughly 30 mm in the second case (less than 20 mm for the ankle joint). Other authors deal with the multiperson issue in a slightly different way [Bridgeman2019, Chu2021, Dong2019]. In average, if the detected persons are correctly associated and the limbs don't swap, the average joint position error for an OpenPose triangulation is mostly below 40 mm.

Some triangulation methods not based on OpenPose reach even better results on benchmarks, although it comes at the cost of either requiring heavy computations, or of being out of reach for non-expert in deep-learning and computer vision. The classic approach reduces the joint detection heatmap to its maximum probability, and then to triangulate these scalar 2D positions. Instead of this, the main state-of-the art methods directly perform a volumetric triangulation of the whole heatmaps, and only then take the maximum probability as a 3D joint center estimate. By working this way, they keep all the information available for as long as possible. They manage to lower their MPJPE to about 20 mm [He2020, Iskakov2019].

1.3.2 3D kinematics from 3D pose estimation

Numerous studies have focused on the accuracy of 3D joint center estimation, but far fewer have examined joint angles [Zheng2022]. Yet, when it comes to the biomechanical analysis of human motion, it is often more useful to obtain joint angles. Joint angles allow for better comparison among trials and individuals, and they represent the first step for other analysis such as inverse dynamics. This issue is starting to be tackled. Zago et al. evaluate gait parameters computed by triangulating 2 videos processed by OpenPose, and notice that straight gait direction, longer distance from subject to camera, and higher resolution make a big difference in accuracy [Zago2020]. D'Antonio at al. perform a simple triangulation of the OpenPose output of two cameras, and compute direct flexion-extension angles for the lower limb [D'Antonio2021]. They compare their results to IMU ones, and point out that errors are higher for running than for walking, and are also rather inconsistent: Range of Motion (ROM) errors can reach up to 14°, although they can get down to 2 to 7° if the two cameras are set laterally rather than in the back of the subject. Wade et al. calculate planar hip and knee angles with OpenPose, AlphaPose, and DeepLabCut with the input of 9 cameras [Wade2021]. They deem the method accurate enough for assessing step length and velocity, but not for joint angle analysis. AniPose, a Python open-source framework, broadens the perspective to the kinematics of any human or animal with a DeepLabCut input, instead of OpenPose. They offer custom temporal filters, as well as spatial constraints on limb lengths [Karashchuk2021]. To our knowledge, it has only been concurrently validated for index finger angles in the sagittal plane, resulting in a root-mean-square error of 7.5° [Geelen2021].

The previous studies calculated simple planar angles between 3 joint centers. However, the human skeleton is complex and not only made of pin joints: aside from the flexion/extension rotation axis, the abduction/adduction axis and the internal/external axis are typically also engaged; and some joints also involves some translation, such as the shoulder. In this case, either several markers per joints or a solid skeletal model are needed. So far, little work has been done towards obtaining 3D angles from multiple views [Zheng2022]. Aside from our solution (see Chapter 3 on Pose2Sim), two main others are worth mentioning. Theia3D is a commercial software application for human gait markerless kinematics. It estimates the positions of a set of keypoints around the joint, and then uses a multi-body optimization approach to solve inverse kinematics [Kanko2021a, Kanko2021b]. They notice an offset in hip and ankle angles between their markerless system and the reference marker-based one, likely due to different skeletal models. Once this offset is removed, the root-mean-square error (RMSE) in lower limb roughly ranges between 2 and 8° for flexion/extension and abduction/adduction angles, and up to 11.6° for internal/external rotation. Although the GUI is user-friendly, it is neither open-source nor customizable. OpenCap [Uhlrich2022] has recently been released, and offers a user-friendly web application working with low-cost hardware. It predicts the coordinates of 43 anatomical markers from 20 triangulated keypoints, imports them in OpenSim, and performs classic inverse kinematics with numerous inferred markers and a skeletal model. However, the source code has not yet been released.

Other approaches don't focus so much on keypoint detections, and capture the whole shape of participants. [Reveret2020] records the 3D shape of a speed climber in a studio equipped with 68 video cameras, and then animates it to follow 2 calibrated drone views by optimizing its manifold parameters. This allows for tracking the center of mass and for detecting hand contacts with holds, without the use of machine learning. Simi shape, a commercial software, jointly learns 2D shape and 2D keypoint coordinates. It claims to be able to obtain accurate kinematics with few cameras, thanks to the additional information shape detection provides (validation with their newer machine learning based process not yet published.) Pose estimation from videos can also be fused with the information provided by other sensors, such as IMUs [Bao2022, Zhang2020]. This enables solving occlusions in videos, and compensation of the drift consecutive to the integration of accelerations and rotation speeds in IMUs. For example, Haralabidis et al. fuse OpenPose results from a single monocular video and two IMU outputs, and solve kinematics of the upper body in OpenSim (an open-source biomechanical 3D analysis software [Delp2007, Seth2018]) in order to examine the effects of fatigue on boxing [Haralabidis2020]. Results are promising, but this cannot be considered as fully markerless. Fusing the depth map of a single RGB-D camera with its image processed by OpenPose has also been investigated [Liu2022b], although 3D coordinate errors were close to 10 cm.

1.4 Statement of need

According to Atha [Atha1984], an ideal motion analysis system involves the collection of accurate information, the elimination of interference with natural movement, and the minimization of capture and analysis times. Yet, even though a marker-based system gives relatively accurate results, it requires placing markers on the body which can hinder natural movement, it is hard to set up outdoors or in context, and it is strenuous to analyze. As a consequence, in the overwhelming majority of cases, coaches solely use subjective visual observation to assess an athlete's movement patterns and to compare performances. As a matter of fact, despite the advantages of technology, investing in it has its pitfalls: the information gathered can be unhelpful, or inaccurate, or not easily interpretable, or simply not implementable in the context of sports [Windt2020].

The emergence of markerless kinematics opens up new possibilities. Indeed, a network of RGB cameras does not assume any particular environment, and it does not hinder the athlete's movement and focus. However, it still requires delicate calibration, complex setup, large storage space, and high computational capacities. Gathering reliable and usable kinematic data in context is an ambitious challenge, but research has been accelerating in the last few years (Figure 1.4), as have better results.

The objective of this thesis is to participate in building a bridge between the communities of computer vision and biomechanics, by providing a simple and open-source pipeline connecting the two aforementioned state-of-the-art tools: OpenPose and OpenSim. Robustness and accuracy will be assessed, and concrete applications in elite sports context will be discussed.

Sensor type	Mono/Multi camera	2D/3D	Pros and Cons
Opto-electronic	Multi	3D	+ Standard + Good ease-of-use/accuracy trade-off - Not suitable in sports contexts
IMU	N/A	3D	+ Good angle accuracy - Angle drift & poor position analysis - Can be cumbersome
RGB-D	Mono	2.5D	+ Markerless - Generally poor accuracy - Frame-rate $\leq$ 30 Hz - Needs distance $\leq$ 5 m and no direct sunlight
	Multi	3D	+ Full 3D markerless + Better accuracy - Same as above re. frame-rate, distance, and light
RGB video	Mono	2D	+ Very robust in all contexts + Cheap and easy to set up - Only 2D - Not very accurate
		3D	+ Full 3D with one single RGB camera - Probabilistic guess when occlusions: accuracy $\searrow$ - Slow
	Multi uncalibrated	3D	+ Removes difficult step of calibration - Still experimental
	Multi calibrated	3D	+ Solves occlusions + Robust - Systematic offsets due to labelling errors - Calibration can be challenging
	Multi calibrated with kin. constraints	3D	+ Compensates offsets + Constrains limb lengths and joint angles - Still inaccurate pelvis angles
Sensor fusion	N/A	3D	• With IMUs: More accurate, but not markerless • With one RGB-D camera (Depth + OpenPose on RGB): still inaccurate

Tableau 1.1: Pros and cons in state-of-the-art approaches for human motion analysis. The multi-person prospect is not addressed, as it can be available with all approaches, but it is not always. IMU: Inertial Measurement Unit. N/A: Not Applicable. kin.: kinematic. RGB-D: red-green-blue-depth.

2
Theoretical framework

Obtaining 3D kinematics from a network of calibrated video cameras involves understanding a certain theoretical framework. First, keypoints must be recognized in images. This is mostly achieved with machine learning models. Then, all the 2D features detected for each cameras need to be reconstructed in the 3D space. Finally, these coordinates must be constrained to a biomechanically consistent model, in order to obtain coherent 3D joint kinematics.

Contents

2.1 2D pose detection

2.1.1 Why machine learning?

As a first step, achieving motion analysis from a network of cameras involves detecting features in images. These features can be whole human beings, joint centers, body landmarks, sports gear such as tennis balls, climbing holds, or much more.

Two broad approaches can be implemented: the first one consists in using dedicated algorithms for each task. The gist of it is to understand the task well enough to build an appropriate solution: this is a knowledge-driven approach. Among other techniques, corner and contour detection, color thresholding, affine transformation, template matching, watershed segmentation, can be used. For example, if one wants to differentiate two boxers wearing respectively a blue and a red shirt, they can filter them by color. If one needs to identify on which portion of a speed climbing wall an athlete is, they can match the template of each holds on the whole image. OpenCV [Bradski2000] provides convenient tools for this purpose, in C++ and Python languages. This approach is often fast, but also quite complicated to implement, and neither flexible nor robust. If there is other red or blue patches in the boxing scene, if the boxer wears green or if the light is poor, this will not work anymore. Likewise for holds, if the sun casts a large shadow which changes its apparent shape, or if holds are seen from a different perspective.

The second approach takes advantage of machine learning algorithms, which constitute an entirely different paradigm. The idea is to show the machine enough examples for it to "understand" by itself its underlying attributes, so that it manages to detect and label automatically new images: this is a data-driven approach. It can be used for both aforementioned tasks, in a much more flexible way: if one wants the system to recognize boxing gloves or holds in challenging conditions, they simply have to include such examples while training the model. The machine learning approach is also suitable for other tasks, such as whole-image classification (e.g., determining whether this is a boxing or a BMX scene), object detection (e.g., localization of a bike and of a person with a bounding box), background extraction [Bouwmans2019], semantic and instance segmentation (e.g., extracting the shape of the bike and of the person) [Minaee2021], or keypoint detection (e.g., localization of human joint centers and keypoints on a bike [Chen2020]) (Figure 2.1). By 2015, data-driven methods definitely took over knowledge-driven ones in vision analysis problems, and by extension in sports motion analysis from videos (Figure 1.4).

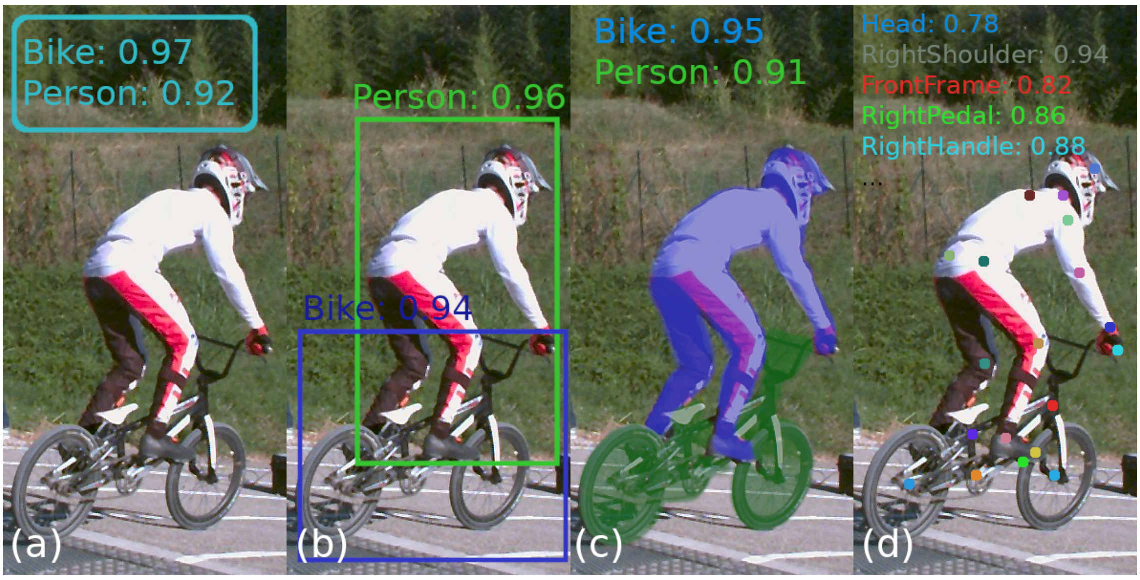


Figure 2.1: Different types of image analysis. (a) Whole image classification, (b) Object detection and localization, (c) Instance segmentation and shape extraction, (d) Keypoint detection.

2.1.2 Machine learning timeline and principles

Machine learning is a subset of artificial intelligence (AI.) As such, one can trace its origin back to the discovery of the natural neuron at the end of the 19th century, by Nobel Prize Ramón y Cajal [López-Muñoz2006], followed half a century later by the first model of an artificial neuron [McCulloch1943]. A natural neuron is a simple learning unit, which collects the nervous influx sent by other neurons to its dendrites, and sends an action potential when the total influx weighted and summed in the soma overcomes a threshold value. This potential is then transmitted through the axon to the next neuron as a new influx. Similarly, an artificial neuron receives output vectors from previous neurons, weighs and sums them with a summation function, and transfers the resulting output vector to the next neurons if it reaches a certain threshold determined by an activation function (Figure 2.2a-b).

The perceptron, invented in 1956 [Rosenblatt1958], represents the first practical application of an artificial neuron. It acts as a binary classifier which predicts class 1 if the neuron is fired, and class 0 otherwise. It automatically adjusts its weights by learning from previously labeled example data (see Algorithm 1 and Figure 2.2b). It could be used, for example, to predict whether an athlete is going to be "good" or not, given his force-velocity results on an ergometer test (see step-by-step Example 1 and Figure 2.3), and given enough example data. Needing previously labeled data makes it is a supervised classifier – we will not discuss unsupervised methods here. Of course, this example is oversimplified. Being good or not as a sport is a complex and multifactorial outcome, and two variables can't sum it up. However, the perceptron can take more than two variables as inputs (for example, force, velocity, and endurance), and it can also be generalized to multiclass classification with more than two outputs (for example, to differentiate between strong, explosive, and resistant type of athletes.)

Nevertheless, it often takes a lot of iterations over good quality training data for the perceptron to converge. Moreover, it does converge if and only if the data are linearly separable, i.e., if they can be separated with a straight line [Novikoff1963] (see Figure 2.4). Some fundamental problems such as the XOR gate can't be solved with a basic single layer Artificial Neural Network (ANN) [Minsky1969]. This constituted one of the early setbacks for AI. Then, the high computational cost of these approaches, combined with the complexity of common-sense problems, hampered the trust in learning methods. Indeed, vision and language problems require enormous amounts of data, and can't be solved with a simple dictionary (for example, "the spirit is willing but the flesh is weak" becomes "the vodka is good but the meat is rotten" when translated back and forth from English to Russian.) Overinflated promises and expectations, followed by disappointment in academia and industries, led to cuts in funding, and eventually loss of skills in the 1970s: this is referred to as the first AI winter.

The AI field survived by focusing on specific problems, called expert systems. In the early 1980s, a new rise was triggered by massive funding such as the Japanese Fifth Generation Computer project, aiming to build a supercomputer that could solve any problem. Shortly after, multi-layer neural networks were made possible with the (re)discovery of backpropagation [Rumelhart1986], or more rigorously of weight adjustment thanks to the backpropagation of error gradient, from the last layer to the first one. As it is not the central subject of this thesis, the algorithm and early references will not be detailed here, but the interested reader can refer to [Goodfellow2016]. This allowed for solving non-linearly separable problems, and for tackling real world issues (Figure 2.2c.). [Cybenko1989] proved that one single intermediate layer is enough to solve any given classification problem, granted that this layer contains enough neurons (although sometimes too many to make it possible in practice.) On the other hand, kernel tricks were also rediscovered [Aizerman1964, Hofmann2008], and made non-neural networks such as support vector machines (SVMs) [Boser1992] able to treat non-linearly separable data with much less training data, more optimally, and on a clearer mathematical ground (Figure 2.4). However, again, unrealistic expectations were confronted with unplanned technical difficulties both on expert systems and on general intelligence projects. This led to a second AI winter in the 1990s.

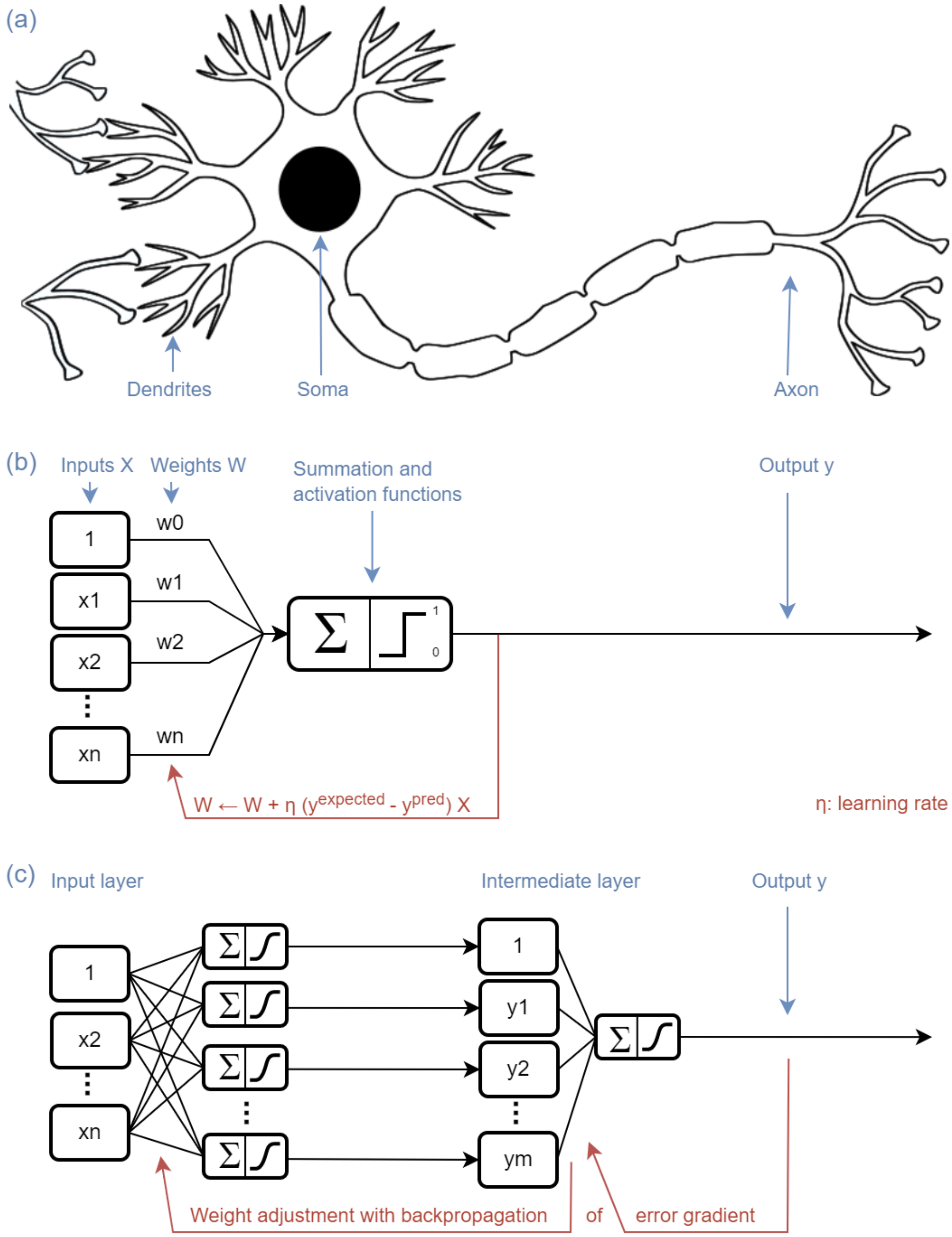


Figure 2.2: The artificial neuron (b) has been modeled after the natural neuron (a). Inputs and weights act as the total nervous influx firing the dendrites. The collected values are summed, and a signal is activated if a threshold is overcome, as the soma does in a natural neuron. The output signal is conveyed the axon in a natural neuron. (b) In the case of a perceptron, the neuron adjusts its weights to minimize the error between the predicted and the expected output. It can be used as a classifier, which outputs class 1 or class 0 depending on the inputs. (c) A dense (fully connected) neural network with one intermediate layer and backpropagation can solve any non-linearly separable classification.

Algorithm 1 Perceptron

Let $\overrightarrow{X^0}$ be the input vector of a first instance of variables $(1, x_1^0, \cdots x_M^0)$, $\overrightarrow{W^0}$ the corresponding weights randomly initialized $(w_0^0, w_1^0, \cdots w_M^0)$ with w_0^0 a bias, and $y^{0,pred}$ the output predicted binary class.

1: The summation function is computed:

$$\overrightarrow{W^0} \cdot \overrightarrow{X^0} = \sum_{m \in [0,M]} w_m^0 x_m^0 \tag{2.1}$$

2: This result is processed by an activation function, which is a threshold in the case of the perceptron. It determines whether the neuron will be fired or not, i.e., whether one or the other class will be predicted. $y^{0,pred} = 1$ corresponds to one class, and $y^{0,pred} = 0$ to the other.

$$y^{0,pred} = \begin{cases} 1 & \text{if } \overrightarrow{W^0} \cdot \overrightarrow{X^0} > \theta, \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

3: This prediction $y^{0,pred}$ is compared to the actual class $y^{0,expected}$.

$$\varepsilon^0 = y^{0,expected} - y^{0,pred} \tag{2.3}$$

4: Then weights are updated:

$$\overrightarrow{W^1} = \overrightarrow{W^0} + \eta\ \varepsilon^0\ \overrightarrow{X^0} \tag{2.4}$$

with η the learning rate $\in [0,1]$. Note that if the class is correctly predicted, then $\varepsilon^0 = 0$ and weights are not adjusted.

5: The algorithm is repeated with another example $\overrightarrow{X^1}$, and so on until it has gone through the whole batch of the training set. If weights still need to be updated, one can go over it again, for a determined number of epochs or until the average error is under a given value. Then the perceptron is considered trained, and ready to correctly predict a class y with the retained weights.

Example 1 Athlete classification with a perceptron

N.B. The code for running this example is available on the thesis repository `https://github.com/davidpagnon/These_David_Pagnon/blob/main/Thesis/Chap2/perceptron.py`.

Let's consider force-velocity test results as an input
$\overrightarrow{X} = (1, velocity\ (m/s), force\ (hN))$,
and the classification of an athlete as "good" or "bad" as an output $y = 1\ or\ 0$.
A batch of training data, i.e., example data the perceptron will learn from, could be:
$\{(\overrightarrow{X^i}, y^{i,expected})\}_{i \in [0,4]} = \{((1,1,5),1),((1,2,3),0),((1,7,1),1),((1,4,1),0),((1,5,4),1)\}$.
Let's randomly initialize weights at $\overrightarrow{W^0} = (-9,1,3)$, take a threshold θ=0.1, and a learning rate η = 0.3.

The first instance of the training set gives:
$\overrightarrow{W^0} \cdot \overrightarrow{X^0} = \sum_{m \in [0,2]} w_m^0 x_m^0 = -9 \times 1 + 1 \times 1 + 3 \times 5 = 7.$
Now $\overrightarrow{W^0} \cdot \overrightarrow{X^0} = 7 > \theta = 0.1$, so $y^{0,pred} = 1$.

$y^{0,expected} = 1 = y^{0,pred}$, so the prediction is true and weights don't need to be updated.
As a consequence, $\overrightarrow{W^1} = \overrightarrow{W^0} = (-9,1,3)$.

The second instance gives $\overrightarrow{W^1} \cdot \overrightarrow{X^1} = (-9,1,3) \cdot (1,2,3) = 2 > \theta = 0.1$, so $y^{1,pred} = 1$.
But $y^{1,expected} = 0 \neq y^{1,pred} = 1$, so weights need to be updated.
The error is $\varepsilon^1 = y^{1,expected} - y^{1,pred} = 0 - 1 = -1$.
As a consequence, $\overrightarrow{W^2} = \overrightarrow{W^1} + \eta \ \ \varepsilon^1 \ \ \overrightarrow{X^1} = (-9,1,3) + 0.1 \times (-1) \times (1,2,3) = (-9.3,0.4,2.1)$.

Third instance: $\overrightarrow{W^2} \cdot \overrightarrow{X^2} = (-9.3,0.4,2.1) \cdot (1,7,1) = 3 - 4.4 < 0.1$, so $y^{2,pred} = 0$.
$y^{2,expected} = 1 \neq y^{2,pred} = 0$, so weights need to be updated.
$\varepsilon^2 = y^{2,expected} - y^{2,pred} = 1$.
$\overrightarrow{W^3} = \overrightarrow{W^2} + \eta \ \varepsilon^2 \ \overrightarrow{X^2} = (-9.3,0.4,2.1) + 0.1 \times 1 \times (1,7,1) = (-9,2.5,2.4)$.

Fourth instance: $\overrightarrow{W^3} \cdot \overrightarrow{X^3} = (-9,2.5,2.4) \cdot (1,4,1) = 3.4 > 0.1$, so $y^{3,pred} = 1$.
$y^{3,expected} = 0 \neq y^{3,pred} = 1$, so weights need to be updated.
$\varepsilon^3 = y^{3,expected} - y^{3,pred} = -1$.
$\overrightarrow{W^4} = \overrightarrow{W^3} + \eta \ \varepsilon^3 \ \overrightarrow{X^3} = (-9,2.5,2.4) + 0.1 \times (-1) \times (1,4,1) = (-9.3,1.3,2.1)$.

Fifth instance: $\overrightarrow{W^4} \cdot \overrightarrow{X^4} = (-9.3,1.3,2.1) \cdot (1,5,4) = 17.6 > 8$, so $y^{4,pred} = 1$.
$y^{4,expected} = 1 = y^{4,pred} = 1$, so weights don't need to be updated.
$\overrightarrow{W^5} = \overrightarrow{W^4} = (-9.3,1.3,2.1)(Figure\ 2.3)$.

Next instances: Once we have gone over the batch of training data, if the average error is below a given value, we can assume that the perceptron is trained. If not, we can use the next batch to pursue training. If it still didn't converge after all batches, we can iterate over all training data again, for a given number of times. If results are still not satisfying, either the data are not linearly separable, or the training sample is not large enough or of good enough quality. In our case, it seems like our example data have allowed us to correctly separate good and bad athletes based on their force and velocity test results (Figure 2.3).

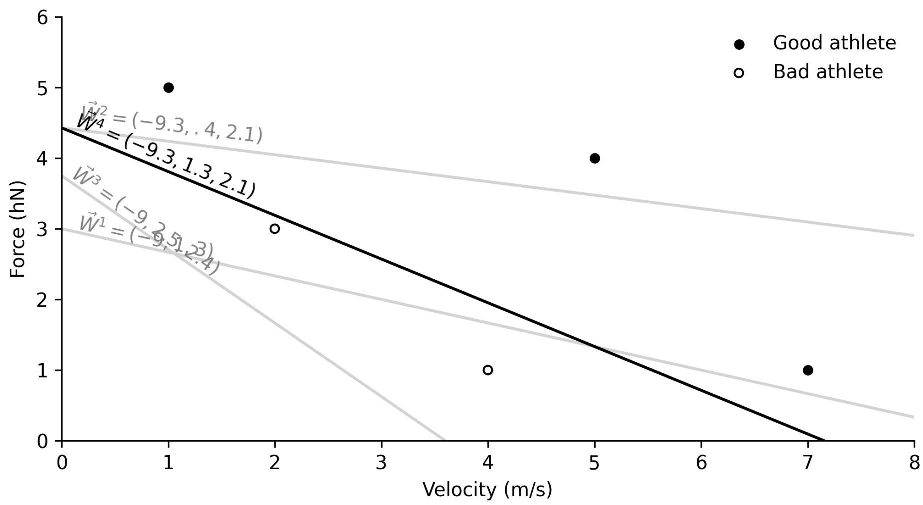


Figure 2.3: Classification of athletes as "good" (black dot) or "bad" (circle) according to their Force-Velocity results. Weights are adjusted (grey lines), until the perceptron classifies athletes correctly (black line.)

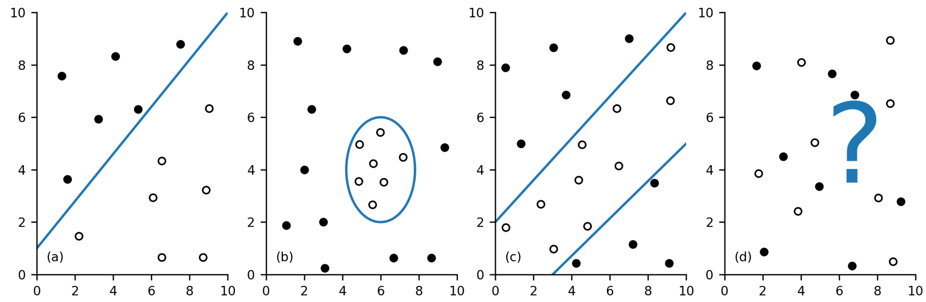


Figure 2.4: Single layer artificial neural networks such as the perceptron can only classify linearly separable data. (a) is linearly separable. (b) is not linearly separable. However, data are contained in an ellipse. The equation of an ellipse is of the form $a \times x^2 + b \times y^2 = 1$, so if we transform the feature variables into $X = x^2$ and $Y = y^2$, the data become linearly separable. (c) is equivalent to a fundamental XOR gate, and is not linearly separable, which was part of the reasons for the first AI winter. It can either be solved by combining several layers of artificial neurons, or by complex kernel tricks which map the data from the original space into a higher dimensional space where they become linearly separable. (d) is possibly not separable at all. AI: Artificial Intelligence. XOR: Exclusive OR.

From the end of the 1990s, there has been no theoretical breakthrough in AI, but larger databases have become available with the advent of the Internet, and greater computational power has become accessible, especially thanks to groundbreaking progress in Graphics Processing Units (GPUs), which made heavy parallel computing available to the wider audience. As a consequence, more layers could be used in neural networks, which progressively set off the onset of deep learning. Finally, complex "common-sense" problems, such as natural language processing or image recognition, could be treated with some success [Baral2018].

One particular type of deep learning algorithms is the convolutional neural network (CNN), which is particularly suited for image recognition. It was first used for classifying handwritten and low-resolution digits [LeCun1998], and then applied to more complex images as greater computing resources became available [Krizhevsky2017]. Nowadays, CNNs have sometimes surpassed humans at image classification [Cireşan2012, Lu2015]. A convolution layer consists in a series of filters that slide across the image, each of them outputting a result close to 0 or to 1, depending on how well it can be overlaid on each image area. In the same way as with a simple artificial neuron, each of these filters can be seen as a weight vector $\overrightarrow{W}$, and each image area as an input vector $\overrightarrow{X}$. The filters of the first convolution layer are simple patterns such as lines, but then they become circles and corners, until the last layers, when they have developed into complex features corresponding to whole object parts. Once a filter has covered the whole image, it forms a feature map, which will then be downsampled by a pooling layer in order to save computing resources. All the feature maps produced by each filter are processed by a determined number of other convolution layers, and then flattened into a 1D vector. This 1D vector is processed by a few dense layers (dense layers are fully connected, i.e., all outputs are produced by a weighted sum of each input), and lastly a softmax layer computes a probability for the image to correspond to each available class. If the CNN is correctly trained, the class with highest probability corresponds to the correct one: for example, if the image displays a BMX start, the probability for the bike class will be the highest (Figure 2.5).

However, results will not be good until a lot of iterations are done on a lot of data. Indeed, filters at each layer are randomly initialized, and then refined with backpropagation in order to predict all classes as best as possible. One of the risks is overfitting, i.e., to excessively adapt to the training data and to fail to generalize to new data. This is dealt with by cross-validation, i.e., the separation between training and test data, by regularization methods such as batch normalization and dropout, and by data augmentation, e.g., image rotations, crops, color distortion, noise addition, etc. [Hawkins2004, Chicco2017]. An enormous amount of data is also needed to correctly train the CNN, which makes it complicated when unusual classes need to be recognized (for example, a climbing hold, a BMX starting gate, a medial malleolus on the ankle, etc.) Fortunately, one can consider that a CNN trained on a massive dataset, such as ImageNet and its 14 million annotated images [Deng2009], has learned to recognize most features that can be found in any image. One can take the learned filters of its convolutional layers as is, use them as a feature extractor (sometimes called backbone), and just fine-tune the last dense layers to recognize new classes. It will be much less computationally expensive to train, and will need much fewer data: about a hundred images, instead of thousands. This is called transfer learning [Pan2009].

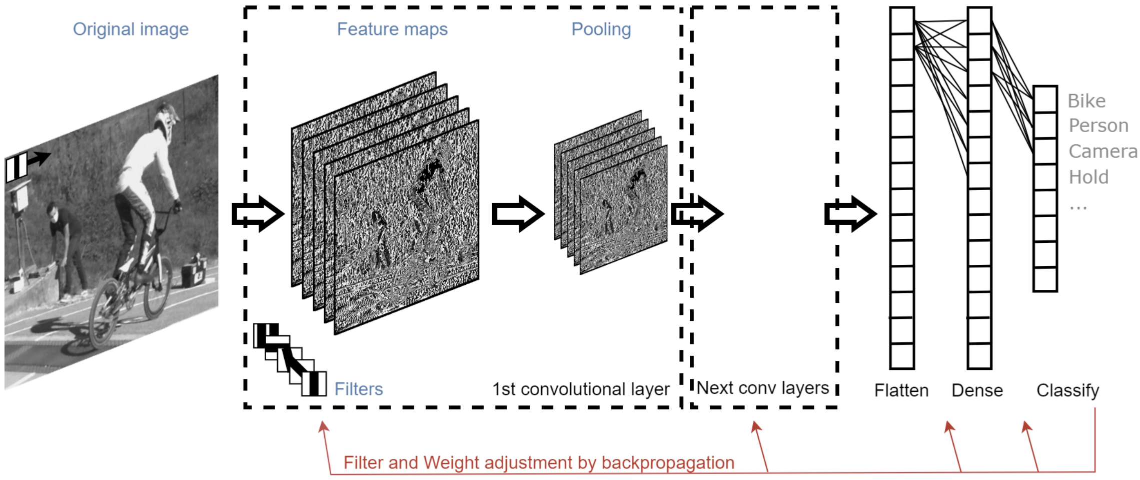


Figure 2.5: A simplified convolutional neural network (CNN.) A convolutional layer consists in a series of filters running across the input image, and producing feature maps, which are then downsampled by pooling. Filters become more and more elaborated along layers, and produce feature maps which look like whole object parts. Filters and weights are randomly initialized at first, and then are adjusted by backpropagation. After the convolutional layers, the feature maps are flattened to produce a 1D vector, which is then processed by dense layers, and finally a softmax layer computes a probability for the image to correspond to each available class.

Now, classification of a whole image is not sufficient in sports motion analysis. One needs to detect where an object or a person is, and ideally to localize more precise features such as joint centers so as to estimate the person's pose.

2.1.3 Machine learning for 2D pose detection

Older methods for object detection used to run a sliding and pyramidal window across the image, and then to apply a non-neural classifier on each window, such as an SVM on carefully handcrafted histogram of oriented gradients descriptors (HOG) [Dalal2005]. They then had to be followed by non-maximum suppression, in order to select one bounding box over many overlapping ones. As the classifier is run on each window iteration like if they were independent images, these methods were very computationally intensive, and in the same time not very robust nor accurate.

More modern approaches are based on CNNs, and as such, they involve a preliminary step: extracting the last layer of a pre-trained neural network such as ImageNet, in order to make it able to classify the objects of interest. One of the precursors, R-CNN (Regions with CNN features) [Girshick2014], first looks for a lesser amount of regions of interest (ROIs) by selective search, instead of with a sliding window. Selective search is an algorithm which segments image based on pixel intensities, without any learning involved [Uijlings2013]. Then three learning models are used: one CNN for extracting features from each ROI, an SVM for classifying each ROI, and a regression model for adjusting bounding boxes. It takes about 45 seconds to process a single image on benchmarks. Fast R-CNN [Girshick2015] uses one single network for all steps, and switches the first two: it first extracts features from the whole image, and only then uses selective search to find ROIs on the resulting feature map, and finally classifies the ROIs and tightens the bounding boxes. It is much faster and takes about 2 seconds per image. A last incrementation on this basis is Faster R-CNN [Ren2015], which works similarly to the latter, but finds ROIs with a neural network instead of with selective search, which is very time-consuming. This allows for predicting an "objectness" score on each ROI, and for fitting the bounding boxes directly, and thus on avoiding the last regression step. It is even faster, and takes about 0.2 seconds per image. YOLO (standing for You Only Look Once) [Redmon2016] proposes another approach, and does not separate the steps of finding ROIs with classification. It divides the image into regions, and predicts both classes and bounding boxes for each region. For example, if there is a shoulder in a region, it will predict a "person" class, and a larger box in which this person is likely to fit. YOLO takes about 0.02 seconds per image (45 fps), and is thus able to run real time. However, it is not as accurate as the previous methods, especially on smaller objects. This being said, new versions are very frequently released (although not by the same authors), and the current YOLOv7 [Wang2022a] is both faster and more accurate than all previous approaches as it entirely reviews the whole network architecture to deal with all observed bottlenecks.

But again, in order to perform joint kinematics, one cannot just detect whole objects: precise keypoints need to be localized. Mask R-CNN [He2017] still predicts the bounding boxes and their class like Faster R-CNN does, but it also adds a small overhead in parallel, which predicts the shapes of masks overlaying the object in a pixel-to-pixel manner. Keypoints can be seen as a very small mask, and Mask R-CNN can also detect them in order to predict human pose estimation. In the next paragraph, only multi-person pose estimation models will be considered. Datasets, evaluation metrics, and comparison of results won't be detailed: see [Topham2021] for a comprehensive overview.

Two main approaches for multi-person 2D pose estimation coexist. The "top-down" one first detects bounding boxes around persons, and then finds keypoints inside each box. In the area of object detection methods, they are analogous to region-proposed methods such as the R-CNN suite, which propose ROIs and then find and classify objects. Conversely, the "bottom-up" approach first finds keypoints, and then groups them into persons. They are analogous to the single-shot object detection methods such as the YOLO suite, which first find small details, and then predict full-size objects. These approaches are nowadays almost as fast as the top-down ones, however their inference time does not increase with the amount of persons detected.

Mask R-CNN belongs to the first kind, as well as AlphaPose [Fang2017], which mostly differentiates from the latter by using a network predicting higher quality bounding boxes from inaccurate ones, in order to facilitate the task of the joint regressor. On the opposite, DeepCut and DeeperCut [Pishchulin2016, Insafutdinov2016], as well as DeepLabCut [Mathis2018, Lauer2022] upon which it is built, are bottom-up approaches. They find a large number of keypoint candidates, label them as hand, head, etc., and then select the best candidates and separate them into persons. Since they calculate every possible association between keypoints, this is very slow. OpenPose [Cao2019] uses a network which jointly predicts keypoint locations, and the connections between them (i.e., it also predicts limbs, which define a skeleton), and is much faster while still being accurate. OpenPifPaf [Kreiss2022] adds to it both temporal consistency across frames, and an intensity map for each keypoint instead of punctual locations (i.e., a further keypoint will have a lower intensity). This allows for better accuracy in low-resolution regime and in occluded images. YOLOv7 supports keypoint detection by integrating YOLO-Pose [Maji2022], and claims to be faster and more accurate than all other state-of-the-art methods. It brings together top-down and bottom-up approaches, and uses a single network predicting both bounding boxes and their corresponding poses. SLEAP [Pereira2022], which is built for training animal pose estimation models, implements both top-down and bottom-up approaches. In this context, top-down approaches are slightly more accurate, and considerably faster as long as few animals are in the scene.

Like all previously presented methods, OpenPose has been trained the COCO dataset [Lin2014]. However, OpenPose body_25 standard model provides foot keypoints, which are primordial in sports motion analysis. To do so, 6 more keypoints have been labeled for the feet on the COCO dataset before training. OpenPose also supports the single-network whole-body pose estimation network [Hidalgo2019], which has been trained in the same time on COCO+foot, MPII [Andriluka2014], and on Total Capture [Xiang2019] in order to provide hand, face, feet, and body keypoints in one single network. A submodel of it is body_25b, which provides body and foot keypoints as body_25 does (although in slightly different locations), and in addition decreases the number of false positives without hampering speed (Figure 2.6). In a similar way, AlphaPose provides a full-body model, trained on the Halpe dataset [Li2020]. Note that BlazePose also provides hand and feet keypoints, but as it is a single-person pose estimation model, it is not addressed here.

Figure 2.6: The body_25b OpenPose model is more accurate than the default body_25 one. As an example, the left knee is slightly misplaced on the default model. Keypoint definition and order also differ between both models.

2.2 3D reconstruction

Once the pose of an athlete is correctly detected, the next step is to obtain their 3D pose. While some approaches strive to infer 3D pose from a monocular video source, they are generally not considered sufficiently accurate, especially when body parts are occluded. It is, then, important to use several cameras, and to fuse their 2D pose estimation results to obtain more reliable 3D coordinates.

2.2.1 Pinhole camera model

camera coordinate system

2.2.2 Calibration

test

2.2.3 Triangulation

suite

2.3 3D joint kinematics

2.3.1 Physically consistent model

autre

2.3.2 Scaling

bref

2.3.3 Inverse kinematics

As opposed to forward kinematics
Compare with 2D angles between 3 points
Different methods (model based vs autres) for angles (cf mail starred)

3

Proposed solution: Pose2Sim Python package

We propose the Pose2Sim python package, as an alternative to the more usual marker-based motion capture methods.Pose2Sim stands for "OpenPose to OpenSim", as it uses OpenPose inputs (2D keypoints coordinates obtained from multiple videos) and leads to an OpenSim result (physically consistent full-body 3D joint angles). Code is available at `https://github.com/perfanalytics/pose2sim`.

This chapter is adapted from the article published in the Journal of Open Source Software: "Pose2Sim: An Open-source Python Package for multiview markerless kinematics" [Pagnon2022b].

Contents

3.1 Introduction to the workflow

Although some developments are relevant to both, specifics differ between medicine and the sports field. In this regard and as stated in the Statement of need, marker-based methods are not well suited for sports motion analysis [Colyer2018]. In sports, capture should not hinder the movement. Placing markers on the naked body takes time and is cumbersome, therefore markerless approaches are favored. Sports environments are usually much more challenging than lab settings: frequent occlusions, fast and unusual movements, and complex background make it important to resort to using multiple view points, from RGB rather than RGB-D cameras, processed with machine learning methods. Competition conditions are often fast-paced and congested, so a light-weight, fast, and easy to set up system is relevant. However, as coaches and athletes usually need a mere feedback rather than a definitive diagnosis, they don't need as thorough of an accuracy as physicians. Ideally, results should be given in real time, and they should be more visual than graphs of time series. Moreover, 3D kinematics are more relevant than 2D sagittal plane kinematics; and full-body analysis (including upper-limb) is desired.

We propose the Python package Pose2Sim [Pagnon2022b], which aims to deal with these constraints. It provides a framework for 3D markerless kinematics, as an alternative to the more usual marker-based motion capture methods. Pose2Sim stands for "OpenPose to OpenSim", as it uses OpenPose inputs (2D coordinates obtained from multiple videos) [Cao2019] and leads to an OpenSim result (full-body 3D joint angles) [Delp2007, Seth2018]. Pose2Sim is accessible at `https://github.com/perfanalytics/pose2sim`.

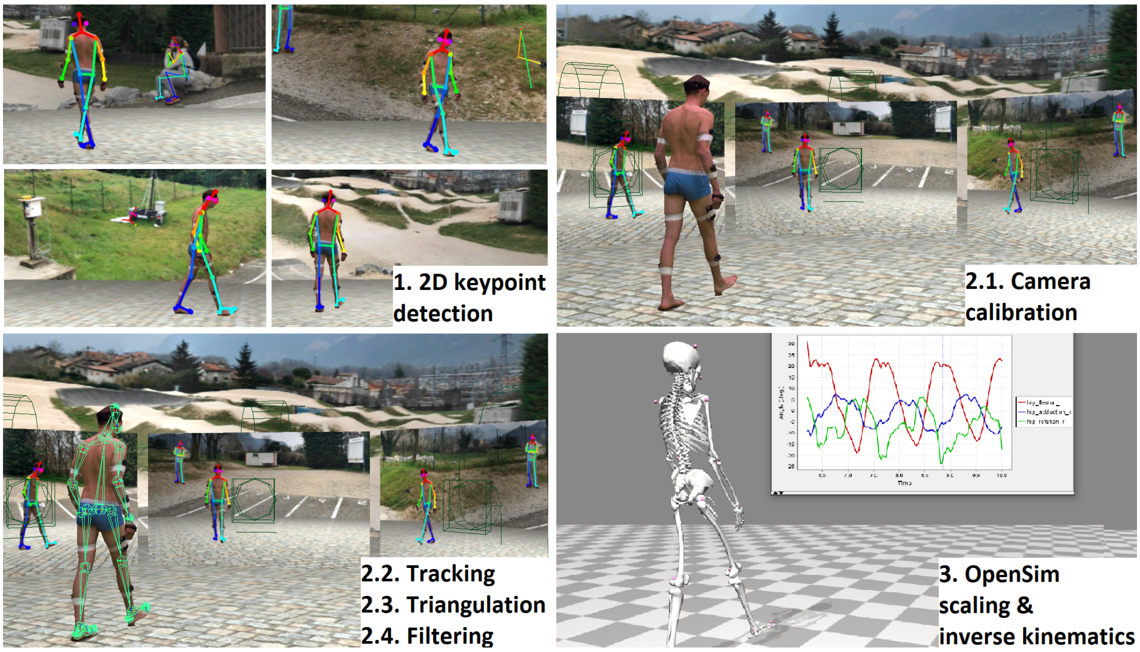


Figure 3.1: Pose2Sim full pipeline: (1) 2D keypoint detection; (2.1) Camera calibration; (2.1-2.4) Tracking of the person of interest, Triangulating of keypoint coordinates; and Filtering; (3) Constraining the 3D coordinates to an individually scaled, physically consistent OpenSim skeletal model.

The repository presents a framework which consists in (Figures 3.1):

1. Preliminary 2D joint coordinate detections from multiple videos, e.g. with OpenPose.
2. Pose2Sim core, including 4 customizable steps:
   2.1. Camera calibration.
   2.2. 2D tracking of the person of interest.
   2.3. 3D keypoint triangulation.
   2.4. 3D coordinate filtering.
3. Scaling a full-body skeleton to each individual subject, and computing inverse kinematics via OpenSim so as to obtain 3D joint angles.

Each task is easily customizable, and requires only moderate Python skills. The whole workflow runs from any video cameras, on any computer, equipped with any operating system (although OpenSim has to be compiled from source on Linux.) Pose2Sim has already been used and tested in a number of situations (walking, running, cycling, dancing, balancing, swimming, boxing), and published in peer-reviewed scientific publications assessing the quality of its code [Pagnon2022c], its robustness (see Chapter 4 on Robustness assessment) [Pagnon2021] and its accuracy (see Chapter 5 on Accuracy assessment) [Pagnon2022a]. Its results for inverse kinematics were deemed good when compared to marker-based ones, with errors generally below 4.0° across several activities, on both lower and on upper limbs. The combination of its ease of use, customizable parameters, and high robustness and accuracy makes it promising, especially for "in-the-wild" sports movement analysis.

3.2 Installation and demonstration

3.2.1 Installation

1. Install **OpenPose** (instructions here).
   Windows portable demo is enough.
2. Install **OpenSim 4.x** from there.
   Tested up to v4.4-beta on Windows. Has to be compiled from source on Linux (see there.
3. *Optional:* Install **Anaconda** or Miniconda.
   Open an Anaconda terminal and create a virtual environment by typing:
   ```
   conda create -n Pose2Sim python=3.8.8
   conda activate Pose2Sim
   ```
4. Install **Pose2Sim**
   If you don't use Anaconda, type `python -V` in terminal to make sure python>=3.6 is installed.
   - OPTION 1: *Quick install.* Type in terminal:
     ```
     pip install pose2sim
     ```
   - OPTION 2: *Build from source.* Open a terminal in the directory of your choice and clone the Pose2Sim repository:
     ```
     git clone https://gitlab.inria.fr/perfanalytics/pose2sim.git
     cd pose2sim
     pip install .
     ```

3.2.2 Demonstration Part-1: Build 3D TRC file on Python

This demonstration provides an example experiment of a person balancing on a beam, filmed with 4 calibrated cameras processed with OpenPose.

Open a terminal and check package location with `pip show pose2sim | grep Location`. Copy this path and go to the Demo folder with `cd <path>\pose2sim\Demo`. Type `python`, and test the following code (Figures 3.4):

```
from Pose2Sim import Pose2Sim
Pose2Sim.calibrateCams()
Pose2Sim.track2D()
Pose2Sim.triangulate3D()
Pose2Sim.filter3D()
```

You should obtain a plot of all the 3D coordinates trajectories (Figures 3.2). You can check the logs in `Demo\Users\logs.txt`. Results are stored as .trc files in the `Demo\pose-3d` directory (Figures 3.3). Note that when the functions are called without any argument, the Config file is searched in the default `Users\Config.toml` location. These parameters can be edited by the user.

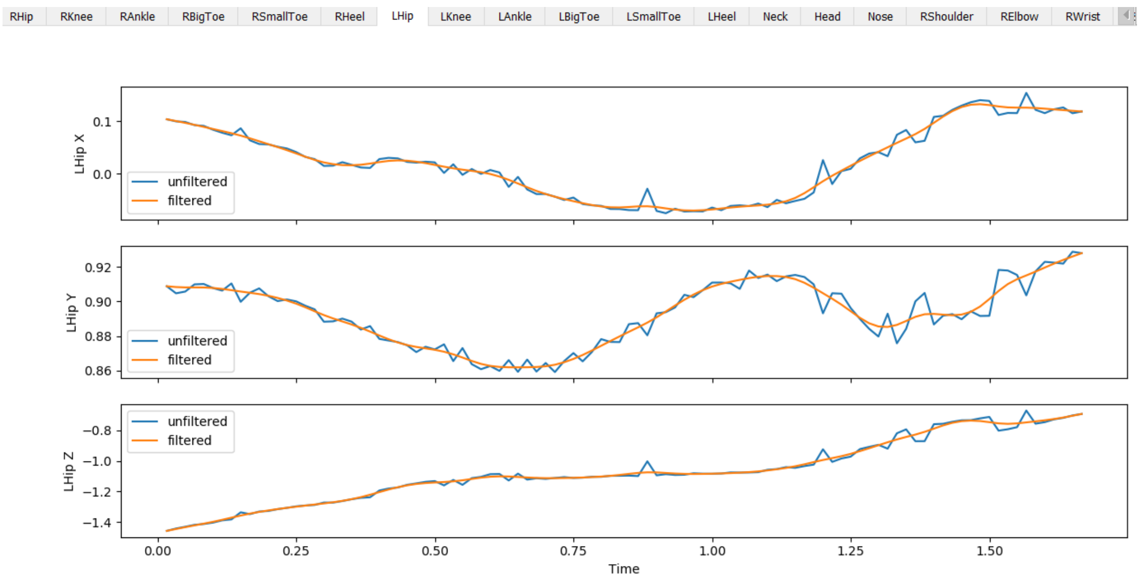


Figure 3.2: Filtering results, for each keypoint in a different tab.

PathFileType	4	(X/Y/Z)	Demo_0-100.trc								
DataRate	CameraRate	NumFrames	NumMarkers	Units	OrigDataRate	OrigDataStartFrame	OrigNumFrames				
60	60	100	21	m	60	0	100				
Frame#	Time	RHip			RKnee			RAnkle			RBigToe
		X1	Y1	Z1	X2	Y2	Z2	X3	Y3	Z3	X4
1	0.0166666667	-0.064972148	0.9015045551	-1.4005886926	-0.0396263662	0.4930973651	-1.4485228257	0.0437901401	0.1438754982	-1.5950846772	-0.0169084827
2	0.0333333333	-0.0740068294	0.9044249595	-1.3887505524	-0.0396716745	0.4930610544	-1.4481465416	0.039886297	0.1434935164	-1.5931576221	-0.0169371875
3	0.05	-0.0799400351	0.9088363315	-1.3905580361	-0.0372341964	0.4955765014	-1.4425663246	0.0424186998	0.1500265582	-1.5949272593	-0.0157499669
4	0.0666666667	-0.0834212999	0.9118448511	-1.3790501541	-0.0378442483	0.4986109124	-1.4352189865	0.041841984	0.1506211073	-1.5951903296	-0.0159322546
5	0.0833333333	-0.0821238866	0.910708592	-1.3705528594	-0.0415058215	0.4929167908	-1.4391550118	0.0368223321	0.1492766387	-1.5990002244	-0.0214821932
6	0.1	-0.0870228272	0.9113384284	-1.3568997099	-0.0434174115	0.4981952646	-1.4247995005	0.0306840105	0.1528813191	-1.5954295987	-0.0237343535
7	0.1166666667	-0.0920100974	0.9116316951	-1.3447088632	-0.0445856424	0.5002300425	-1.4213497807	0.0290451125	0.1540803887	-1.5892342384	-0.0245350272
8	0.1333333333	-0.0906673188	0.9161769285	-1.3309245886	-0.046053813	0.5073858348	-1.4072542077	0.0334937714	0.1591193395	-1.5866813244	-0.0225906144

Figure 3.3: An example .trc file of triangulated keypoint coordinates, directly usable in OpenSim.

```
In [6]: Pose2Sim.calibrateCams('User/Config.toml')

Calibrating cameras...
--> Residual (RMS) calibration errors for each camera are respectively [0.221, 0.235, 0.171, 0.191] px,
 which corresponds to [0.402, 0.445, 0.45, 0.505] mm.
Calibration file is stored at [redacted]
```

(a) Calibration can either be done from a checkerboard, or by simply converting a Qualisys calibration file. Calibration errors are computed and provided.

```
In [11]: Pose2Sim.track2D('User/Config.toml')

Tracking the person of interest for Demo, for frames 0 to 100.
100%|████████████████████████████████████████| 100/100 [00:00<00:00, 383.53it/s]
--> Mean reprojection error for Neck point on all frames is 12.3 px, which roughly corresponds to 22.4 mm.
--> In average, 0.01 cameras had to be excluded to reach the demanded 20 px error threshold.
Tracked json files are stored in [redacted]
```

(b) If several persons are detected in the scene, a tracking step can be carried out in order to make sure that the right person from each camera will be triangulated.

```
In [12]: Pose2Sim.triangulate3D('User/Config.toml')

Triangulation of 2D points for Demo, for frames 0 to 100.
D:\softs\github_david\Pose2Sim\Demo\calib-2d\Calib_qca.toml
100%|████████████████████████████████████████| 100/100 [00:02<00:00, 33.71it/s]
Mean reprojection error for RHip is 8.0 px (~ 0.015 m), reached with 0.99 excluded cameras.
Mean reprojection error for RKnee is 9.4 px (~ 0.017 m), reached with 0.61 excluded cameras.
Mean reprojection error for RAnkle is 10.8 px (~ 0.02 m), reached with 0.1 excluded cameras.
Mean reprojection error for RBigToe is 10.9 px (~ 0.02 m), reached with 0.57 excluded cameras.
Mean reprojection error for RSmallToe is 10.6 px (~ 0.019 m), reached with 0.44 excluded cameras.
Mean reprojection error for RHeel is 11.1 px (~ 0.02 m), reached with 0.31 excluded cameras.
Mean reprojection error for LHip is 8.8 px (~ 0.016 m), reached with 0.83 excluded cameras.
Mean reprojection error for LKnee is 10.6 px (~ 0.019 m), reached with 0.8 excluded cameras.
Mean reprojection error for LAnkle is 12.3 px (~ 0.022 m), reached with 0.15 excluded cameras.
Mean reprojection error for LBigToe is 10.2 px (~ 0.019 m), reached with 0.33 excluded cameras.
Mean reprojection error for LSmallToe is 11.2 px (~ 0.02 m), reached with 0.46 excluded cameras.
Mean reprojection error for LHeel is 10.6 px (~ 0.019 m), reached with 0.38 excluded cameras.
Mean reprojection error for Neck is 11.1 px (~ 0.02 m), reached with 0.17 excluded cameras.
Mean reprojection error for Head is 9.8 px (~ 0.018 m), reached with 0.56 excluded cameras.
Mean reprojection error for Nose is 8.4 px (~ 0.015 m), reached with 1.95 excluded cameras.
Mean reprojection error for RShoulder is 9.4 px (~ 0.017 m), reached with 0.61 excluded cameras.
Mean reprojection error for RElbow is 9.0 px (~ 0.016 m), reached with 0.63 excluded cameras.
Mean reprojection error for RWrist is 9.7 px (~ 0.018 m), reached with 0.49 excluded cameras.
Mean reprojection error for LShoulder is 10.2 px (~ 0.019 m), reached with 0.5 excluded cameras.
Mean reprojection error for LElbow is 12.1 px (~ 0.022 m), reached with 0.39 excluded cameras.
Mean reprojection error for LWrist is 11.6 px (~ 0.021 m), reached with 0.38 excluded cameras.
--> Mean reprojection error for all points on all frames is 10.3 px, which roughly corresponds to 18.8 mm.
--> Cameras were excluded if likelihood was below 0.3 and if the reprojection error was above 15 px.
In average, 0.55 cameras had to be excluded to reach these thresholds.
3D coordinates are stored at [redacted]
```

(c) The triangulation is weighted by the OpenPose likelihood, and constrained by some thresholds defined in the Config.toml file. If these constraints are not met, e.g., if the reprojection error is too large or if the likelihood of a keypoint is too low, one or several cameras are excluded. The mean reprojection error and the number of cameras that have been excluded to meet the constraints is printed, for each keypoints.

```
In [13]: Pose2Sim.filter3D('User/Config.toml')

Filtering 3D coordinates for Demo, for frames 0 to 100.
--> Filter type: Butterworth low-pass. Order 4, Cut-off frequency 6 Hz.
Filtered 3D coordinates are stored at [redacted]
```

(d) Triangulated data can be filtered, either with a low-pass Butterworth filter or with other types, and parameters can be adjusted.

Figure 3.4: First steps of Pose2Sim pipeline in Python. Calibration can either be done from a checkerboard, or by simply converting a Qualisys calibration file. Note that the functions can be used without any arguments if the Config.toml file is left in the default location.

3.2.3 Demonstration Part-2: Obtain 3D joint angles with OpenSim

In the same vein as we would do with marker-based kinematics, the model first needs to be scaled to each individual, and then inverse kinematics can be performed (Figures 3.5).

Scaling:

1. Open OpenSim.
2. Open the provided `Model_Pose2Sim_Body25b.osim` model from `pose2sim/Demo/opensim`. (File $\mapsto$ Open Model)
3. Load the provided `Scaling_Setup_Pose2Sim_Body25b.xml` scaling file from `pose2sim/Demo/opensim`. (Tools $\mapsto$ Scale model $\mapsto$ Load)
4. Run. You should see your skeletal model take the static pose.

Inverse kinematics

1. Load the provided `IK_Setup_Pose2Sim_Body25b.xml` scaling file from `pose2sim/Demo/opensim`. (Tools $\mapsto$ Inverse kinematics $\mapsto$ Load)
2. Run. You should see your skeletal model move in the Vizualizer window.

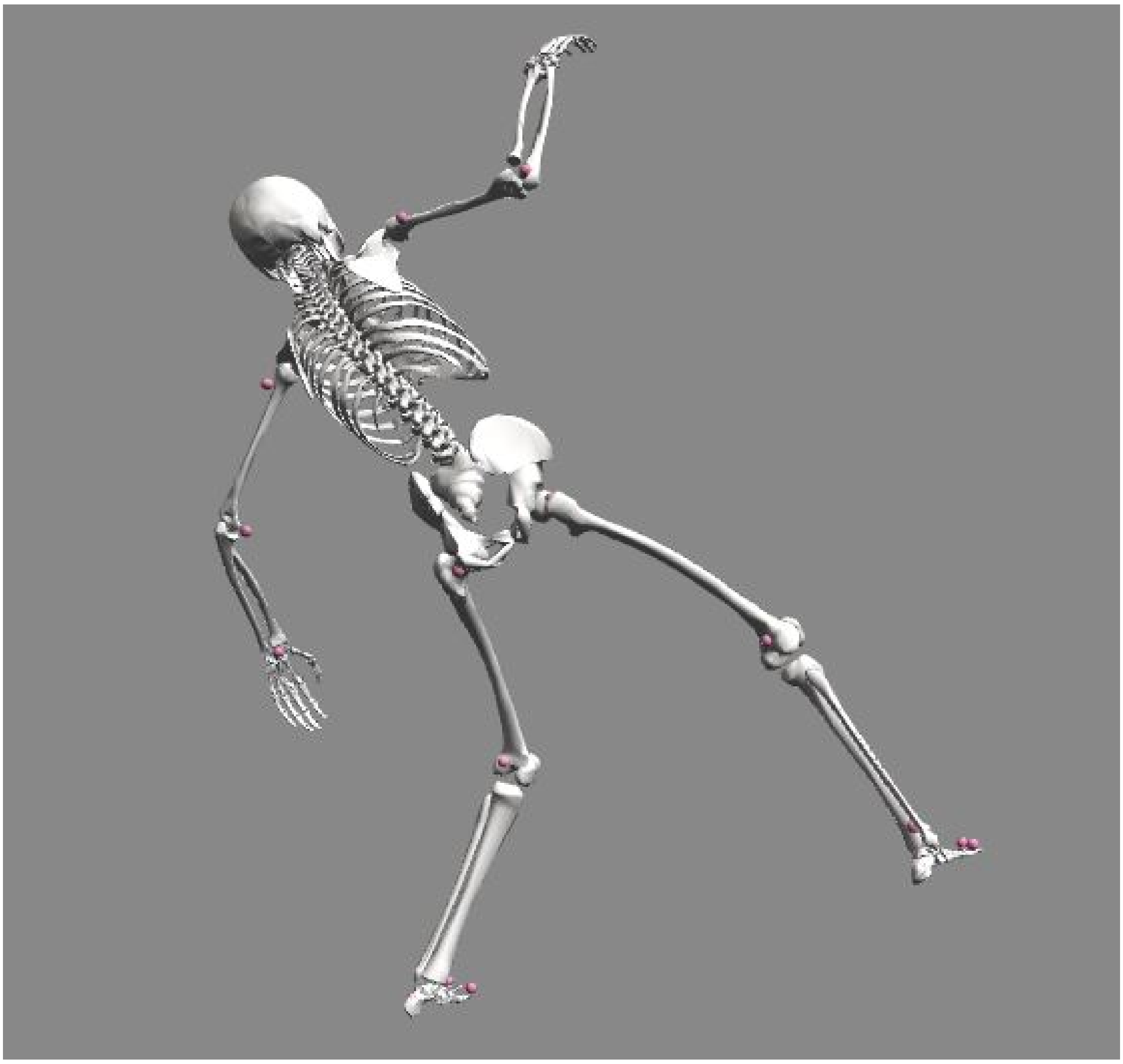


Figure 3.5: At the end of the demonstration, you should have a skeleton balancing on a beam in OpenSim.

3.3 Method details

3.3.1 Project

Pose2Sim is meant to be as fully and easily configurable as possible, by editing the `User/Config.toml` file. Optional tools are also provided for extending its usage (Figures 3.6). First of all, the user can specify the project path and folder names, the video frame rate, and the range of analyzed frames.

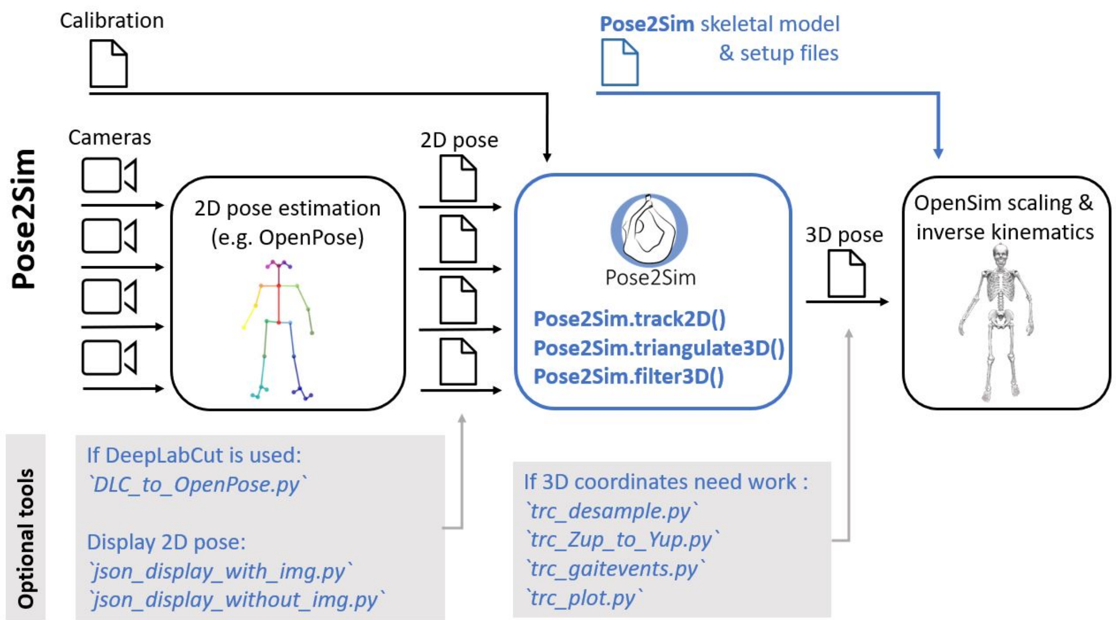


Figure 3.6: The Pose2Sim workflow, along with some optional utilities provided in the package.

3.3.2 2D keypoint detection

The interest in deep-learning pose estimation neural networks has been growing fast since 2015 [Zheng2022], which makes it now possible to collect accurate and reliable data without the use of physical markers. OpenPose, for example, is a widespread open-source software which provides 2D joint coordinate estimates from videos.

As feet are usually needed in sports kinematic analysis, this is the software we recommend. Indeed, unlike other potentially faster and more accurate networks, OpenPose comes with {body + feet} models such as BODY_25 or BODY_25B, as well as with a {body + feet + hands + face} one called BODY_135 [Hidalgo2019]. Note that BlazePose does provide feet and hand keypoints, but it is limited to single person analysis. This is rarely suitable in sports conditions, since there are often people in the background.

This being said, the user can choose any deep-learning pose estimation network. This choice will affect how keypoint indices will be mapped to model markers in OpenSim, corresponding to anatomical landmarks or joint centers. The OpenPose BODY_25B experimental model is recommended, as it is as fast as the standard BODY_25 model, while being more accurate [Hidalgo2019]. Only 21 of the 25 keypoints detected are tracked, since eye and ear keypoints would be redundant in the determination of the head orientation.

The OpenPose BODY_25, BODY_25B, BODY_135, COCO, and MPII models are fully supported. The AlphaPose COCO and HALPE models are also supported, as well as the BlazePose one. COCO and MPII model are the ones generally used by other networks such OpenPifPaf [Kreiss2022], YOLO-pose [Maji2022, Wang2022a], and others, which means that they are also supported. It is also possible to build custom skeletons in the `skeleton.py` file. They will be triangulated, but the user will need to build an OpenSim model and set the keypoints in the right place before being able to perform inverse kinematics.

Two optional standalone scripts are also provided if the user desires a visual display of the resulting 2D pose estimation (Figures 3.6).

3.3.3 Camera calibration

3.3.4 Tracking the person of interest

3.3.5 Triangulating

3.3.6 Filtering and other operations

3.3.7 OpenSim scaling and inverse kinematics

A full-body OpenSim [Delp2007, Seth2018] skeletal model with OpenPose keypoints is provided, as well as scaling and inverse kinematics setup files.

OpenSim is another widespread open-source software which helps compute 3D joint angles, usually from marker coordinates. It lets scientists define a detailed musculoskeletal model, scale it to individual subjects, and perform inverse kinematics with customizable biomechanical constraints. It provides other features such as net calculation of joint moments or resolution of individual muscle forces, although this is beyond the scope of our contribution.

3.4 Limitations and perspectives

3.5 Helper functions and visualization tools

Not real time nor visual feedback. Maya MoCap Bath MPP2SOS (not free nor opensource) [Barreto2022]

4
Robustness assessment

Résumé du chapitre possible ici.

Contents

4.1 Introduction

4.1.1 Robustness definition

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1.2 Assessing robustness

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2 Methods

4.2.1 Experimental setup

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2.2 Participant and protocol

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2.3 Challenging robustness

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet

and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2.4 Statistical analysis

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.3 Results

4.3.1 Data collection and 2D pose estimation

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.3.2 Pose2Sim tracking, triangulation, and filtering

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.3.3 Relevance, repeatability and robustness of angles Results

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.4 Discussion

4.4.1 Pose2Sim

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.4.2 Relevance, repeatability and robustness

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.4.3 Limits and perspectives

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5
Accuracy assessment

Résumé du chapitre possible ici.

Contents

5.1 Introduction

5.1.1 State of the art

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.1.2 Assessing accuracy

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.2 Methods

5.2.1 Data collection

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.2.2 Markerless analysis

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.2.3 Marker-based analysis

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet

and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.2.4 Statistical analysis

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.3 Results

5.3.1 Concurrent validation

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.3.2 Comparison with other systems

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.4 Discussion

5.4.1 Strengths of Pose2Sim and of markerless kinematic

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.4.2 Limits and perspectives

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.5 Conclusions

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6

Application to boxing, using action cameras

Pose2Sim in suboptimal conditions:

This chapter is adapted from the poster presented at the congress of the Europen College of Sport Science (ECSS): "A 3D markerless protocol with action cameras – Key performance indicators in boxing" [Pagnon2022c].

Contents

6.1 Objectives

6.1.1 Key Performance Indicators in boxing

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.1.2 Limits of research-grade systems in competitions

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.1.3 Objectives

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2 Methods

6.2.1 4 conditions

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2.2 Pose-calibration on ring dimensions

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet

and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2.3 Post-synchronization on 2D movement speeds

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2.4 GoPro spatio-temporal base into Qualysis’

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2.5 Statistical analysis

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.3 Results

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.4 Discussion

6.4.1 Equipment and protocol vs. pose estimation model

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”?

Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.4.2 Pros and cons of different systems

Auto-calibration with person?
Cloud computing?
Temporal consistency?
Shape information for less cameras?

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7

Application to BMX racing, capturing jointly pilot and bike

Résumé du chapitre possible ici.

Contents

7.1 Introduction

7.1.1 The start in BMX racing

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.2 Methods

7.2.1 Material and protocol

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.2.2 Pilot inverse kinematics

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.2.3 Bike inverse kinematics

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.2.4 Joined pilot and bike inverse kinematics

Marche pas avec nos qualités de vidéo : simulations

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet

and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.3 Results

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.4 Discussion

7.4.1 On these data

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.4.2 Limits and perspectives

Mathis2020 Principles, pitfalls and perspectives

General conclusion

Conclusion here.

Bibliography

[Ahmad2013] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M. Khairi et Vijayabaskar Kasi. *Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications*. International Journal of Signal Processing Systems, pages 256–262, 2013.

[Aizerman1964] Mark A Aizerman. *Theoretical foundations of the potential function method in pattern recognition learning*. Automation and remote control, vol. 25, pages 821–837, 1964.

[Andriluka2014] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler et Bernt Schiele. *2D Human Pose Estimation: New Benchmark and State of the Art Analysis*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

[Atha1984] J Atha. *Current techniques for measuring motion*. Applied ergonomics, vol. 15, no. 4, pages 245–257, 1984.

[Baker2007] Richard Baker. *The history of gait analysis before the advent of modern computers*. Gait and Posture, vol. 26, no. 3, pages 331–342, 9 2007.

[Bao2022] Yiming Bao, Xu Zhao et Dahong Qian. *FusePose: IMU-Vision Sensor Fusion in Kinematic Space for Parametric Human Pose Estimation*. arXiv preprint arXiv:2208.11960, 2022.

[Baral2018] Chitta Baral, Olac Fuentes et Vladik Kreinovich. *Why deep neural networks: a possible theoretical explanation*. In Constraint programming and decision making: Theory and applications, pages 1–5. Springer, 2018.

[Barreto2022] Carlos Barreto. *Mocap MPP2SOS*, 2022.

[Bazarevsky2020] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang et Matthias Grundmann. *Blazepose: On-device realtime body pose tracking*. arXiv preprint arXiv:2006.10204, 2020.

[Benoit2015] D. L. Benoit, M. Damsgaard et M. S. Andersen. *Surface marker cluster translation, rotation, scaling and deformation: Their contribution to soft tissue artefact and impact on knee joint kinematics*. Journal of Biomechanics, vol. 48, no. 10, pages 2124–2129, 7 2015.

[Boser1992] Bernhard E Boser, Isabelle M Guyon et Vladimir N Vapnik. *A training algorithm for optimal margin classifiers*. In Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152, 1992.

[Bouwmans2019] Thierry Bouwmans, Sajid Javed, Maryam Sultana et Soon Ki Jung. *Deep neural network concepts for background subtraction: A systematic review and comparative evaluation*. Neural Networks, vol. 117, pages 8–66, 2019.

[Bradski2000] G. Bradski. *The OpenCV Library*. Dr. Dobb's Journal of Software Tools, 2000.

[Bridgeman2019] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut et Adrian Hilton. *Multi-Person 3D Pose Estimation and Tracking in Sports*. pages 2487–2496, Long Beach, CA, USA, 6 2019. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE.

[Cao2019] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei et Yaser Sheikh. *OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields*. IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 1, pages 172–186, 2019.

[Cappozzo1995] A Cappozzo, F Catani, U Della Croce et A Leardini. *Position and orientation in space of bones during movement: anatomical frame definition and determination*. Clinical Biomechanics, vol. 10, no. 4, pages 171–178, 6 1995.

[Carraro2017] Marco Carraro, Matteo Munaro, Jeff Burke et Emanuele Menegatti. *Real-time marker-less multi-person 3D pose estimation in RGB-Depth camera networks*. arXiv:1710.06235 [cs], 10 2017. arXiv: 1710.06235.

[Ceseracciu2014] Elena Ceseracciu, Zimi Sawacha et Claudio Cobelli. *Comparison of Markerless and Marker-Based Motion Capture Technologies through Simultaneous Data Collection during Gait: Proof of Concept*. PLoS ONE, vol. 9, no. 3, page e87640, 3 2014.

[Chambers2015] Ryan Chambers, Tim J Gabbett, Michael H Cole et Adam Beard. *The use of wearable microsensors to quantify sport-specific movements*. Sports medicine, vol. 45, no. 7, pages 1065–1081, 2015.

[Chen2020] Yucheng Chen, Yingli Tian et Mingyi He. *Monocular human pose estimation: A survey of deep learning-based methods*. Computer Vision and Image Understanding, vol. 192, page 102897, 3 2020.

[Chicco2017] Davide Chicco. *Ten quick tips for machine learning in computational biology*. BioData mining, vol. 10, no. 1, pages 1–17, 2017.

[Choppin2013] Simon Choppin et Jonathan Wheat. *The potential of the Microsoft Kinect in sports analysis and biomechanics*. Sports Technology, vol. 6, no. 2, pages 78–85, 5 2013.

[Chu2021] Hau Chu, Jia-Hong Lee, Yao-Chih Lee, Ching-Hsien Hsu, Jia-Da Li et Chu-Song Chen. *Part-Aware Measurement for Robust Multi-View Multi-Human 3D Pose Estimation and Tracking*. page 10, 2021.

[Cireşan2012] Dan Cireşan, Ueli Meier, Jonathan Masci et Jürgen Schmidhuber. *Multi-column deep neural network for traffic sign classification*. Neural networks, vol. 32, pages 333–338, 2012.

[Colombel2020] Jessica Colombel, Vincent Bonnet, David Daney, Raphael Dumas, Antoine Seilles et François Charpillet. *Physically Consistent Whole-Body Kinematics Assessment Based on an RGB-D Sensor. Application to Simple Rehabilitation Exercises*. Sensors, vol. 20, no. 10, page 2848, 5 2020.

[Colyer2018] Steffi L Colyer, Murray Evans, Darren P Cosker et Aki IT Salo. *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, vol. 4, no. 1, pages 1–15, 2018.

[Cronin2019] Neil J. Cronin, Timo Rantalainen, Juha P. Ahtiainen, Esa Hynynen et Ben Waller. *Markerless 2D kinematic analysis of underwater running: A deep learning approach*. Journal of Biomechanics, vol. 87, pages 75–82, 4 2019.

[Cronin2021] Neil J. Cronin. *Using deep neural networks for kinematic analysis: challenges and opportunities*. Journal of Biomechanics, page 110460, 5 2021.

[Cybenko1989] George Cybenko. *Approximation by superpositions of a sigmoidal function*. Mathematics of control, signals and systems, vol. 2, no. 4, pages 303–314, 1989.

[Dalal2005] Navneet Dalal et Bill Triggs. *Histograms of oriented gradients for human detection*. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), volume 1, pages 886–893. Ieee, 2005.

[D'Antonio2021] Erika D'Antonio, Juri Taborri, Ilaria Mileti, Stefano Rossi et Fabrizio Patane. *Validation of a 3D Markerless System for Gait Analysis based on OpenPose and Two RGB Webcams*. IEEE Sensors Journal, pages 1–1, 2021.

[della Croce1999] U. della Croce, A. Cappozzo et D. C. Kerrigan. *Pelvis and lower limb anatomical landmark calibration precision and its propagation to bone geometry and joint angles*. Medical and Biological Engineering and Computing, vol. 37, no. 2, pages 155–161, 3 1999.

[Delp2007] Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman et Darryl G Thelen. *OpenSim: open-source software to create and analyze dynamic simulations of movement*. IEEE transactions on biomedical engineering, vol. 54, no. 11, pages 1940–1950, 2007.

[Deng2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li et L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. 2009.

[Dong2019] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao et Xiaowei Zhou. *Fast and Robust Multi-Person 3D Pose Estimation From Multiple Views*. pages 7784–7793, Long Beach, CA, USA, 6 2019. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE.

[Dong2020] Junting Dong, Qing Shuai, Yuanqing Zhang, Xian Liu, Xiaowei Zhou et Hujun Bao. *Motion Capture from Internet Videos*. In Andrea Vedaldi, Horst Bischof, Thomas Brox et Jan-Michael Frahm, editeurs, Computer Vision – ECCV 2020, volume 12347, pages 210–227. Springer International Publishing, Cham, 2020.

[Drazan2021] John F. Drazan, William T. Phillips, Nidhi Seethapathi, Todd J. Hullfish et Josh R. Baxter. *Moving outside the lab: Markerless motion capture accurately quantifies sagittal plane kinematics during the vertical jump*. Journal of Biomechanics, vol. 125, page 110547, 8 2021.

[Ershadi-Nasab2021] Sara Ershadi-Nasab, Shohreh Kasaei et Esmaeil Sanaei. *Uncalibrated multi-view multiple humans association and 3D pose estimation by adversarial learning*. Multimedia Tools and Applications, vol. 80, no. 2, pages 2461–2488, 1 2021.

[Fang2017] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai et Cewu Lu. *RMPE: Regional Multi-person Pose Estimation*. pages 2353–2362, Venice, 10 2017. 2017 IEEE International Conference on Computer Vision (ICCV), IEEE.

[Fernández-González2020] Pilar Fernández-González, Aikaterini Koutsou, Alicia Cuesta-Gómez, María Carratalá-Tejada, Juan Carlos Miangolarra-Page et Francisco Molina-Rueda. *Reliability of kinovea® software and agreement with a three-dimensional motion system for gait analysis in healthy subjects*. Sensors, vol. 20, no. 11, page 3154, 2020.

[Fisch2020] Martin Fisch et Ronald Clark. *Orientation Keypoints for 6D Human Pose Estimation*. arXiv:2009.04930 [cs], 9 2020. arXiv: 2009.04930.

[Geelen2021] Jinne E Geelen, Mariana P Branco, Nick F Ramsey, Frans CT Van Der Helm, Winfred Mugge et Alfred C Schouten. *MarkerLess Motion Capture: ML-MoCap, a low-cost modular multi-camera setup*. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pages 4859–4862. IEEE, 2021.

[Girshick2014] Ross Girshick, Jeff Donahue, Trevor Darrell et Jitendra Malik. *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, 2014.

[Girshick2015] Ross Girshick. *Fast r-cnn*. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[Goodfellow2016] Ian Goodfellow, Yoshua Bengio et Aaron Courville. Deep learning. MIT Press, 2016. http://www.deeplearningbook.org.

[Gorton2009] George E. Gorton, David A. Hebert et Mary E. Gannotti. *Assessment of the kinematic variability among 12 motion analysis laboratories*. Gait and Posture, vol. 29, no. 3, pages 398–402, 4 2009.

[Guo2022] Jiamin Guo, Qin Zhang, Hui Chai et Yibin Li. *Obtaining lower-body Euler angle time series in an accurate way using depth camera relying on Optimized Kinect CNN*. Measurement, vol. 188, page 110461, 2022.

[Han2013] Jungong Han, Ling Shao, Dong Xu et Jamie Shotton. *Enhanced Computer Vision With Microsoft Kinect Sensor: A Review*. IEEE Transactions on Cybernetics, vol. 43, no. 5, pages 1318–1334, 10 2013. event: IEEE Transactions on Cybernetics.

[Haralabidis2020] Nicos Haralabidis, David John Saxby, Claudio Pizzolato, Laurie Needham, Dario Cazzola et Clare Minahan. *Fusing Accelerometry with Videography to Monitor the Effect of Fatigue on Punching Performance in Elite Boxers*. Sensors (Basel, Switzerland), vol. 20, no. 20, 10 2020.

[Hartley1997] Richard I. Hartley et Peter Sturm. *Triangulation*. Computer Vision and Image Understanding, vol. 68, no. 2, pages 146–157, 11 1997.

[Hawkins2004] Douglas M Hawkins. *The problem of overfitting*. Journal of chemical information and computer sciences, vol. 44, no. 1, pages 1–12, 2004.

[He2017] Kaiming He, Georgia Gkioxari, Piotr Dollár et Ross Girshick. *Mask r-cnn*. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[He2020] Yihui He, Rui Yan, Katerina Fragkiadaki et Shoou-I Yu. *Epipolar Transformers*. pages 7776–7785. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6 2020. ISSN: 2575-7075.

[Hidalgo2019] Gines Hidalgo, Yaadhav Raaj, Haroon Idrees, Donglai Xiang, Hanbyul Joo, Tomas Simon et Yaser Sheikh. *Single-network whole-body pose estimation*. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6982–6991, 2019.

[Hofmann2008] Thomas Hofmann, Bernhard Schölkopf et Alexander J Smola. *Kernel methods in machine learning*. The annals of statistics, vol. 36, no. 3, pages 1171–1220, 2008.

[Insafutdinov2016] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka et Bernt Schiele. *Deepercut: A deeper, stronger, and faster multi-person pose estimation model*. In European conference on computer vision, pages 34–50. Springer, 2016.

[Iskakov2019] Karim Iskakov, Egor Burkov, Victor Lempitsky et Yury Malkov. *Learnable Triangulation of Human Pose*. pages 7717–7726, Seoul, Korea (South), 10 2019. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE.

[Johnston2019] William Johnston, Martin O'Reilly, Rob Argent et Brian Caulfield. *Reliability, validity and utility of inertial sensor systems for postural control assessment in sport science and medicine applications: a systematic review*. Sports Medicine, vol. 49, no. 5, pages 783–818, 2019.

[Kanko2021a] Robert M. Kanko, Elise Laende, W. Scott Selbie et Kevin J. Deluzio. *Inter-session repeatability of markerless motion capture gait kinematics*. Journal of Biomechanics, vol. 121, page 110422, 5 2021.

[Kanko2021b] Robert M. Kanko, Elise K. Laende, Elysia M. Davis, W. Scott Selbie et Kevin J. Deluzio. *Concurrent assessment of gait kinematics using marker-based and markerless motion capture*. Journal of Biomechanics, page 110665, 8 2021.

[Karashchuk2021] Pierre Karashchuk, Katie L Rupp, Evyn S Dickinson, Sarah Walling-Bell, Elischa Sanders, Eiman Azim, Bingni W Brunton et John C Tuthill. *Anipose: a toolkit for robust markerless 3D pose estimation*. Cell reports, vol. 36, no. 13, page 109730, 2021.

[Kidziński2020] Łukasz Kidziński, Bryan Yang, Jennifer L. Hicks, Apoorva Rajagopal, Scott L. Delp et Michael H. Schwartz. *Deep neural networks enable quantitative movement analysis using single-camera videos*. Nature Communications, vol. 11, no. 1, page 4054, 12 2020.

[Kreiss2022] Sven Kreiss, Lorenzo Bertoni et Alexandre Alahi. *OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association*. IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pages 13498–13511, 2022.

[Krizhevsky2017] Alex Krizhevsky, Ilya Sutskever et Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks*. Communications of the ACM, vol. 60, no. 6, pages 84–90, 2017.

[Labuguen2020] Rollyn T. Labuguen, Wally Enrico M. Ingco, Salvador Blanco Negrete, Tonan Kogami et Tomohiro Shibata. *Performance Evaluation of Markerless 3D Skeleton Pose Estimates with Pop Dance Motion Sequence*. Rapport technique, 4 2020. DOI: 10.1101/2020.04.15.010702.

[Lauer2022] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Steffen Schneider, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng*et al.* *Multi-animal pose estimation, identification and tracking with DeepLabCut*. Nature Methods, vol. 19, no. 4, pages 496–504, 2022.

[Leboeuf2019] F. Leboeuf, J. Reay, R. Jones et M. Sangeux. *The effect on conventional gait model kinematics and kinetics of hip joint centre equations in adult healthy gait*. Journal of Biomechanics, vol. 87, pages 167–171, 4 2019.

[LeCun1998] Yann LeCun, Léon Bottou, Yoshua Bengio et Patrick Haffner. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, vol. 86, no. 11, pages 2278–2324, 1998.

[Li2019] Zongmian Li, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard et Josef Sivic. *Estimating 3D Motion and Forces of Person-Object Interactions From Monocular Video*. pages 8632–8641, Long Beach, CA, USA, 6 2019. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE.

[Li2020] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen et Cewu Lu. *Pastanet: Toward human activity knowledge engine*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 382–391, 2020.

[Liao2020] Rijun Liao, Shiqi Yu, Weizhi An et Yongzhen Huang. *A model-based gait recognition method with body pose and human prior knowledge*. Pattern Recognition, vol. 98, page 107069, 2 2020.

[Lin2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár et C Lawrence Zitnick. *Microsoft coco: Common objects in context*. In European conference on computer vision, pages 740–755. Springer, 2014.

[Liu2022a] Kang Liu, Lingling Chen, Liang Xie, Jian Yin, Shuwei Gan, Ye Yan et Erwei Yin. *Auto calibration of multi-camera system for human pose estimation*. IET Computer Vision, 2022.

[Liu2022b] Pin-Ling Liu et Chien-Chi Chang. *Simple method integrating OpenPose and RGB-D camera for identifying 3D body landmark locations in various postures*. International Journal of Industrial Ergonomics, vol. 91, page 103354, 2022.

[Liu2022c] Wu Liu et Tao Mei. *Recent Advances of Monocular 2D and 3D Human Pose Estimation: A Deep Learning Perspective*. ACM Comput. Surv., mar 2022.

[Loper2015] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll et Michael J. Black. *SMPL: a skinned multi-person linear model*. ACM Transactions on Graphics, vol. 34, no. 6, pages 1–16, 11 2015.

[López-Muñoz2006] Francisco López-Muñoz, Jesús Boya et Cecilio Alamo. *Neuron theory, the cornerstone of neuroscience, on the centenary of the Nobel Prize award to Santiago Ramón y Cajal*. Brain research bulletin, vol. 70, no. 4-6, pages 391–405, 2006.

[Louis2022] Nathan Louis, Tylan N. Templin, Travis D. Eliason, Daniel P. Nicolella et Jason J. Corso. *Learning to Estimate External Forces of Human Motion in Video*. no. arXiv:2207.05845, Jul 2022.

[Lu2015] Chaochao Lu et Xiaoou Tang. *Surpassing human-level face verification performance on LFW with GaussianFace*. In Twenty-ninth AAAI conference on artificial intelligence, 2015.

[Maji2022] Debapriya Maji, Soyeb Nagori, Manu Mathew et Deepak Poddar. *YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2637–2646, 2022.

[Mathis2018] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis et Matthias Bethge. *DeepLabCut: markerless pose estimation of user-defined* *body parts with deep learning*. Nature Neuroscience, vol. 21, no. 9, pages 1281–1289, 9 2018.

[McCulloch1943] Warren S McCulloch et Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, vol. 5, no. 4, pages 115–133, 1943.

[Mehta2020] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll et Christian Theobalt. *XNect: real-time multi-person 3D motion capture with a single RGB camera*. ACM Transactions on Graphics, vol. 39, no. 4, page 82:82:1–82:82:17, 7 2020.

[Miller1980] Norman R. Miller, Robert Shapiro et Thomas M. McLaughlin. *A technique for obtaining spatial kinematic parameters of segments of biomechanical systems from cinematographic data*. Journal of Biomechanics, vol. 13, no. 7, pages 535–547, 1 1980.

[Minaee2021] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz et Demetri Terzopoulos. *Image segmentation using deep learning: A survey*. IEEE transactions on pattern analysis and machine intelligence, 2021.

[Minsky1969] Marvin Minsky et Seymour Papert. *Perceptrons: An introduction to computational geometry*. Cambridge tiass., HIT, vol. 479, page 480, 1969.

[Mündermann2006] Lars Mündermann, Stefano Corazza et Thomas P. Andriacchi. *The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications*. Journal of NeuroEngineering and Rehabilitation, vol. 3, no. 1, page 6, 3 2006.

[Nakano2019] Nobuyasu Nakano, Tetsuro Sakura, Kazuhiro Ueda, Leon Omura, Arata Kimura, Yoichi Iino, Senshi Fukashiro et Shinsuke Yoshioka. *Evaluation of 3D markerless motion capture accuracy using OpenPose with multiple video cameras*. Rapport technique, 11 2019. DOI: 10.1101/842492.

[Needham2021] Laurie Needham, Murray Evans, Darren P Cosker, Logan Wade, Polly M McGuigan, James L Bilzon et Steffi L Colyer. *The accuracy of several pose estimation methods for 3D joint centre localisation*. Scientific reports, vol. 11, no. 1, pages 1–11, 2021.

[Novikoff1963] Albert B Novikoff. *On convergence proofs for perceptrons*. Rapport technique, STANFORD RESEARCH INST MENLO PARK CA, 1963.

[Pagliari2015] Diana Pagliari et Livio Pinto. *Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors*. Sensors, vol. 15, no. 11, pages 27569–27589, 2015.

[Pagnon2021] David Pagnon, Mathieu Domalain et Lionel Reveret. *Pose2Sim: An End-to-End Workflow for 3D Markerless Sports Kinematics—Part 1: Robustness*. Sensors, vol. 21, no. 19, 2021.

[Pagnon2022a] David Pagnon, Mathieu Domalain et Lionel Reveret. *Pose2Sim: An End-to-End Workflow for 3D Markerless Sports Kinematics—Part 2: Accuracy*. Sensors, vol. 22, no. 7, 2022.

[Pagnon2022b] David Pagnon, Mathieu Domalain et Lionel Reveret. *Pose2Sim: An open-source Python package for multiview markerless kinematics*. Journal of Open Source Software, vol. 7, no. 77, page 4362, 2022.

[Pagnon2022c] David Pagnon, Mathieu Domalain, Thomas Robert, Bhrigu-Kumar Lahkar, Issa Moussa, Guillaume Saulière, Thibault Goyallon et Lionel Reveret. *A 3D markerless protocol with action cameras – Key performance indicators in boxing*. 2022. Poster.

[Pan2009] Sinno Jialin Pan et Qiang Yang. *A survey on transfer learning*. IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pages 1345–1359, 2009.

[Pereira2022] Talmo D Pereira, Nathaniel Tabris, Arie Matsliah, David M Turner, Junyu Li, Shruthi Ravindranath, Eleni S Papadoyannis, Edna Normand, David S Deutsch, Z Yan Wang*et al.* *SLEAP: A deep learning system for multi-animal pose tracking*. Nature methods, vol. 19, no. 4, pages 486–495, 2022.

[Pishchulin2016] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler et Bernt Schiele. *Deepcut: Joint subset partition and labeling for multi person pose estimation*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4929–4937, 2016.

[Redmon2016] Joseph Redmon, Santosh Divvala, Ross Girshick et Ali Farhadi. *You only look once: Unified, real-time object detection*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[Rekant2022] Julie Rekant, Scott Rothenberger et April Chambers. *Inertial measurement unit-based motion capture to replace camera-based systems for assessing gait in healthy young adults: Proceed with caution*. Measurement: Sensors, page 100396, 2022.

[Rempe2020] Davis Rempe, Leonidas J Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas et Jimei Yang. *Contact and Human Dynamics from Monocular Video*. page 27, 2020.

[Rempe2021] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar et Leonidas J Guibas. *HuMoR: 3D Human Motion Model for Robust Pose Estimation*. page 23, 2021.

[Ren2015] Shaoqing Ren, Kaiming He, Ross Girshick et Jian Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks*. Advances in neural information processing systems, vol. 28, 2015.

[Reveret2020] Lionel Reveret, Sylvain Chapelle, Franck Quaine et Pierre Legreneur. *3D visualization of body motion in speed climbing*. Frontiers in Psychology, vol. 11, page 2188, 2020.

[Rosenblatt1958] Frank Rosenblatt. *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological review, vol. 65, no. 6, page 386, 1958.

[Rumelhart1986] David E Rumelhart, Geoffrey E Hinton et Ronald J Williams. *Learning representations by back-propagating errors*. nature, vol. 323, no. 6088, pages 533–536, 1986.

[Seethapathi2019] Nidhi Seethapathi, Shaofei Wang, Rachit Saluja, Gunnar Blohm et Konrad P. Kording. *Movement science needs different pose tracking algorithms*. arXiv:1907.10226 [cs, q-bio], 7 2019. arXiv: 1907.10226.

[Serrancolí2020] Gil Serrancolí, Peter Bogatikov, Joana Palés Huix, Ainoa Forcada Barberà, Antonio J. Sánchez Egea, Jordi Torner Ribé, Samir Kanaan-Izquierdo et Antoni Susín. *Marker-Less Monitoring Protocol to Analyze Biomechanical Joint Metrics During Pedaling*. IEEE Access, vol. 8, pages 122782–122790, 2020. event: IEEE Access.

[Seth2018] Ajay Seth, Jennifer L. Hicks, Thomas K. Uchida, Ayman Habib, Christopher L. Dembia, James J. Dunne, Carmichael F. Ong, Matthew S. DeMers, Apoorva Rajagopal, Matthew Millard, Samuel R. Hamner, Edith M. Arnold, Jennifer R. Yong, Shrinidhi K. Lakshmikanth, Michael A. Sherman, Joy P. Ku et Scott L. Delp. *OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement*. PLOS Computational Biology, vol. 14, no. 7, page e1006223, 7 2018.

[Slembrouck2020] Maarten Slembrouck, Hiep Luong, Joeri Gerlo, Kurt Schütte, Dimitri Van Cauwelaert, Dirk De Clercq, Benedicte Vanwanseele, Peter Veelaert et Wilfried Philips. *Multiview 3D Markerless Human Pose Estimation from OpenPose Skeletons*. In Jacques Blanc-Talon, Patrice Delmas, Wilfried Philips, Dan Popescu et Paul Scheunders, editeurs, Advanced Concepts for Intelligent Vision Systems, volume 12002, pages 166–178. Springer International Publishing, Cham, 2020.

[Stenum2021] Jan Stenum, Cristina Rossi et Ryan T. Roemmich. *Two-dimensional video-based analysis of human gait using pose estimation*. PLoS Computational Biology, vol. 17, no. 4, 4 2021.

[Takahashi2018] Kosuke Takahashi, Dan Mikami, Mariko Isogawa et Hideaki Kimata. *Human Pose as Calibration Pattern: 3D Human Pose Estimation with Multiple Unsynchronized and Uncalibrated Cameras*. pages 1856–18567, Salt Lake City, UT, USA, 6 2018. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE.

[Topham2021] Luke K Topham, Wasiq Khan, Dhiya Al-Jumeily et Abir Hussain. *Human Body Pose Estimation for Gait Identification: A Comprehensive Survey of Datasets and Models*. ACM Computing Surveys, 2021.

[Topley2020] Matt Topley et James G. Richards. *A comparison of currently available optoelectronic motion capture systems*. Journal of Biomechanics, vol. 106, page 109820, 6 2020.

[Tsushima2003] Hitoshi Tsushima, Meg E Morris et Jennifer McGinley. *Test-Retest Reliability and Inter-Tester Reliability of Kinematic Data from a Three-Dimensional Gait Analysis System*. Journal of the Japanese Physical Therapy Association, vol. 6, no. 1, pages 9–17, 2003.

[Uhlrich2022] Scott D. Uhlrich, Antoine Falisse, Łukasz Kidziński, Julie Muccini, Michael Ko, Akshay S. Chaudhari, Jennifer L. Hicks et Scott L. Delp. *OpenCap: 3D human movement dynamics from smartphone videos*. page 2022.07.07.499061, Jul 2022.

[Uijlings2013] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers et Arnold WM Smeulders. *Selective search for object recognition*. International journal of computer vision, vol. 104, no. 2, pages 154–171, 2013.

[Viswakumar2019] Aditya Viswakumar, Venkateswaran Rajagopalan, Tathagata Ray et Chandu Parimi. *Human Gait Analysis Using OpenPose*. pages 310–314. 2019 Fifth International Conference on Image Information Processing (ICIIP), 11 2019. ISSN: 2640-074X.

[Wade2021] Logan Wade, Laurie Needham, Murray Evans, Steffi Colyer, Darren Cosker, James Bilzon et Polly McGuigan. *Application of deep learning-based pose estimation methods for clinical gait outcome measures*. In Proceedings of the Congress of the International Society of Biomechanics, Stockholm, Sweden, pages 25–29, 2021.

[Wang2021] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He et Ling Shao. *Deep 3D human pose estimation: A review*. Computer Vision and Image Understanding, page 103225, 5 2021.

[Wang2022a] Chien-Yao Wang, Alexey Bochkovskiy et Hong-Yuan Mark Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. arXiv preprint arXiv:2207.02696, 2022.

[Wang2022b] Wenming Wang, Kaixiang Zhang, Haopan Ren, Dejian Wei, Yanyan Gao et Juncheng Liu. *UULPN: An ultra-lightweight network for human pose estimation based on unbiased data processing*. Neurocomputing, vol. 480, pages 220–233, 2022.

[Windt2020] Johann Windt, Kerry MacDonald, David Taylor, Bruno D Zumbo, Ben C Sporer et David T Martin. *“To tech or not to tech?” A critical decision-making framework for implementing technology in sport*. Journal of Athletic Training, vol. 55, no. 9, pages 902–910, 2020.

[Xiang2019] Donglai Xiang, Hanbyul Joo et Yaser Sheikh. *Monocular total capture: Posing face, body, and hands in the wild*. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10965–10974, 2019.

[Xu2020] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang et Wenjun Zhang. *Deep kinematics analysis for monocular 3d human pose estimation*. In Proceedings of the IEEE/CVF Conference on computer vision and Pattern recognition, pages 899–908, 2020.

[Xu2021] Yan Xu, Yu-Jhe Li, Xinshuo Weng et Kris Kitani. *Wide-baseline multi-camera calibration using person re-identification*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13134–13143, 2021.

[Zago2020] Matteo Zago, Matteo Luzzago, Tommaso Marangoni, Mariolino De Cecco, Marco Tarabini et Manuela Galli. *3D Tracking of Human Motion Using Visual Skeletonization and Stereoscopic Vision*. Frontiers in Bioengineering and Biotechnology, vol. 8, 2020.

[Zhang2013] Jun-Tian Zhang, Alison C Novak, Brenda Brouwer et Qingguo Li. *Concurrent validation of Xsens MVN measurement of lower limb joint angular kinematics*. Physiological measurement, vol. 34, no. 8, page N63, 2013.

[Zhang2020] Zhe Zhang, Chunyu Wang, Wenhu Qin et Wenjun Zeng. *Fusing wearable IMUs with multi-view images for human pose estimation: A geometric approach*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2200–2209, 2020.

[Zheng2022] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz et Mubarak Shah. *Deep learning-based human pose estimation: A survey*. arXiv, 2022.

List of Figures

List of Tables

A
Appendix A : Title

Summary here

A.1 Section 1

A.1.1 Sous section 1

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1.2 Sous section 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

B
Appendix B : Title

Summary here.

B.1 Section 1

B.1.1 Sous section 1

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

B.1.2 Sous section 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

C
Appendix C : Title

Summary here.

C.1 Section 1

C.1.1 Sous section 1

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

C.1.2 Sous section 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

"Design, evaluation, and application of a workflow for biomechanically consistent markerless kinematics in sports"

"Conception, évaluation, et application d'une méthode biomécaniquement cohérente de cinématique sans marqueurs en sport"

Résumé

Ici ... résumé en français.

Mots-clés : Mots clés

Abstract

Ici ... résumé en anglais.

Keywords : markerless motion capture; sports performance analysis; kinematics; computer vision; openpose; opensim; python package