International Journal of Computer Vision (2021) 129:3338–3366
https://doi.org/10.1007/s11263-021-01526-z

DeMoCap: Low-Cost Marker-Based Motion Capture

Anargyros Chatzitofis[1,2] · **Dimitrios Zarpalas**[2] · **Petros Daras**[2] · **Stefanos Kollias**[1]

Received: 16 October 2020 / Accepted: 8 September 2021 / Published online: 15 October 2021

Abstract
Optical marker-based motion capture (MoCap) remains the predominant way to acquire high-fidelity articulated body motions. We introduce DeMoCap, the first data-driven approach for end-to-end marker-based MoCap, using only a sparse setup of spatio-temporally aligned, consumer-grade infrared-depth cameras. Trading off some of their typical features, our approach is the sole robust option for far lower-cost marker-based MoCap than high-end solutions. We introduce an end-to-end differentiable markers-to-pose model to solve a set of challenges such as under-constrained position estimates, noisy input data and spatial configuration invariance. We simultaneously handle depth and marker detection noise, label and localize the markers, and estimate the 3D pose by introducing a novel spatial 3D coordinate regression technique under a multi-view rendering and supervision concept. DeMoCap is driven by a special dataset captured with 4 spatio-temporally aligned low-cost Intel RealSense D415 sensors and a 24 MXT40S camera professional MoCap system, used as input and ground truth, respectively.

Keywords Motion capture · Low-cost · Marker-based · Depth-based · Pose regression · Multi-view

1 Introduction

Extensive research efforts have been devoted to the development of motion capture (MoCap), one of the de facto standards for human-centric interactive media capture and production. Nowadays, optical MoCap solutions, especially the marker-based ones, are considered essential to several applications and industry sectors such as film making and VFX, sports, health, gaming and immersive realities (XR). MoCap enables online and offline high-fidelity capturing and digitization of body, hand and facial movements derived from the performances of real people and beyond. This enables its use in various applications such as 3D character animation, computer-human interaction, robotic units control, physical exercise monitoring, and more.

Despite the appearance of several commercial and academic motion capture solutions, marker-based MoCap still remains the gold-standard in the field. That is due to its extremely high accuracy and frequency, as well as the production-ready maturity that ensures high quality outcomes in a short amount of time.

Nevertheless, the marker-based MoCap production process is not flawless, it suffers from well-known drawbacks. Raw optical motion capture data are often erroneous, due to marker occlusions or mislabeling from marker swapping during tracking, with high frequency noise or jitter and requiring time-consuming post-processing by hand. Beyond data cleaning, articulated body part fitting to marker subsets and skeleton retargeting for local joint transformation solving are further required, undoubtedly making it a laborious and time-consuming process. On top of that, the complexity and costs of marker-based MoCap systems with numerous infrared specialized cameras are high, making them inaccessible to the wider interested audience.

These complications attract the interest of the research community to investigate and propose novel alternatives. On the one hand, computer graphics researchers intensify their efforts on models that resolve or soften these issues (Holden

Communicated by Gregory Rogez.

✉ Anargyros Chatzitofis
tofis3d@gmail.com

Dimitrios Zarpalas
zarpalas@iti.gr

Petros Daras
daras@iti.gr

Stefanos Kollias
stefanos@cs.ntua.gr

[1] National Technical University of Athens, Zografou Campus, Iroon Polytechniou 9, 15780 Zografou, Athens, Greece

[2] Centre for Research and Technology Hellas, 6th km Charilaou-Thermi, 57001 Thermi, Thessaloniki, Greece

(2018); Han et al. (2018); Bascones (2019); Perepichka et al. (2019)), while computer vision labs concentrate their efforts on ground-breaking markerless methods, posing MoCap mostly as a 3D pose estimation task (Iskakov et al. (2019); Qiu et al. (2019); Tu et al. (2020)).

Nevertheless, existing marker-based solutions do not satisfy the need for flexible and low-cost options, while recent markerless models following the deep learning paradigm, though effective and more flexible, cannot reach equal robustness and precision levels in the absence of strong and deterministic priors such as body-worn markers. Focusing on this gap, our research is driven by three main factors: (i) the accuracy and precision of marker-based solutions that obtain the articulated body movements with the aid of body-attached markers in a highly accurate and deterministic way, (ii) the remarkable ability of deep models to solve vision-based problems and (iii) the recent developments on consumer-grade and low-cost depth-sensing cameras. We inherit the best of their characteristics and blend them in an efficient way to present a "hybrid" lightweight motion capture approach.

In this paper, we propose DeMoCap, targeting low-cost marker-based motion capture by combining traditional marker-based MoCap and deep neural networks applied on visual data captured with low-cost depth-sensors. To the best of our knowledge, DeMoCap, though not equal to professional high-end MoCap systems on several aspects such as the high frequency or the size of the capturing space, is the first deep model that enables the use of far lower-cost equipment and professional retro-reflective marker configuration for robust motion capture (Fig. 1).

We employ a sparse, spatio-temporally aligned multi-view setup of consumer-grade and low-cost depth-sensing cameras to track a dense configuration of spherical retro-reflective markers. To address the marker-based MoCap challenges, i.e. marker denoising, labeling, tracking and joint transformation solving and retargeting, and the limitations of low-cost depth sensor setup, i.e. viewpoint sparsity, depth noise, under constraint data and infrared image blurriness for accurate blob detection, we propose an end-to-end, fully differentiable data-driven model to directly regress the 3D pose. This way, we pose the problem of MoCap as a markers-to-pose regression task.

Even though our input is three dimensional, we avoid the use of 3D convolutions, as we target lightweight, real-time and close to real-time applications. We introduce a novel spatial 3D regression module on top of latent heatmaps predicted by a 2D fully convolutional neural network (FCN) to regress the markers and joints 3D coordinates, in a fully differentiable manner. We experiment with various multi-stage FCN architectures by building super-stages, i.e. grouping the former and the latter stages to regress the marker and joint 3D coordinates, respectively, staging a smooth representation transition from markers to 3D pose (markers-to-pose).

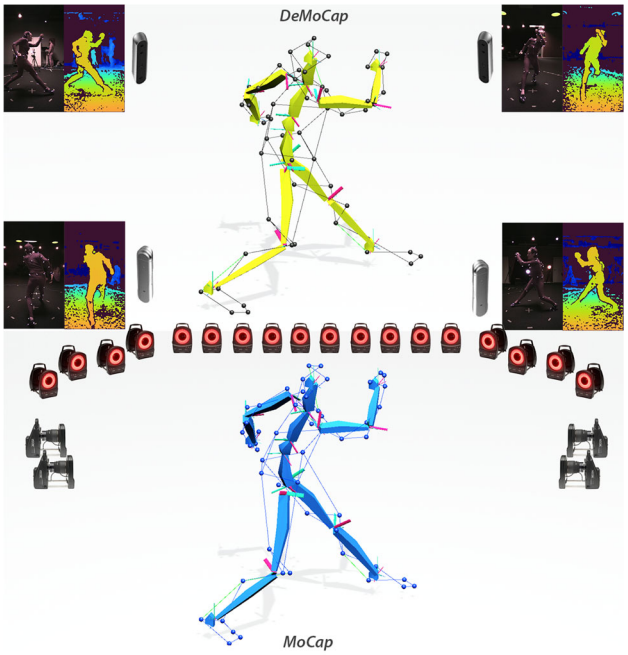

Fig. 1 DeMoCap stands as the first marker-based MoCap solution enabling the use of only a sparse set of consumer-grade depth sensors for far lower costs, and higher portability and flexibility, than commercial high-end solutions, trading off, however, some of their typical features

Following this approach, we drive the network to learn the spatial and hierarchical relation between the markers and the underlying pose .

For feeding our network, we apply a volumetric scale normalization to uniformly distribute the marker cloud in a cuboid 3D space for higher data spatial invariance and sparsity. Then, we adopt a multi-view rendering technique to render the markers from opposing orthographic cameras whose principal axis passes through the center of the sparse marker cloud. Notably, we preserve the 3D information of the markers by splatting their relative depth on multiple views, sticking to two opposing views for DeMoCap, creating "sparse" depth maps of the markers to feed our model. From that point, we approach our task as a 3D keypoint regression problem from dual-view 2D depth maps. The model is driven to assimilate the articulated relation between markers and joints, sequentially regressing their 3D coordinates in a forward pass per infrared-depth multi-view frame, without any body structure prior or explicit association between body parts and marker subsets. Summarizing, our contributions are:

- To the best of our knowledge, DeMoCap constitutes the first data-driven approach that employs efficient fully convolutional neural networks to simultaneously regress optical markers and 3D pose from sparse 3D point sets, captured with the use of a low-cost multi-view depth-sensor setup .

- A scale and translation invariant representation in a normalized 3D space is introduced to train our network. Learning upon it, the model overcomes bias on our relatively limited training data and generalizes well.
- We stage a smooth representation transition from markers to 3D pose. Our model is driven to learn the underlying structural relation between human body and marker configuration placement, decoding marker and bone associations from sparse and noisy 3D marker point clouds, resulting in accurate pose estimation.
- We pose 3D keypoint estimation as a joint 2D localization and regression objective within our normalized 3D space to embed the z-dimension indirectly with the introduction of a new fully differentiable module for 3D regression.
- We make our special dataset publicly available. Our dataset contains inter- and intra-system spatio-temporally aligned infrared-depth and motion capture data. On top of that, the former have been captured with hardware synchronization for precise temporal consistency between the multiple views.

The rest of the paper is structured as follows. Section 2 gives a brief overview of related data-driven works on (a) marker-based MoCap automation and refinement, (b) multi-view markerless vision-based 3D pose estimation and (c) keypoint localization techniques applicable on pose estimation. Section 3 presents the dataset creation and processing to familiarize the reader with the nature of our challenge and the way we approach it. In Sect. 4, our methodical approach is discussed, presenting and explaining in depth the rationale behind its contributions. In Sect. 5, we present quantitative and qualitative experimental results along with ablation outcomes to justify our contributions. In Sect. 6, we discuss the pros and cons of DeMoCap in comparison with existing marker-based motion capture solutions and recent markerless pose estimation approaches. Sect. 7 concludes and presents potential avenues for future work.

2 Related work

Our approach requires domain knowledge from marker-based practices and recent data-driven 3D pose estimation techniques. To this end, in Sect. 2.1, we first review approaches recently appeared in the literature that focus on the processing of marker-based optical motion capture data (Bascones (2019); Loper et al. (2014); Holden (2018); Han et al. (2018); Perepichka et al. (2019)). These works highlight traditional MoCap challenges such as marker labeling and denoising, or direct joint transformation solving from optical data in an efficient and automatic fashion by applying machine learning techniques. Another challenge addressed by recent works, such as the ones proposed by Loper et al. (2014); Mahmood et al. (2019), is related to body shape deformation issues. Similar to most prior works, we do not explicitly account for this, but instead, focus on simultaneous overcoming of the aforementioned MoCap challenges.

On the other hand, in spite of the use of reflective markers in the present work, we pose our solution mostly as a 3D pose estimation task from noisy and sparse 3D data. Thus, in Sect. 2.2, we discuss recent data-driven approaches that estimate 3D pose using FCNs for keypoint 3D coordinate regression, highlighting the pros and cons of each solution. Finally, in Sect. 2.3, we give a short overview of keypoint localization approaches to correlate them with the 3D coordinate regression technique we introduce.

2.1 Marker-Based Optical Motion Capture

The classic marker-based optical solutions have been undoubtedly the gold-standard in motion capture for decades. Nevertheless, the existence of drawbacks such as the need for post-processing for data cleaning as well as the expensive hardware and complexity of their setups, are considered a challenge and attract the interest of the research community. We discuss recent and novel works that apply machine learning techniques on marker-based optical data for marker denoising and joint transformation solving.

Bascones (2019) tackles automatic marker labeling as a machine learning classification problem, to train a set of weak classifiers in an ensemble of partial solvers. The result is used to feed an online algorithm providing efficient and lightweight marker labeling. Alexanderson et al. (2017) present a robust online method for identification and tracking of passive motion capture markers attached to non-rigid structures. The method is especially suited for large capture volumes and sparse marker sets. By using multiple assignment hypotheses and soft decisions, it can robustly recover from challenging poses with several simultaneous occlusions and false observations (ghost markers). Holden (2018) proposes a fast method for robust joint transformations solving of optical motion capture data by using machine learning denoising techniques. This data-driven approach, being robust to erroneous marker 3D positions, replaces the solving part of the motion capture pipeline, removing the need for manual data cleaning. However, the method is limited to be used with motion capture data from commercial solutions, while not applicable in real-time use cases. Recently, Perepichka et al. (2019) introduced a method that robustly detects and repairs marker trajectories by replacing erroneous segments with synthetically generated ones producing kinematically correct paths. Using the joint transformation solver proposed by Holden (2018), an initial kinematic motion is constructed, and using it as reference, erroneous trajectories are detected and filled by transferring the paths from the kinematic solver in a shape preserving way.

Han et al. (2018) introduce an online optical marker labeling model for hand tracking, framing the labeling problem as a direct keypoint regression demonstrating it to sequences with occluded and ghost markers. The model is accurate and fast, since trained in a great amount of synthetic data, albeit not evaluated on highly noisy and challenging marker sets. The model regresses the marker 3D coordinates with the use of fully connected layers which are prone to overfitting when the pool of data is limited, hampering the generalization ability of the overall network.

Chatzitofis et al. (2019) firstly introduced the joint use of deep neural networks and multi-view depth-sensing for marker-based motion capture. Nevertheless, the method is not based on an end-to-end data-driven model. The model detects and labels the markers, however the body pose is estimated by applying forward kinematics to an articulated human body prior. Moreover, instead of spherical commercial markers that allow for high-fidelity tracking, a sparse and coarse set of custom retro-reflective straps and patches was attached on the subjects' bodies. Despite its valid concept, the method has limited degrees-of-freedom due to the low number of markers and its custom marker placement, while the marker labeling takes place separately for each camera, most likely making the model biased to the camera poses and their intrinsic parameters.

Even though classic marker-based MoCap constitutes a specialized solution for professional motion capture, the research works discussed in this section try to overcome well-known issues that MoCap suffers from, i.e. marker labeling, ghost marker denoising, occluded marker recovery, marker swapping and joint transformation solving. In the present work, we overcome these issues in an end-to-end manner by directly regressing the marker and pose coordinates, staging in our network a smooth representation transition between them.

2.2 Markerless 3D Pose Estimation

On top of these challenges, the use of markers increases the complexity of the motion capture setup, while the body joint solving from markers is a non-trivial task. During the last decade, numerous research labs intensively work on simple, markerless and more flexible approaches using low-cost resources (Sigal et al. (2012)). Most recent methods focus on monocular vision, mostly using color (Mehta et al. (2017); Pavllo et al. (2018); Mehta et al. (2019); Cheng et al. (2019); Guler and Kokkinos (2019); Rüegg et al. (2020)), and some using depth (Haque et al. (2016); Park et al. (2017); Martínez-González et al. (2018b); Martínez-González et al. (2018a)). Fewer but not limited methods approach 3D pose estimation from multi-view color streams (Burenius et al. (2013); Elhayek et al. (2015); Rhodin et al. (2018); Qiu et al. (2019); Iskakov et al. (2019); Tu et al. (2020)), while pose estimation from multi-view depth maps is relatively unexplored (Bekhtaoui et al. (2020)). More relevant to our approach are recent 3D pose estimation works on spatio-temporally aligned multi-view visual streams.

Iskakov et al. (2019) present two variations of a learnable triangulation-based technique, an algebraic and a volumetric one, to estimate 3D pose jointly from multiple 2D color views. The former is based on soft triangulation with learnable camera-joint confidence weights, while the latter is based on dense geometric aggregation of 2D heatmap predictions from multiple viewpoints. The aggregated volume is then refined via 3D convolutions to produce 3D heatmaps that allow modelling a human pose prior. The method showcases satisfying results, as presented in the original work and shown in our experiments in Sect. 5.3, however it is slow and requires multi-view color input, a domain sensitive input (Buhrmester et al. (2019)).

Qiu et al. (2019) propose another approach under a similar concept, i.e. estimating 2D heatmaps in multi-view images and recovering 3D poses from multi-view 2D predictions. At first, a convolutional neural network (CNN) jointly estimates 2D poses through a cross-view fusion scheme which allows for refined 2D pose estimation. Then, applying a recursive pictorial structure model (RPSM), the 3D pose is recovered from the multi-view 2D poses and gradually improves, since RPSM recursively discretizes the volume around each joint previously predicted 3D location into a finer-grained grid. The inference performance limits the online operation of the method.

Tu et al. (2020) recently presented VoxelPose, a multi-view and multi-person data-driven 3D pose estimation approach. Contrary to the aforementioned multi-view methods whose cross-view correspondence is based on weak 2D pose estimates, VoxelPose directly operates in the 3D space. Features from the camera views are aggregated in the 3D space and fed into a cuboid proposal network to localize multiple subjects in the capturing space by predicting a number of 3D cuboid proposals from the 3D feature volume. Then, a separate finer-grained feature volume, centered at each proposal, is created and fed into a 3D pose regression network. Despite the frequent occlusions between multiple people at the same scene, the approach is accurate and robust.

Other approaches are based on parametric body models with skeleton hierarchy that allow the expression of the body pose and motion. Joo et al. (2018) build upon generative body deformation models to fit to data from multiple viewpoints. Leveraging face, body and hand landmarks with the use of 2D detectors from multiple views, 3D keypoints are obtained and used to train the parametric models, also allowing for capturing of additional variations of hair and clothing. To that end, full body (face, hands, body) motion capture is achieved with the use of 3D deformable models. Potential errors of single-view 2D keypoint detection can add bias to the models.

Zhang et al. (2020b) propose a new multi-view and multi-person motion capture approach. Based on confidence map (heatmaps) and part-affinity fields (PAFs) predicted with the use of OpenPose by Cao et al. (2017), the proposed method unifies per-view parsing, cross-view matching and temporal tracking with the introduction of a 4D association graph. The 4D association graph is efficiently solved with the introduction of 4D limb bundle parsing based on heuristic searching and a bundle Kruskal's algorithm.

Contrary to our approach, most of the aforementioned methods, though effective, are not lightweight, cannot perform even close to real-time, require multi-view color images (a domain sensitive input), while most of them are biased to the errors of the initially required 2D detections.

2.3 Keypoint Localization

We approach MoCap as a cooperative marker and joint 3D coordinate regression task from sparse depth images and, thus, we offer an overview of the state-of-the-art approaches for keypoint localization. Nowadays, many vision-based problems are posed as 2D/3D keypoint localization tasks, where deep 2D CNNs and FCNs have been proven effective. In the recent literature, the state-of-the-art methods for keypoint localization fall into three main categories, i.e. direct regression, dense prediction, and spatial regression.

Direct regression methods, used in several tasks, such as pose estimation in DeepPose by Toshev and Szegedy (2014) and optical marker labeling (Han et al. (2018)), bypass the spatial nature of images due to its fixed-size representation. Instead, these methods implicitly learn to directly regress the 2D/3D positions based on the expressive power of the models. Nevertheless, direct regression can be supervised with distance-based losses, which is the direct objective of keypoint localization.

Dense, heatmap-based prediction methods employ FCNs to predict confidence scores for each input pixel and are supervised during training with heatmaps reconstructed in most cases via 2D Gaussian distributions with fixed-variance. Such techniques have been employed in well-known pose estimation works (Cao et al. (2017); Wei et al. (2016)), presenting higher image translation invariance than direct regression due to their spatial aspect. The keypoints in these methods are localized by calculating the *ArgMax* or alternative heuristic approaches (Tompson et al. (2014)) on the predicted dense heatmap. The main drawback of these methods is the use of intermediate, structural loss functions that train the network to predict pixel confidence scores, thus, the supervision during training is not aligned to the direct objective of the method, as in direct regression.

Spatial regression methods have proved the most effective in various vision-based tasks, combining the strengths of direct regression and dense heatmap prediction methods. In particular, as in dense heatmap prediction, these models are translation invariant due to the use of FCNs, also allowing for distance-based loss supervision. While dense heatmap prediction methods train the network to predict heatmaps matching to pre-defined arbitrary heatmaps, spatial regression networks learn the optimal latent heatmap that yields the most accurate point localization. Spatial regression, the Center of Mass (CoM) of a probability map as discussed in the work proposed by Tensmeyer and Martinez (2019), was almost simultaneously introduced by Sun et al. (2018) as *integral regression*, by Nibali et al. (2018) as *differentiable spatial-to-numerical transform* (DSNT), and by Luvizon et al. (2019) as *Soft-argmax*.

Given its proved effectiveness, we also adopt spatial regression for point localization. However, aiming efficient 3D localization, we go beyond standard techniques by jointly encoding x-, y- the z-dimension in latent heatmaps with the introduction of a fully differentiable module for 3D coordinate regression. We describe this new 3D coordinate regression module in Sect. 4.2.1.

3 Depth-Based Optical Marker Data

We created a special and unique dataset of spatio-temporally aligned motion capture and multi-view infrared-depth data to serve our scope (see Sect. 3.2). Our dataset constitutes the first visual data collection that contains spatio-temporally aligned multi-view colored infrared and depth images with 3D pose and marker annotations. We captured various activities performed by actors with retro-reflective markers attached on their bodies, being visible and distinguishable to the infrared images. To achieve that, we used the depth sensor infrared emitters to emit infrared light in the scene causing reflections on the retro-reflective markers (see Fig. 2). The infrared and depth images, which are aligned and defined on the same image domain, enable single-view marker 3D localization, which is discussed in Sect. 3.3. We achieve marker and pose 3D keypoint annotations by addressing the challenge of synchronization and spatial calibration between a professional motion capture and a multi-view depth sensor system, described in detail in Sect. 3.4.

We captured data of 4 actors, 2 males and 2 females, performing 11 different activities of approximately 20 s each, starting their performances in a T-Pose ensuring appropriate tracking initialization of the commercial MoCap system.[1]

In total, more than 20,000 samples are included in our dataset, however details on how we split it for training and evaluation based on the subject and activity criteria are given in Sect. 5.1.

[1] For the sake of clarity, we mention that T-Pose or any other pose is not required for the initialization of our method.

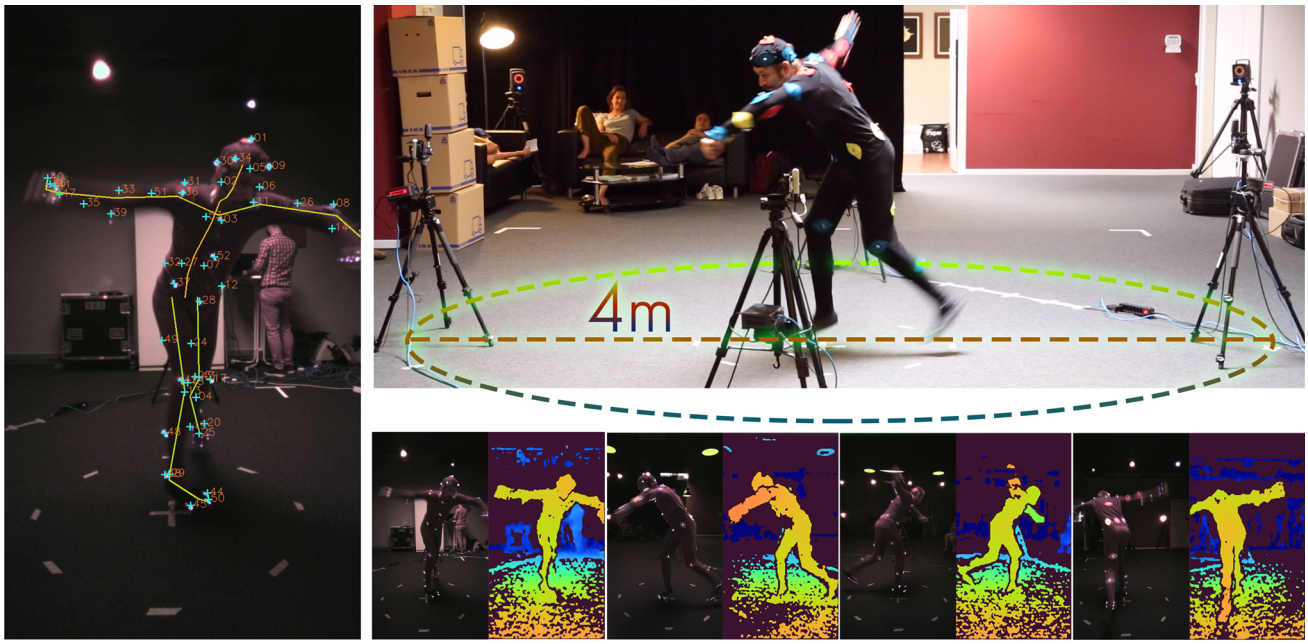

Fig. 2 The capturing setup with 24 VICON (1984) MXT40S cameras and a low-cost multi-view depth sensor system equipped with 4 Intel RealSense D415 stereo-based depth sensing devices. Intense marker reflections are provoked to the infrared streams by emitting infrared light to the retro-reflective markers attached on the subjects' body. To limit the image blurriness, we reduced the exposure time of the sensors which, consequently, reduced the lightness of the image leading to more distinguishable marker reflections in comparison to the default settings. Infrared-depth image pairs are shown on the bottom of the figure, while at the left side, the ground truth markers and pose are projected on one of the infrared views to depict the spatio-temporal alignment between the motion capture and the infrared-depth systems

3.1 Capturing Setup

A professional VICON (1984) motion capture system with 24 Vicon MXT40S cameras and a low-cost volumetric capturing system[2] with 4 Intel RealSense D415 stereo-based depth sensing devices (Keselman et al. (2017)) were used, recording the data at 120 and 30 frames/second, respectively.

3.2 Marker configurations and body structure

For the marker set, we used $M = 53$ adhesive spherical retro-reflective markers of 14 mm diameter, which were attached on the motion capture suits of the actors.

We consider the post-processed, clean marker data $\mathbf{M}_{gt} \in \mathbb{R}^{M\times 3}$ as ground truth. With respect to the body structure, the original sequences provide poses of 33 different joints, however we simplify the structure and use $J = 19$. The clean pose data, $\mathbf{J}_{gt} \in \mathbb{R}^{J\times 3}$, are considered as ground truth. Samples of the annotated depth-infrared image pairs along with the capturing setup in the MoCap studio where the dataset creation took place, are depicted in Fig. 2.

[2] For the dataset recordings, we used the publicly available volumetric capturing tool (https://github.com/VCL3D/VolumetricCapture) proposed by Sterzentsenko et al. (2018).

3.3 Optical Marker Data from Multiple Depth Sensors

Most of the recent low-cost consumer-grade depth sensing devices are equipped with infrared cameras and emitters (Keselman et al. (2017); Zhang (2012)). The Intel RealSense D415 depth camera is based on active stereo vision to calculate depth, consisting of a two infrared camera eyes and an infrared projector to improve depth accuracy in scenes with low texture features. The infrared projector casts static infrared pattern to the scene where the markers reflect back to the receiver enabling straightforward amplitude-based detection. We consider C spatio-temporally aligned depth cameras $c \in \{1, \ldots, C\}$, perimetrically placed around a capturing space of approximately $4m$ diameter, as shown in Fig. 2. Each camera acquires a pair of one colored infrared image $\mathbf{I}_c(\mathbf{p}) \in \mathbb{R}^3$ and one depth map $D_c(\mathbf{p}) \in \mathbb{R}$, with $\mathbf{p} := (x, y) \in \Omega$ being the coordinates of the pixels in the image domain Ω defined in a $w \times h$ grid, with w and h being its width and height, respectively. The sensor poses $\mathbf{T}_c := \left[\begin{smallmatrix} \mathbf{R}_c & \mathbf{t}_c \\ \mathbf{0} & 1 \end{smallmatrix}\right]$ are known in a common coordinate system, where $\mathbf{R}_c$ and $\mathbf{t}_c$ denote rotation and translation respectively. Hence, we can transform the depth image domain coordinates of each view to a global coordinate system by:

$$\mathcal{T}_c(\mathbf{p}) = \mathbf{T}_c \pi^{-1}(D_c(\mathbf{p}), \mathbf{K}_c, \mathbf{p}), \quad (1)$$

with $\mathbf{T}_c$ being the relative pose from the local coordinate system of sensor c to the global one and π^{-1} denoting the deprojection function that transforms the depth pixel to 3D coordinates, using sensor's intrinsic parameters matrix $\mathbf{K}_c$. Given that the infrared $\mathbf{I}_c$ and depth D_c images of camera c are aligned and defined on the same image domain Ω, we apply a linear thresholding as proposed by Gaschler (2011) for efficient marker segmentation. Following, a fast contour detection algorithm is applied to yield the blob centers per view, which we then map in the common, global 3D space using Eq. 1. During this process, the points not contained in a 3D bounding box set to limit the capturing 3D space, are considered as outliers and are discarded. To this end, a sparse marker cloud, $\mathbf{M}_r \in \mathbb{R}^{M_f \times 3}$ of M_f 3D points is extracted per frame, containing the raw marker 3D coordinates as obtained from the multiple sensors. Given the noise of the sensors as well as the separate detection from each view, M_f is varying around the real number of markers, i.e. $M = 53$.

The quality of the raw marker tracking is analogous to the number of views, as in the majority of multi-view systems, mostly due to the elimination of occlusions and, consequently, of missing markers. We selected a 4-sensor setup in a cross placement for the creation of our dataset, considering it the trade-off between avoiding occlusions and low cost. Nevertheless, the proposed model is trained on highly noisy data resulted from weak optical marker tracking (one valid observation is enough at least in one of the views) in the form of a sparse and spatially invariant 3D data representation, eliminating the bias of the camera poses, intrinsic parameters or systematic depth noise, as we discuss in our ablation study, in Sect. 5.4.

3.4 Spatio-Temporal Alignment

High synchronization precision between our low-frequency depth sensors (30 frames/second in our dataset, see Sect. 3.1 for further details) is a prerequisite. The selected Intel RealSense D415 offers intra- and inter-sensor hardware synchronization, allowing for high precision temporal alignment. With respect to the inter-system (D415, low-cost - VICON, high-cost) synchronization, the global temporal offset between the systems was detected with a clapperboard equipped with 2 markers at the beginning of each sequence. The varying frame rate inter-system sequences were then aligned considering their local time steps after removing the global offset.

The spatial alignment between the VICON and depth sensor system is achieved by a two-step process. We perform an initial alignment by using the 3D positions of the markers, as estimated by each modality, i.e. triangulation-based for VICON and depth-based for D415. We exploit the starting, static T-Pose phase of each sequence where the markers are easily detected from the low-cost system, as the infrared images are crisp and sharp. We apply Iterative Closest Point (ICP) to coarsely transform $\mathbf{M}_r$ to the coordinate system of the ground-truth markers $\mathbf{M}_{gt}$, resulting in $\mathbf{M}_r^{'}$. We use ICP since, as mentioned in Sect. 3.3, the number of the detected markers is varying per frame and there are no direct correspondences.

For an accurate spatial registration, we follow up with a sensor pose refinement step. At first, we find the correspondences between $\mathbf{M}_{gt}$ and $\mathbf{M}_{r,c}^{'}$, where $\mathbf{M}_{r,c}^{'} \subset \mathbf{M}_r^{'}$ is the subset of the markers belonging to each sensor c. To solve this, we construct bipartite graphs between $\mathbf{M}_{gt}$ and $\mathbf{M}_{r,c}^{'}$, where the edge weights represent euclidean distances between 3D points. Finally, we apply minimum weight bipartite matching by:

$$\mathcal{B}_{M,c}(\mathbf{M}_{r,c}^{'}, \mathbf{M}_{gt}) = \min_{M_{gt}} \sum_i ||x_i - y_i||_2 \tag{2}$$

where $x_i \in \mathbf{M}_{r,c}^{'}$ and $y_i \in \mathbf{M}_{gt}$. From $\mathcal{B}_{M,c}$, we only use the correspondences under a strict threshold, ensuring a high quality correspondence group between $\mathbf{M}_{gt}$ and the detected blobs on $\mathbf{I}_c$. Then, we apply Bundle Adjustment (Hartley and Zisserman (2003)) to refine the sensor poses using $\mathbf{M}_{gt}$ as reference. In detail, considering $\mathbf{M}_{gt}$ and intrinsic camera parameters $\mathbf{K}_c$ constant, we jointly and iteratively refine the camera poses to minimize the reprojection errors. This provides a refined spatial alignment based on ground-truth optical marker data $\mathbf{M}_{gt}$, resulting in $\mathbf{M}_{r,c}^{''}$ for the markers of each view and, consequently, to $\mathbf{M}_r^{''}$, also by applying a strict spatial clustering for marker points distances lower than 10 mm to merge only the ones that have been detected extremely close to each other.

It is worth noting that this alignment process is considered for the creation of the dataset and is not required during the model inference where the VICON data are absent. The results of the spatial and temporal alignment between the VICON and the multi-sensor system are presented in Fig. 2 where the pose and markers from VICON are overlayed on an infrared image sample.

3.5 Normalized Orthographic Depth Rendering

Working on a sparse 3D cloud instead of raw infrared or depth images allows us to overcome well-known limitations of such under constraint data. Several data-driven models suffer from these limitations such as the bias on specific camera poses and lightning conditions, the overfitting on the domain specific training set or the distance and systematic depth sensor noise.

We simplify our task by posing it as a spatial 3D regression problem on orthographically rendered depth maps under a multi-view context, i.e. with the use of two (or more) opposing renderings to overcome single-view ambiguities.

We apply a volumetric scale and translation normalization transform $\mathcal{T}_{\mathcal{N}}$ in order for $\mathbf{M}_r^{''}$ to occupy 80% of each dimension of a normalized cuboid 3D space , i.e. ranging between [0.1, 0.9] along the axes, resulting in $\hat{\mathbf{M}}_r$. Although obvious, it is worth noting that the same normalization transform $\mathcal{T}_{\mathcal{N}}$ is applied on the ground-truth data $\mathbf{M}_{gt}$ and $\mathbf{J}_{gt}$, resulting in $\hat{\mathbf{M}}_{gt}$ and $\hat{\mathbf{J}}_{gt}$ respectively for supervision. To that end, the 3D bounding box containing every sample occupies the same volume in 3D, while the margin of 10% from the boundaries ensures the appropriate behaviour of the 2D convolutions across the network layers. We then render $\hat{\mathbf{M}}_r$ from two opposing views resulting in two sparse depth images through two orthographic cameras with the principal point centered on the center of $\hat{\mathbf{M}}_r$. We render the depth images in high pixel resolution (i.e. 800 × 800) and we next linearly interpolate them to 160 × 160 input's resolution, aiming to eliminate the encoding of quantization error and information loss due to quantized rendering (Zhang et al. (2020a)). On rendering, the range of [0.1, 0.9] across "z" axis makes the marker points distinguishable from zero-values to both rendered depth maps. To that end, the depth values preserve marker 3D positions by representing their normalized depth as small areas of splatted depth pixels, creating two "sparse" depth images, $\mathcal{D}_{front}$ and $\mathcal{D}_{back}$, to feed our network. Samples from these rendered normalized depth maps are illustrated in Fig. 3.

Before the application of the normalization transform $\mathcal{T}_{\mathcal{N}}$, we apply a random rotational augmentation around the X, Y and Z axes by $[-10°, 10°]$, $[-180°, 180°]$ and $[-10°, 10°]$ ranges respectively, increasing the variance of the human body part lengths depending on their orientation across the axes of the cuboid volume. It is worth noting that the 3D rotational augmentation in our case, with the input being a sparse cloud of 3D points, has the physical meaning of changing the rendering viewpoint of the camera. This enables the creation of completely new depth map inputs for the network during training, contrary to the limited effect of pseudo-rotational augmentation applied on dense input representations such as color images, depth maps or any other 2D-grid input.

To this end, we introduce a lightweight model for efficient inference on sparse 3D data. We avoid the use of 3D convolutional architectures since, despite their effectiveness (Riegler et al. (2017); Qi et al. (2017); Tu et al. (2020)), 3D convolutions are still computationally expensive and inefficient.

4 Deep Depth-Based Motion Capture

The key factor of DeMoCap's low cost is its reliance on cheap commodity stereo-based infrared-depth sensors, which, despite their noisy sensing, can satisfactorily observe the 3D locations of the retro reflective markers in a monocular fashion (i.e without the need for triangulation between multi-view observations). The low cost of stereo-based infrared-depth sensing still has the price of observation inaccuracy which, along with the aforementioned challenges of marker-based motion capture, we overcome by utilizing a *deep neural network* which enables simultaneous and effective marker and joint 3D coordinate regression.

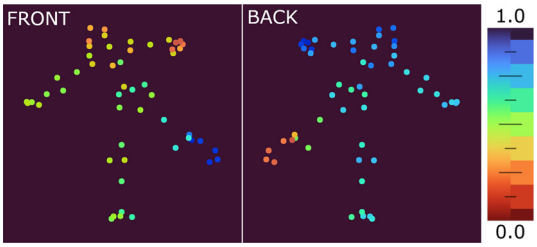

Fig. 3 Input data visualization. $\mathcal{D}_{front}$ and $\mathcal{D}_{back}$ with colorization for the sake of clarity

With DeMoCap, we introduce a data-driven approach for marker-based motion capture from multiple infrared-depth streams, modeled as a staged markers-to-pose 3D regression from noisy marker data, orthographically rendered to multiple viewpoints as depth maps. Our end-to-end data-driven model for marker-based MoCap introduces:

- Marker observation clustering by grouping the raw 3D points as captured by the different viewpoints.
- Ghost marker denoising by ignoring ghost markers caused by erroneous marker detection.
- Missed marker recovery of either occluded or undetected markers.
- Recovering from marker swaps, due to the discrete, one-shot inference.
- Labeled marker localization by spatially regressing the 3D coordinates from latent marker heatmaps.
- Instantaneous 3D pose regression from labeled 3D markers without prior knowledge of body structure. This is backed up by all the above marker-robustness traits in combination with marker labelling and exploitation of 3D information.

The overall pipeline of the proposed method is illustrated in Fig. 4. The acquired multi-view infrared-depth frames captured from the sensors are processed for the extraction of raw marker 3D positions $\mathbf{M}_r^{''}$, which are then normalized yielding $\hat{\mathbf{M}}_r$. $\hat{\mathbf{M}}_r$ are then orthographically rendered on the two opposing depth images, $\mathcal{D}_{front}$ and $\mathcal{D}_{back}$. Given that $\mathcal{D}_{front}$ and $\mathcal{D}_{back}$ represent the spatial distribution of the markers attached on the human body, we feed them to a fully convolutional markers-to-pose staged network. This model sequentially predicts marker and joint latent heatmaps on which we apply a novel dual-view and fully-differentiable

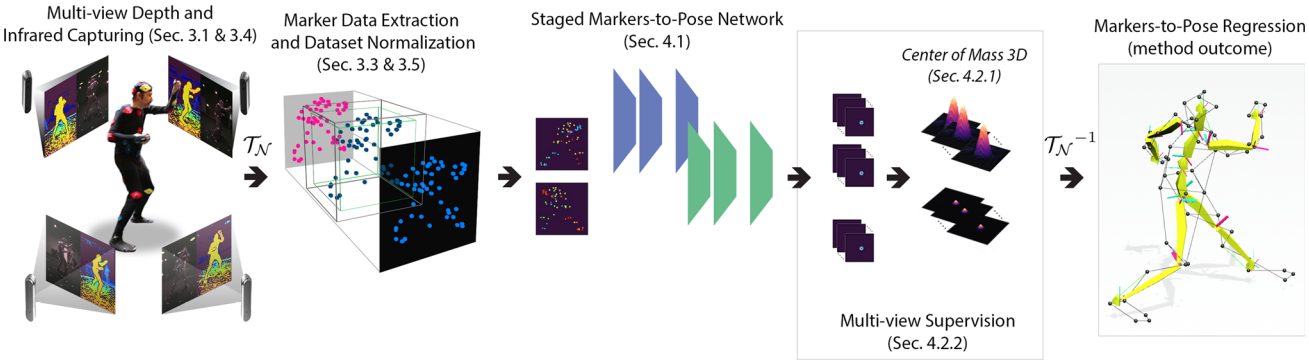

Fig. 4 Using a multi-view setting with an arbitrary number of spatio-temporally aligned depth-infrared cameras perimetrically placed around a subject with reflective markers attached on the body, we capture the body movements. We detect the markers exploiting markers' intense reflections provoked on the infrared images and sensors' depth perception. Rendering the 3D markers after normalization on two opposing depth images, we train a FCN to jointly and sequentially predict marker and joint heatmaps, decoding then with a novel fully differentiable module the 3D markers and joint positions. At run-time, we conduct a two-step forward pass, where the first stages localize the markers and the latter stages estimate the body pose (we illustrate the two-view example of our multi-view input/supervision concept for the sake of brevity)

spatial 3D regression to precisely regress the normalized 3D coordinates both of $M = 53$ markers, $\hat{\mathbf{X}}_M \in \mathbb{R}^{M\times 3}$, and $J = 19$ joints, $\hat{\mathbf{X}}_J \in \mathbb{R}^{J\times 3}$. Finally, we apply the inverse scale and translation transformation ${\mathcal{T}_{\mathcal{N}}}^{-1}$ (Sect. 3.5) to recover the marker and pose 3D coordinates to their original physical dimensions.

4.1 Staged Markers-to-Pose Networks

4.1.1 Network Architecture

Our approach is based on a particular concept with respect to the network design. We propose multi-stage FCN architectures with smooth staging from markers-to-pose heatmap predictions that lead to better performing, efficiently designed models, as proved and discussed in Sect. 5.3.1. Given that the prior of the final pose is the optical marker spatial distribution, we design our networks to predict/refine the marker coordinates at an early stage, letting the joint coordinates to be localized from the latter stages. That way, the prior is refined before localizing the joint coordinates, resulting in robust and reliable predictions.

We build upon highly effective heatmap prediction networks such as Convolutional Pose Machines (CPM) by Wei et al. (2016), Stacked Hourglass (SH) by Newell et al. (2016) and a more recent one, HRNET by Wang et al. (2020). We predict dual-view heatmaps by feeding $\mathcal{D}_{front}$ and $\mathcal{D}_{back}$ to the networks in two separate forward passes. Hence, our models process each of the views, while both inferences are later fused to produce a final prediction supervised by a shared objective.

We follow the same architecture design for all networks. At first, we feed each $\mathcal{D}_v$, $v = \{front, back\}$ depth map to an initial pre-processing module to extract a feature map $\mathbf{F}_v$ ($\mathbf{F}$ for the sake of brevity). We design a $2K$-stage network, $K \in \mathbb{N}$, and we split it in two super-stages consisting of K stages each. The former super-stage predicts the marker heatmaps $\bar{\mathbf{H}}_M$ and the latter the joint heatmaps $\bar{\mathbf{H}}_J$, resulted as aggregations of the intermediate heatmaps $\mathbf{H}_{S,s_t}$ predicted by each stage $s_t \in \{1, 2, \ldots, 2K\}$ of each super-stage $S \in \{M, J\}$. The feature map $\mathbf{F}$ is concatenated with $\mathbf{H}_{S,s_t}$, $s_t < 2K$ at every stage to feed the next one.

Rather than supervising heatmaps as originally proposed by the authors of the networks (per stage intermediate supervision in CPM and SH and last stage heatmap supervision in HRNET), we supervise only the aggregated heatmaps $\bar{\mathbf{H}}_M$ and $\bar{\mathbf{H}}_J$ with the coordinates and structural heatmaps of the respective ground truth joints. As further discussed in Sect. 5.3, this aggregation scheme converges faster and orchestrates slightly better the staged transition from dense observations to sparse marker and joint coordinate regression, as also used and validated by Zanfir et al. (2018).

Network details along with a high-level overview sketch (Fig. 12) of the proposed architectures are presented in "Appendix A".

4.1.2 Heatmap Prediction

Let $\mathbf{H}^k_{s_t}$, $s_t \in \{1, 2, \ldots, 2K\}$ denote the k-th latent heatmap before *Softmax* at stage s_t. Then, heatmap aggregation across stages is performed via summation:

$$\bar{\mathbf{H}}^k = \sum_{s_t=f}^{l} \mathbf{H}^k_{s_t} \tag{3}$$

In total, two aggregations are being performed, one for marker position regression with $f = 1$ and $l = K$ and one for joint position regression with $f = K + 1$ and $l = 2K$.

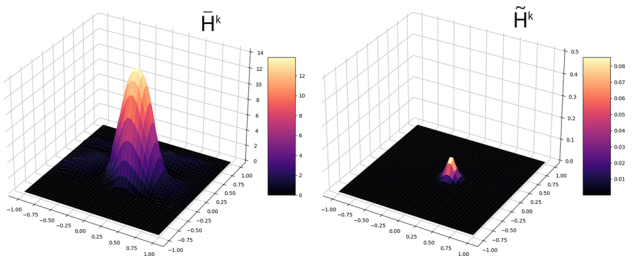

Fig. 5 Surface plotting of the predicted heatmaps before ($\bar{\mathbf{H}}^k$) and after *Softmax* ($\tilde{\mathbf{H}}^k$). We let the network predict latent heatmaps satisfying both tasks, i.e. the average of $\bar{\mathbf{H}}^k$ values to be equal to the z-coordinate of the 3D keypoint, while after Softmax, the heatmap $\tilde{\mathbf{H}}^k$ to approach a gaussian distribution for xy-coordinate estimation

By applying *Softmax* on each of the aggregated heatmaps $\bar{\mathbf{H}}^k$ results in $\tilde{\mathbf{H}}^k$:

$$\tilde{\mathbf{H}}^k(\mathbf{p}) = \frac{e^{\bar{\mathbf{H}}^k(\mathbf{p})}}{\sum_{\mathbf{p}\in\Omega_k} e^{\bar{\mathbf{H}}^k(\mathbf{p})}} \tag{4}$$

where $\mathbf{p}$ denotes a heatmap layer pixel and Ω_k the spatial domain of the heatmaps. Visualizations of $\bar{\mathbf{H}}$ and $\tilde{\mathbf{H}}$ heatmaps are illustrated in Fig. 5.

4.2 Multi-view Spatial 3D Regression

We regress $M = 53$ normalized marker position coordinates $\hat{\mathbf{X}}_M \in \mathbb{R}^{M\times 3}$ and $J = 19$ normalized joint position coordinates $\hat{\mathbf{X}}_J \in \mathbb{R}^{J\times 3}$ by decoding heatmaps $\bar{\mathbf{H}}$ and $\tilde{\mathbf{H}}$ predicted by each corresponding super-stage.

4.2.1 Center of Mass 3D with zMean Layer

We contribute to 3D coordinate regression by proposing Center of Mass in 3-dimensional space (*CoM3D*) with the insertion of a fully differentiable *zMean* layer to the well-known CoM coordinate decoding from heatmaps technique. Our statement is that fully convolutional networks can learn to predict heatmap distributions in varying value ranges, underlying an extra spatial information layer that allows the encoding of the third dimension. Thus, we regress 3D coordinates with the introduction of *CoM3D* combining two fully differentiable layers, $zMean$ and CoM, ($zMean + CoM = CoM3D$), with $zMean$ being our proposed contribution. While for CoM we follow the standard procedure introduced by Tensmeyer and Martinez (2019) for x and y coordinate regression, the motivation behind $zMean$ is to exploit one extra degree of freedom which *Softmax* allows under direct heatmap supervision, in order to additionally constrain the average of $\bar{\mathbf{H}}^k$ to approach a ground-truth z coordinate, leading to a compact 3D coordinate encoding. In detail, let (x_k, y_k, z_k) denote the predicted normalized 3D coordinates for marker or joint k, with k being either in $\{1, \ldots, M\}$ or in $\{1, \ldots, J\}$, respectively. Then, we regress z_k by:

$$z_k = \frac{1}{N_x N_y} \sum_{\mathbf{p}\in\Omega_k} \bar{\mathbf{H}}^k(\mathbf{p}) \tag{5}$$

and $(x, y)_k$ by:

$$(x, y)_k = \left(\frac{1}{N_x}, \frac{1}{N_y}\right) \circ \sum_{\mathbf{p}\in\Omega} \tilde{\mathbf{H}}^k(\mathbf{p}) \cdot \mathbf{p} \tag{6}$$

with $N_x = 40$, $N_y = 40$ the cardinality of each 2D heatmap pixel coordinate domain, as designed to our network architectures, and $\circ$ denoting element-wise multiplication.

4.2.2 Multi-view Supervision

Finally, for two opposing rendering views, we conduct joint dual-view supervision by estimating one single 3D point per dual input rotating the normalized coordinate prediction $(x_{k,back}, y_{k,back}, z_{k,back})$ for $\mathcal{D}_{back}$ by 180° around the Y-axis and averaging it with the normalized coordinate prediction $(x_{k,front}, y_{k,front}, z_{k,front})$ for $\mathcal{D}_{front}$ by:

$$(\hat{x_k}, \hat{y_k}, \hat{z_k}) = \begin{pmatrix} \frac{1}{2}(x_{k,front} + (1 - x_{k,back})), \\ \frac{1}{2}(y_{k,front} + y_{k,back}), \\ \frac{1}{2}(z_{k,front} + (1 - z_{k,back})) \end{pmatrix}. \tag{7}$$

That way, we approach every single 3D coordinate from two opposing sides covering the 3D volume where the human body is contained, regressing $\hat{\mathbf{X}}_M$ and $\hat{\mathbf{X}}_J$ normalized marker and joint 3D coordinates, correspondingly. In other words, our model learns to predict heatmaps whose average value approaches the normalized ground-truth "z" coordinate while their normalized 2D center of mass, after *Softmax*, approaches the normalized ground-truth "x" and "y" coordinates.

4.3 Losses

During training, we jointly supervise $\hat{\mathbf{X}}_M$ and $\hat{\mathbf{X}}_J$ with the ground truth coordinates, $\hat{\mathbf{M}}_{gt}$ and $\hat{\mathbf{J}}_{gt}$, respectively. On the one hand, contrary to DSNT proposed by Nibali et al. (2018), instead of using Euclidean Distance loss extended in 3D, we use Wing loss, $\mathcal{L}_{wing}$, proposed by Feng et al. (2018) which leads to a better-learnt data representation. Wing constitutes a loss function which behaves as a *log* function with an offset for small errors, while for larger errors as L1. $\mathcal{L}_{wing}$ loss function is defined by:

$$\mathcal{L}_{wing}(x) = \begin{cases} w \ln(1 + |x|/\epsilon) & if |x| < w \\ |x| - C & otherwise \end{cases} \tag{8}$$

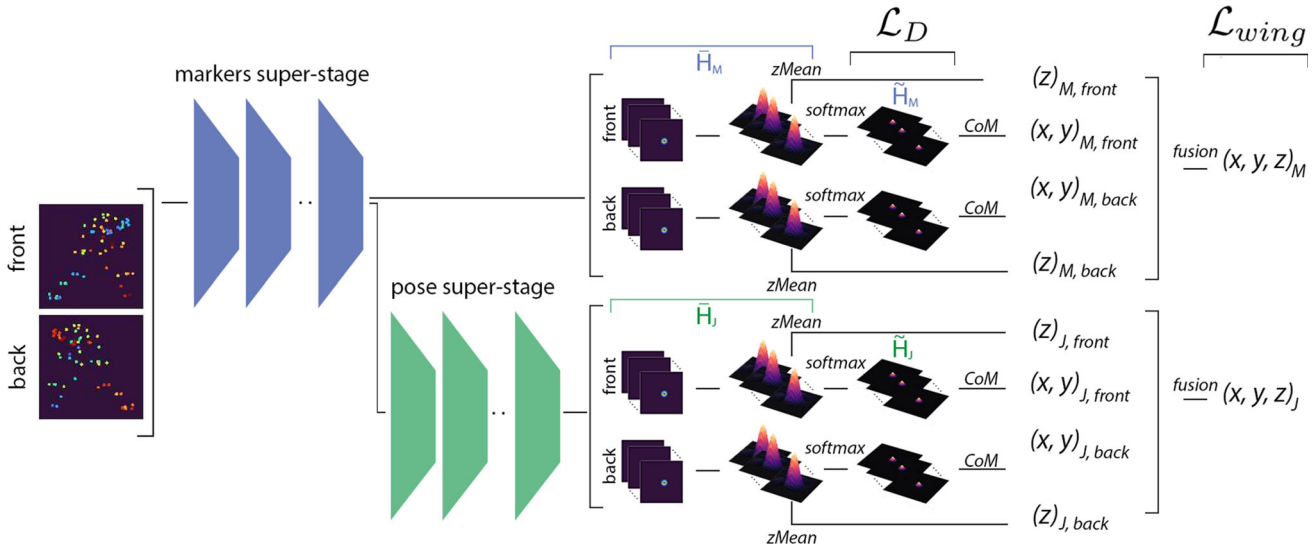

Fig. 6 A high-level diagram of DeMoCap that depicts the various steps of our concept. The rendered depth multi-view input is fed to the former marker super-stage that, sequentially, feeds the latter pose one, both predicting the $\bar{\mathbf{H}}$ marker and joint heatmaps, correspondingly, resulting to $\tilde{\mathbf{H}}$ after *Softmax*. Applying *CoM3D*, we decode z coordinate from $\bar{\mathbf{H}}$ with *zMean* and x,y coordinates from $\tilde{\mathbf{H}}$ with *CoM*. Regressing that way the 3D coordinates from each view, we fuse them in the final stage. We supervise both $\tilde{\mathbf{H}}$ and x, y, z marker and joint predictions with $\mathcal{L}_D$ and $\mathcal{L}_{wing}$ respectively, to train the network in an end-to-end manner

where the non-negative w sets the range of the nonlinear part to $(-w, w)$, ϵ limits the curvature of the non-linear region and $C = w - w\ln(1 + w/\epsilon)$ is a constant value that links in a smooth way the piecewise-defined linear and nonlinear parts of the function. The input x to $\mathcal{L}_{wing}$ is the 3D euclidean distance between predicted and ground-truth points of interest.

On the other hand, similarly to DSNT, we directly supervise also the spread of the heatmap since the strongly supervised pixel-wise gradients enhance the training of the model, improving its performance. We impose strict regularization on latent heatmap to directly drive it towards a certain shape and distribution. More specifically, we force the heatmaps to resemble spherical Gaussians by minimizing the divergence between generated heatmaps and targeted gaussian distributions centered at the 2D orthographic projections $\mathbf{p}_{gt}$ of the normalized ground truth 3D positions from the respective viewpoint. The distribution regularization term is defined by:

$$\mathcal{L}_D(\tilde{\mathbf{H}}, \mathbf{p}_{gt}) = D(\tilde{H}||\mathcal{N}(\mathbf{p}_{gt}, \sigma^2\mathbf{I}_2)) \tag{9}$$

where $D(\cdot||\cdot)$ is the Jensen–Shannon divergence (Fuglede and Topsoe (2004)), σ denotes the target variance and $\mathcal{N}()$ the target normal distribution.

Finally, the total loss used to compute the network gradients is defined as:

$$\mathcal{L}_{total} = \lambda_1(\mathcal{L}_{wing,M} + \mathcal{L}_{wing,J}) + \lambda_2(\mathcal{L}_{D,front} + \mathcal{L}_{D,back}), \tag{10}$$

where λ_1, λ_2 are hyper-parameters that weight the coordinates and heatmap distribution losses, respectively. A high-level diagram that depicts the various steps of our concept is illustrated in Fig. 6.

5 Experimental Evaluation

In this section, we present the experiments we conducted to assess our method. In Sect. 5.1, we present the dataset created according to the pre-processing steps we described in Sect. 3 and used for training, validating and testing our model. Then, we discuss the evaluation methodology we followed by presenting the metrics we used (Sect. 5.2.1), the state-of-the-art methods we compared against ours (Sect. 5.2.2) and the implementation details for the execution of these experiments (Sect. 5.2.3). In Sect. 5.3, we present and discuss quantitative and qualitative experimental results, giving insights with respect to the performance of our model. Finally, an ablation study gives evidence regarding the necessity and impact of our contributions in Sect. 5.4.

5.1 Dataset

Considering the data capturing and pre-processing steps described in Sect. 3, we create a set of 12,197 samples from 11 single-person activities. We divide the subjects $S1$, $S2$, $S3$, $S4$ into two male-female couples using the data of the first couple ($S3$ and $S4$) performing 7 of the 11

Table 1 Training, validation and testing datasets number of samples, activities and subjects involved

Activity	Samples	Set	Subjects
Running	639	Train	{S3,S4}
Basketball_dribbling	666		
Sitting_down	1,205		
Object_dropping_n_picking	754		
Stretching_n_talking	1,201		
Watching_scary_movie	826		
In-flight_safety_announc.	2,874		
Total train samples	*8,165*		
Jumping_jack	692	val	*{S1,S2}
Bending	851		
Total val samples	*1,543*		
Punching_n_kicking	930	test	*{S1,S2}
Sitting_on_a_stool	1,112		
Total test samples	*2,042*		

activities for training (*running*, *basketball_dribbling*, *sitting_down*, *object_dropping_n_picking*, *stretching_n_talking*, *watching_scary_movie* and *in-flight_safety_announcement*) and the data from the second one ($S1$ and $S2$) performing the remaining ones for validation (*jumping_jack* and *bending*) and testing (*punching_n_kicking* and *sitting_on_a_stool*). We split our dataset that way to assess the models on unseen subjects with different body structures and unseen activities, providing reliable and fair conclusions with respect to their performance.

The training, validation and testing data sets consist of 8165, 1990 and 2042 samples, as presented in Table 1.

5.2 Methodology

5.2.1 Metrics

We measure the errors of the methods in physical dimensions by applying the inverse scale and translation transformation $\mathcal{T}_{\mathcal{N}}{}^{-1}$, as described in Sec. 3, with the use of the metrics below:

- We assess the 3D pose regression accuracy by using the commonly used mean per joint position error (M_{PJPE}) (Li et al. (2015)) and mean per marker position error (M_{PMPE}) when applicable, i.e. when markers are predicted by the models.
- In a similar fashion, we also measure the Root Mean Squared Per Joint and Marker Position errors (RMS_{PJPE} and RMS_{PMPE}) which constitute variations of M_{PJPE} and M_{PMPE}, respectively, based on Root Mean Square Error (RMSE) instead of Mean Absolute Error (MAE), as used by Chatzitofis et al. (2020). Both metrics are affected by large outliers but RMS_{PJPE} and RMS_{PMPE} incorporate better the variance of the predictions and their bias.
- We use mean Average Precision (mAP) using Percentage of Correct Keypoints 3D (PCK3D) metric proposed by Yang and Ramanan (2011) in a range of α_{3D} error thresholds.
- Beyond the position-based metrics, we present results calculated by fusing the pose data (forward direction of the bones) and the orientation driven by the various marker groups mapped to joints, providing an extra metric with respect to the stability of the predictions and the capabilities provided by the simultaneous regression of markers and joints. We consider the mean and root mean per joint angular errors, M_{PJAE} and RMS_{PJAE}, by measuring the angle θ in degrees, between the ground truth and predicted joint orientations by:

$$\theta = cos^{-1}(2\langle \hat{q}_{j,gt}, \hat{q_j}\rangle^2 - 1) \tag{11}$$

where $\langle \hat{q}_{j,gt}, \hat{q}\rangle$ denotes the inner product between $\hat{q}_{j,gt}$ and $\hat{q_j}$ of joint j.

It is worth noting that for the sake of comparability against other methods, we use 17 out of the 19 total joints of the regressed pose for the assessment, excluding the toes.

5.2.2 Comparison against State-of-the-art Methods

Due to the lack of public state-of-the-art methods targeting this specific task, i.e. markers and pose regression from noisy optical marker data, relevant methods were identified and retrained to adapt in our dataset, offering valid comparisons. We identify *marker-based* and *markerless* methods; for the former, the input is $\hat{\mathbf{M}}_r$, i.e. the normalized sparse cloud of the detected markers, while for the latter, we use the spatio-temporally aligned multi-view colored infrared images along with the camera intrinsic and extrinsic parameters.

In detail, we compare our model against two marker-based methods, an adaptation of a graph-based model designed for image-based hand-object pose estimation proposed by Doosti et al. (2020) (HOPE) adapted to our task and a direct 3D regression method used for online marker labeling (OML) adapted for simultaneous marker-joint 3D coordinate regression from markers by feeding a single and dual marker depth map (Han et al. (2018)). On top of that, we further assess three markerless methods, two top-down pose estimation methods from spatio-temporally aligned multi-view color images relying on the concept of learnable triangulation (LT), proposed by Iskakov et al. (2019) and one bottom-up multi-view and graph-based pose estimation approach (4DA) from

spatio-temporally aligned multi-view color images proposed by Zhang et al. (2020b).

Marker-based methods: *HOPE* is a lightweight model designed to jointly estimate hand and object pose in 2D and 3D space based on a cascade of two adaptive graph convolutional neural networks. We adapt the first one to estimate 2D coordinates of the joints on the orthographic depth maps, followed by the second, Adaptive Graph-U-Net (Gao and Ji (2019)), to convert 2D to 3D coordinates. We also modify the input to our depth map resolution ($1 \times 160 \times 160$) instead of the color images of the original work and we train the model from scratch with *Xavier* (Glorot and Bengio (2010)) weight initialization.

As in HOPE, we adapt *OML* to the new input depth map resolution (160×160 instead of 52×52 of the original work), initializing the model weights with *Xavier* initialization. Furthermore, we adapt the output of the network to predict a vector with $M = 53$ and $J = 19$ 3D positions, i.e. the target of our task, while we present an extra variation of the approach that consumes multi-view depth data similarly to our concept.

Markerless methods: With respect to the *LT* methods, $\text{LT}_{(alg.)}$ is based on algebraic triangulation with learnable camera-joint confidence weights, while $\text{LT}_{(vol)}$ constitutes a volumetric triangulation approach based on dense geometric aggregation of 2D heatmap predictions from multiple viewpoints. The input used to train these models is a batch of N spatio-temporally aligned color images along with the corresponding camera poses and intrinsic parameters. Aiming at a fair comparison between LT methods and DeMoCap, we re-train the LT models on our dataset initializing with the pre-trained weights due to the domain differences between the datasets.

For the initial bounding box detection, LT methods use Mask R-CNN 2D detector (He et al. (2017)) with ResNet-152 (He et al. (2016)) backbone to predict the human bounding boxes, however we use the ground truth bounding boxes to avoid the need for re-training of the detector.

We use the weights of the pre-trained models[3] trained on Human3.6 (Ionescu et al. (2013)) with the 2D backbone pre-trained on COCO dataset (Lin et al. (2014)) and we further train them on our colored infrared dataset for 10 epochs adopting the training configuration proposed by the authors. We use the colored infrared data to re-train the models instead of the sparse data renderings which would require training of the models from scratch. Note that the predictions obtained from the algebraic are used for the volumetric triangulation approach. For training, we use the official repository and guidelines provided by the authors[4].

4DA considers the temporal aspect of consecutive frames. With the use of OpenPose by Cao et al. (2017), human body part candidates (heatmaps) and connection confidence (part affinity fields) scores between body parts are retrieved from each single view. Fusing the obtained human body part features between two sequential frames, a 4D graph is constructed with per-view parsing edges connecting adjacent body parts, cross-view matching edges connecting the same body part across the various views, and temporal tracking edges for mapping detected 3D nodes on a previous frame with new 2D detections on the next one. With the same practice we followed for LT, we re-train the model for 10 epochs, initializating it with the weigths of the pre-trained model of the official repository of OpenPose[5].

5.2.3 Implementation details

We train our network for 200 epochs using *Adam* (Kingma and Ba (2014)) optimizer with an initial learning rate equal to $1e-4$, while we apply a frequent linear rolling drop of 0.95 every 4 epochs. The batch size is 16, and the heatmap standard deviation during supervision is $\sigma = 1.0$. λ weights of $\mathcal{L}_{wing}$ and $\mathcal{L}_D$ losses, as defined in Eq. 10, are set to $\lambda_1 = 2$ and $\lambda_2 = 1$, and the parameters for the wing loss $\mathcal{L}_{wing}$ are set to $w = 10$ and $\epsilon = 2$.

The model is implemented with PyTorch (Paszke et al. (2019)) and moai (2021) and the experiments ran on a conventional computer with 1 single NVIDIA GTX 1080 Ti graphics card of 12 GB RAM, and an Intel i7(R) processor, using the same manual seed for all epxeriments for fair comparison and reproducibility. The code and the dataset are publicly available online[6].

5.3 Experimental Results

The validation set was used for training hyper-parameter tuning, while the evaluation on the test set took place after the selection of the best models. We present results on both sets, showing the variance of the inference accuracy between them for the various models, indicating their generalization potential.

[3] The models that were publicly available in the official repository of the authors at the time of this work.

[4] https://github.com/karfly/learnable-triangulation-pytorch/tree/9d1a26ea893a513bdff55f30ecbfd2ca8217bf5d

[5] https://github.com/CMU-Perceptual-Computing-Lab/openpose/tree/1f1aa9c59fe59c90cca685b724f4f97f76137224

[6] https://github.com/tofis/democap

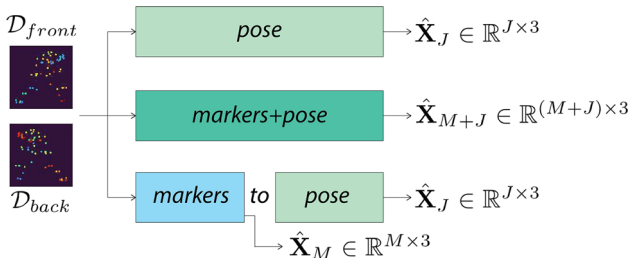

Fig. 7 We design all FCN architectures in 3 variations to assess our staged markers-to-pose concept. We feed $\mathcal{D}_{front}$ and $\mathcal{D}_{back}$ to all variations with ***pose***, ***markers+pose*** and ***markers-to-pose*** yielding joints only ($\hat{\mathbf{X}}_J \in \mathbb{R}^{J\times 3}$), simultaneously markers and joints ($\hat{\mathbf{X}}_{M+J} \in \mathbb{R}^{(M+J)\times 3}$), and sequentially markers ($\hat{\mathbf{X}}_M \in \mathbb{R}^{M\times 3}$) and joints ($\hat{\mathbf{X}}_J \in \mathbb{R}^{J\times 3}$), correspondingly

5.3.1 Assessment with various FCN networks

At first, we assess DeMoCap by building our staged concept upon various FCN architectures, as discussed in Sec. 4.1. The reason is twofold; i) to validate our position that staged architectures for markers-to-pose predictions perform better independently of which FCN architecture used to predict the heatmaps and ii) to select the best performing model for the comparison against state-of-the-art. We train DeMoCap building upon CPM, SH and HRNET in three variations each:

- ***pose***: We train the models in an end-to-end design, regressing and supervising only the pose heatmaps with markers absent, though focusing on one single task, the prediction of joint 3D positions $\hat{\mathbf{X}}_J \in \mathbb{R}^{J\times 3}$.
- ***markers+pose***: Similarly to *pose* variation, we train the models in an end-to-end concept, however regressing and supervising both the joint and the marker heatmaps in every forward pass, resulting in $\hat{\mathbf{X}}_{M+J} \in \mathbb{R}^{(M+J)\times 3}$.
- ***markers-to-pose***: We train DeMoCap on its original concept where the first super-stage predicts the marker, and the second one the pose only heatmaps, resulting sequentially in $\hat{\mathbf{X}}_M \in \mathbb{R}^{M\times 3}$ and $\hat{\mathbf{X}}_J \in \mathbb{R}^{J\times 3}$, correspondingly.

The outcomes of these experiments are illustrated in Table 2. The *markers-to-pose* staged approach achieves higher performance for all models on the main task of the method, i.e. the estimation of the pose, however, in some experiments, the markers are more precisely localized by the *markers+pose* variation. For cooperative marker and pose regression, which also enables joint 3D rotation estimation, the computation load per stage and the network weights are the optimum for the *markers-to-pose* approach, since, for *markers+pose*, the marker and joint heatmaps are predicted across all the stages of the network.

With respect to the models, although SH performs remarkably in the testing set outperforming HRNET, we consider the latter for the rest of the experiments since our indications were based on the experimental results retrieved in the validation set.

5.3.2 Quantitative Analysis

We compare DeMoCap$_{HRNET-8-\{markers-to-pose\}}$, referring to it as DeMoCap for the sake of brevity, against other methods (Sect. 5.2.2) presenting and discussing total, per joint and per action quantitative results.

Total Results. In Table 3, we depict the results of the methods with the use of M_{PJPE}, RMS_{PJPE}, M_{PMPE}, RMS_{PMPE}, $\mathrm{mAP}_{50\mathrm{mm}}$, M_{PJAE} and RMS_{PJAE} metrics.

The markerless multi-view color-based pose estimation models 4DA and LT are effective showing their ability to estimate 3D poses from multiple spatio-temporally aligned color views, being relatively robust to body part occlusions and partial views. As presented in the original work by Iskakov et al. (2019), the volumetric approach performs better in our dataset, showing greater accuracy than the algebraic one.

Our method yields more reliable and accurate predictions than 4DA and LT across all metrics with respect to the total results presenting lower M_{PJPE}, greater $\mathrm{mAP}_{50\mathrm{mm}}$ and lower RMS_{PJPE}, despite the fact that these models have been trained with far larger datasets than ours. That is due to the major differences of DeMoCap against multi-view methods built to operate with dense visual data. DeMoCap is trained to predict markers and pose from sparse marker data where the input is exclusively related to the human body pose in 3D space. There is no context redundancy including various backgrounds, cloth types, fabric or colors, lighting conditions or any other aspect as happening for deep models trained on dense visual streams. In other words, the input for DeMoCap is not domain sensitive as color, depth or any other dense visual stream is. On top of that, DeMoCap is trained on purely 3D data, while LT and 4DA are based on the fusion of multiple partial 2D detections, weakening the localization of the final 3D keypoints. Despite the trial to reduce the weights of erroneous 2D predictions with learnable weighting or graph association techniques before the final 3D fusion, the errors cannot be totally eliminated, leading to erroneous estimates when body parts are exceeding the field of view of at least one of the cameras or being occluded resulting in malicious predictions. For DeMoCap, the disappearance of a marker from one of the views is not considered enough to drive the model in failure. At least one marker detection from one single depth camera, a highly possible case when multiple depth sensors are capturing the performance, can be enough to lead the model to accurate prediction given the low variance of the 3D input.

Despite our efforts to finetune the HOPE and OML models, their performance is relatively low, struggling to

Table 2 M_{PJPE}, RMS_{PJPE}, M_{PMPE}, RMS_{PMPE}, mAP_{50mm}, M_{PJAE} and RMS_{PJAE} results between DeMoCap with CPM, SH and HRNET in three variations each, pose, *marker+pose* and markers-to-pose

DeMoCap \ Metrics (mm/°)	Set	$M_{PJPE}\downarrow$	$RMS_{PJPE}\downarrow$	$M_{PMPE}\downarrow$	$RMS_{PMPE}\downarrow$	$mAP_{50mm}\uparrow$	$M_{PJAE}\downarrow$	$RMS_{PJAE}\downarrow$
CPM-6-{pose}	val	41.45	53.96	-	-	81.72%	-	-
CPM-6-{marker+pose}		40.05	50.00	53.21	65.06	79.33%	20.71	24.37
CPM-6-{markers-to-pose}		38.72	48.26	51.46	62.89	82.31%	20.10	23.60
SH-8-{pose}		38.60	49.67	-	-	82.76%	-	-
SH-8-{marker+pose}		38.55	51.68	42.30	54.44	85.35%	17.70	22.41
SH-8-{markers-to-pose}		36.94	51.42	44.84	58.70	87.49%	18.19	23.55
HRNET-8-{pose}		34.44	42.97	-	-	87.88%	-	-
HRNET-8-{marker+pose}		34.48	43.05	**40.42**	**49.57**	89.50%	**16.91**	**20.23**
HRNET-8-{markers-to-pose}		**33.83**	**42.65**	42.33	51.74	**90.41%**	18.66	22.47
CPM-6-{pose}	test	45.96	58.75	-	-	83.45%	-	-
CPM-6-{marker+pose}		46.28	58.21	62.83	77.91	83.20%	19.18	24.41
CPM-6-{markers-to-pose}		45.14	57.13	61.14	76.11	84.97%	18.22	23.32
SH-8-{pose}		40.40	51.05	-	-	88.39%	-	-
SH-8-{marker+pose}		39.02	49.90	51.13	64.46	87.86%	15.81	21.11
SH-8-{markers-to-pose}		**38.45**	**48.31**	**47.10**	**59.21**	**89.32%**	**15.08**	**20.00**
HRNET-8-{pose}		40.99	53.05	-	-	87.51%	-	-
HRNET-8-{marker+pose}		40.89	52.42	51.88	65.47	87.37%	19.19	22.18
HRNET-8-{markers-to-pose}		40.04	51.69	52.92	66.49	88.05%	19.73	26.18

Table 3 M_{PJPE}, RMS_{PJPE}, M_{PMPE}, RMS_{PMPE}, mAP_{50mm}, M_{PJAE} and RMS_{PJAE} results between HOPE by Doosti et al. (2020), OML by Han et al. (2018), LT by Iskakov et al. (2019), 4DA by Zhang et al. (2020b) and our method are presented. For the sake of clarity, **C** for *Color* and **M** for *Marker* data input with cell colorization indicate the markerless and marker-based methods, respectively

Method \ ***Metrics*** (mm/°)	Set-In	$M_{PJPE}\downarrow$	$RMS_{PJPE}\downarrow$	$M_{PMPE}\downarrow$	$RMS_{PMPE}\downarrow$	$mAP_{50mm}\uparrow$	$M_{PJAE}\downarrow$	$RMS_{PJAE}\downarrow$
Zhang et al. (2020b)	val-C	51.34	63.19	-	-	62.15%	-	-
Iskakov et al. (2019)$_{(alg.)}$		46.91	54.70	-	-	71.55%	-	-
Iskakov et al. (2019)$_{(vol.)}$		43.76	49.66	-	-	83.60%	-	-
Doosti et al. (2020)$_{(adapt.)}$	val-M	124.14	144.42	-	-	0.0%	-	-
Han et al. (2018)$_{(adapt.)}$		113.36	137.61	129.92	151.49	7.45%	38.02	46.17
Han et al. (2018)$_{(dual-adapt.)}$		108.66	131.19	124.15	146.34	8.88%	34.09	42.33
DeMoCap		**33.83**	**42.65**	**42.33**	**51.74**	**90.41%**	**18.66**	**22.47**
Zhang et al. (2020b)	test-C	50.36	76.26	-	-	61.41%	-	-
Iskakov et al. (2019)$_{(alg.)}$		49.57	70.31	-	-	75.95%	-	-
Iskakov et al. (2019)$_{(vol.)}$		46.69	64.91	-	-	81.68%	-	-
Doosti et al. (2020)$_{(adapt.)}$	test-M	115.50	136.54	-	-	3.57%	-	-
Han et al. (2018)$_{(adapt.)}$		97.71	121.36	122.52	138.19	21.64%	28.40	35.50
Han et al. (2018)$_{(dual-adapt.)}$		93.70	111.95	112.39	131.96	18.51%	27.13	33.13
DeMoCap		**40.04**	**51.69**	**52.92**	**66.49**	**88.05%**	**19.73**	**26.18**

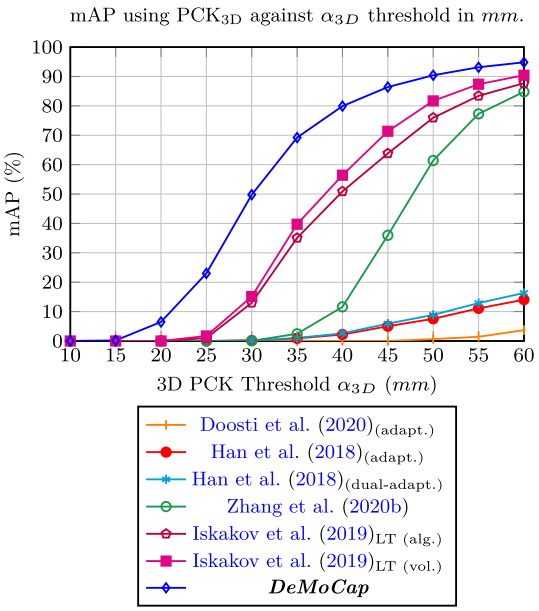

Fig. 8 A plot comparison between Han et al. (2018)$_{OML}$, Iskakov et al. (2019)$_{LT}$ and DeMoCap showing mAP using PCK3D metric against α_{3D} threshold in millimeters in the test set. Our results reach high scores at low α_{3D} thresholds showcasing the effectiveness of our method

accurately regress the 3D coordinates in samples from unseen subjects and activities.

Their low performance could be potentially explained by the use of direct regression with the use of fully connected layers (fully connected and graph layers for HOPE) in the network and our relatively small training dataset, contrary to the dataset size of the original works and the use of pre-trained models which were not applicable on depth map input. Moreover, our 3D coordinate regression task is more challenging since we regress $M + J = 72$ 3D coordinates in comparison with HOPE and OML designed to regress less than 30 3D keypoints. Nevertheless, it is worth noting that the dual-view trained OML presents relatively better results than the single-view model, showcasing the potential of the multi-view supervision concept.

A more comprehensive analysis with respect to the performance of the methods in the testing set is illustrated in Fig. 8, where we compare the methods with mAP metric using PCK 3D against α_{3D} threshold in mm. DeMoCap outperforms the rest of the methods with higher mAP against the whole range of α_{3D} thresholds, while for $\alpha_{3D} = 35$ mm, the mAP is already 70%. OML methods are incapable of being comparable to 4DA, LT and DeMoCap for $\alpha_{3D} \in (0, 60)$ in mm. Finally, both LT models showcase higher precision than 4DA.

Per Joint Results. Beyond the presentation of the total results, we evaluate the performance of the methods on a per human body joint analysis in Table 4, where DeMoCap outperforms the compared methods to most of the body joints.

This analysis allows us to assess the consistency of the models in the estimation of the various joints individually. The rationale behind this analysis is that, traditionally in human pose estimation, the difficulty level for the localization of joints gradually increases while moving from the torso joints of the body, i.e. hips, spines, shoulders, neck, to the head and the end joints of the limbs, i.e. the ankles and the wrists. The latter, being end nodes of an articulated structure, i.e. the human body, move more freely than the rest of the body showing large variance with respect to their global and local body positioning. Nevertheless, we consider it significant for a method that targets human motion capture to present robustness and consistency across all joints estimates. In our experiments, the same challenge applies to all methods, however the errors between end and torso joints for HOPE, OML, 4DA and LT show higher variance than ours, meaning that DeMoCap regresses the pose with more equally balanced accuracy across the joints of the body than the other approaches. Rapid body part movements when fast actions are performed cause image blurriness leading the vision-based models to erroneous estimates. To overcome this challenge, DeMoCap has been explicitly designed to perform the pose regression on two phases. The initial noisy and incomplete marker input is refined/recovered on a first phase, driving the estimation of the pose in a later stage based on refined marker data. The refinement of the markers allow the model to more accurately perform the last stage estimates of the joints, which are only related to the marker positions, without any other contextual binding to the initial blurry color input. We obtain the remarkably lower errors for *Wrists* and *Ankles* joints in Table 4, especially in the testing set where during the totally freely performed *punching_n_kicking* action, many body parts are out of the cameras' field of view for several frames.

Per Action Results. In Table 5, we present and discuss the model outcomes on a per action analysis to assess the model performance across sequences of varying poses. Including the actions both of the validation and testing sets, we present M_{PJPE} for 4 actions: *jumping_jack*, *bending*, *punching_n_kicking* and *sitting_on_a_stool*. Despite the fine-tuning of the models on the validation set, both sets share the same characteristics considering that both include unseen subjects and actions in relation to the training set. Hence, the difficulty level for capturing these motions is determined mostly by the objective challenges of each specific performance such as their complexity and speed, resulting in several occlusions or missing data, or body part movements out of the field of views of the cameras. That is proved from the lower errors in *sitting_on_a_stool* action which demonstrates lower errors than the validation set actions.

For the *punching_n_kicking* action, the subjects were asked to punch and kick in front of the cameras without guidance or other constraints. This resulted in extremely

Table 4 3D Euclidean Distance Error per joint in millimeters

Method \ Joints (mm)	Set-In	Head	Neck	***Shoulders***	Elbows	Wrists	Pelvis	Spines	Hips	Knees	Ankles
Zhang et al. (2020b)	val-C	64.22	27.95	49.54	77.89	97.01	25.47	-	34.00	34.25	33.54
Iskakov et al. (2019)$_{(alg.)}$		32.82	**25.57**	41.08	56.98	68.43	36.12	33.37	45.26	53.22	53.16
Iskakov et al. (2019)$_{(vol.)}$		32.68	25.77	39.43	51.84	61.99	34.82	33.04	42.29	48.54	48.17
Doosti et al. (2020)$_{(adapt.)}$	val-M	103.64	85.76	91.71	155.68	161.11	119.87	82.49	132.99	154.47	122.11
Han et al. (2018)$_{(adapt.)}$		81.76	61.79	88.71	151.48	232.93	91.60	84.38	93.01	98.50	97.00
Han et al. (2018)$_{(dual-adapt.)}$		81.35	60.58	83.50	148.87	219.25	83.27	81.55	90.51	92.70	94.61
DeMoCap		**26.34**	27.06	**35.18**	**35.44**	**40.38**	**26.78**	**30.55**	**28.63**	**35.12**	**42.18**
Zhang et al. (2020b)	test-C	62.25	25.67	50.49	46.31	51.71	38.85	-	47.20	54.82	65.85
Iskakov et al. (2019)$_{(alg.)}$		28.96	20.93	32.63	53.21	65.67	34.51	30.59	54.96	55.69	86.36
Iskakov et al. (2019)$_{(vol.)}$		**28.53**	20.69	31.54	50.23	61.00	**33.18**	**30.21**	51.39	52.15	79.14
Doosti et al. (2020)$_{(adapt.)}$	test-M	98.14	80.23	98.05	128.29	149.24	72.66	68.00	76.08	172.01	164.61
Han et al. (2018)$_{(adapt.)}$		45.35	51.22	80.85	132.28	147.66	73.87	63.56	73.92	117.46	129.59
Han et al. (2018)$_{(dual-adapt.)}$		45.12	50.22	76.38	130.09	138.75	67.15	61.42	71.94	110.25	126.40
DeMoCap		30.09	**19.30**	**23.28**	**28.94**	**45.25**	49.38	31.39	**27.85**	**32.95**	**52.62**

bold-italic indicate bilateral joints for which the average error is presented. For the sake of clarity, **C** for *Color* and **M** for *Marker* data input with cell colorization indicate the markerless and marker-based methods, respectively

Table 5 M_{PJPE} per action results for the validation and testing sets, presenting the performance of the models across different actions. For the sake of clarity, **C** for *Color* and **M** for *Marker* data input with cell colorization indicate the markerless and marker-based methods, respectively

Method \ Action	In	Jumping Jack$_{val}$	Bending$_{val}$	Punching & Kicking$_{test}$	Sitting on stool$_{test}$
Zhang et al. (2020b)	C	51.17	51.52	61.46	37.67
Iskakov et al. (2019)$_{(alg.)}$		44.18	48.63	64.36	34.83
Iskakov et al. (2019)$_{(vol.)}$		41.47	45.22	59.74	33.71
Doosti et al. (2020)$_{(adapt.)}$	M	103.44	143.03	147.82	88.63
Han et al. (2018)$_{(adapt.)}$		101.12	123.87	131.76	69.29
Han et al. (2018)$_{(dual-adapt.)}$		97.34	119.03	124.96	67.71
DeMoCap		**28.74**	**38.47**	**57.19**	**25.77**

challenging data due to the rapid and free-style movements with noisy samples due to blurriness, partial occlusions or body parts out of the field of view of the cameras. This is observed in the per action analysis where the results of all methods on this specific action present the highest errors.

5.3.3 Qualitative Analysis

In this section, we present and discuss qualitative outcomes, as illustrated in Fig. 9. We project the marker and pose 3D coordinates on the infrared views in order to correlate them with the actors' actual performances. For the sake of visualization clarity, we depict the raw noisy input, the ground-truth and the predicted marker data separately. The predicted and ground-truth poses are visualized together to facilitate the visual comparison between them.

Given the infrared images in the background, we highlight the targeted and addressed challenges by our model. In particular, we indicate the marker corrections on the noisy input and the robust pose regression behaviour of the predictions. One can easily observe that the raw marker input captured with the low-cost D415 system is highly noisy. That is due to the marker 3D localization from each sensor separately, which, in combination with the depth sensor errors, result in considerably distant 3D points that represent the 3D position of the same marker. Comparing the regressed marker positions against the initial input, we point the solutions given to this marker-based MoCap problem. Marker observation clustering is achieved by automatically grouping the noisy raw 3D points captured by different sensors (Fig. 9, yellow circles). In this rationale, ghost markers from erroneous detection are ignored (Fig. 9, red circles). Missed markers, either occluded or non-detected, are recovered (Fig. 9, magenta). Image blur-

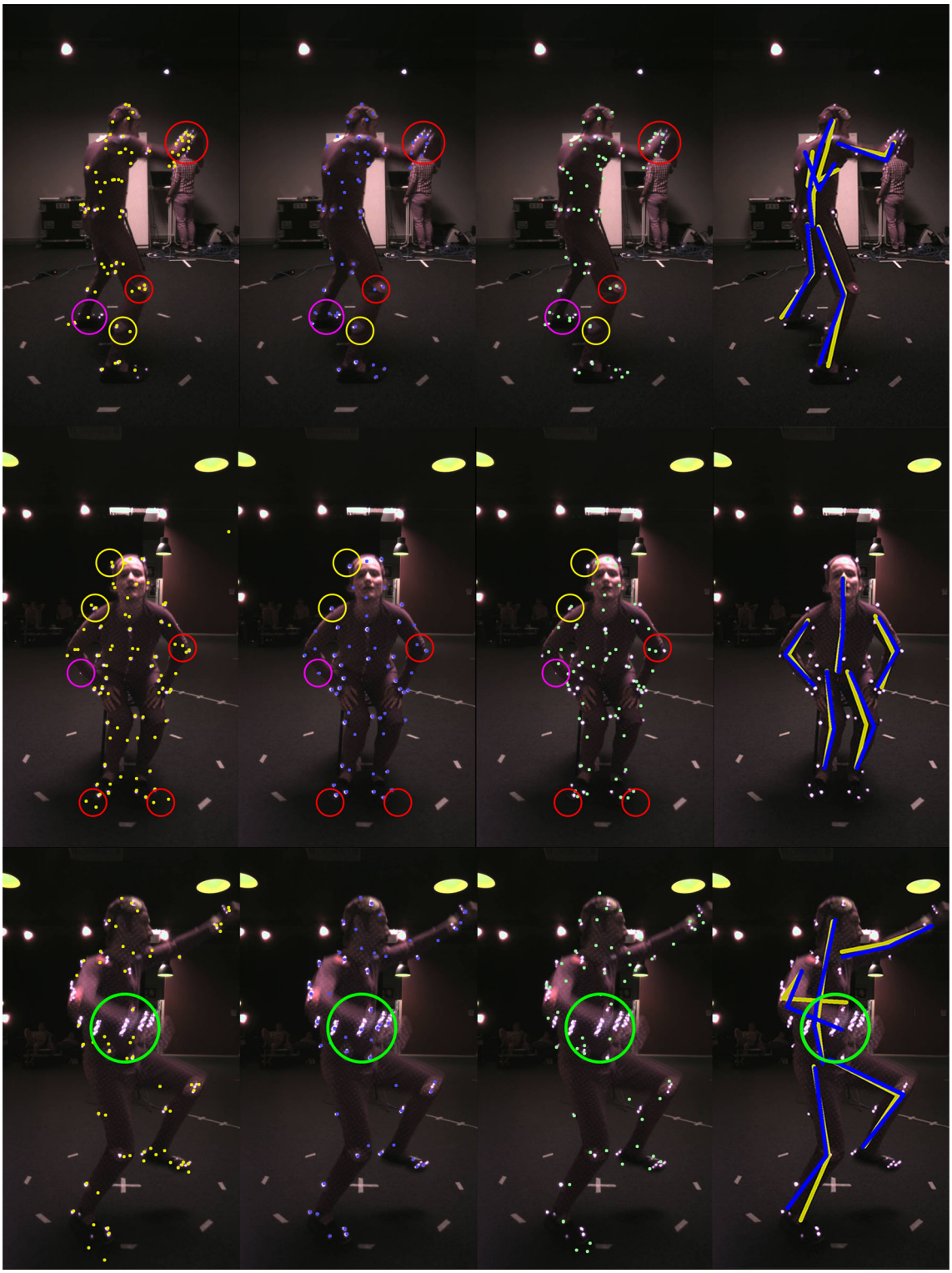

Fig. 9 In the first three columns, we visualize noisy raw (yellow), ground-truth (blue) and predicted (green) markers projected on one single infrared view per group frame (different group frame per row). In the fourth column, we illustrate the predicted poses (yellow) and ground-truth (blue). Red, magenta and yellow circles indicate ghost marker cleaning, recovery of missing markers, and depth sensor errors, correspondingly. Green circles highlight the blurriness issues that our model overcomes

riness are effectively handled to eliminate the joint coordinate regression errors (Fig. 9, green circles). In other words, we accurately localize and label optical markers captured with low-cost sensors that provide highly noisy data. Due to the discrete inference on every single frame, marker swapping present in traditional MoCap is resolved. Our model directly provides instantaneous 3D pose regression from labeled 3D markers without the use of any humanoid prior or knowledge for body structure, meaning that there is no prior information or side inference with respect to bone lengths and joint relative placement, instead, the model predicts the pose in one single-shot. Nevertheless, it is significant to highlight that DeMoCap models are trained based on a specific marker configuration prior, meaning that different marker placement can lead our MoCap model in erroneous predictions.

In Fig. 10, we present more qualitative results with the use of 3D visualizations. We illustrate a batch of 20 samples from the testing set, illustrating the ground-truth data in blue and our predictions in yellow, including failure cases in the last row of the grid.

5.4 Ablation

We conducted and discuss an extensive ablation study to justify the design of the proposed approach. We replace, remove or tune differently one single contribution of our approach per experiment, showcasing its weight to the reader separately. In detail, we ablate:

1. Our newly introduced fully differentiable *CoM3D* module (Sect. 4.2.1) against integral 3D regression module by Sun et al. (2018),
2. The 1- versus dual-view input/supervision (Sect. 4.2.2),
3. The 4- versus dual-view input/supervision (Sect. 4.2.2),
4. The quantization bias between high- (Sect. 3.5) and low-resolution depth rendering of the input,
5. The use of data augmentation (Sect. 3.5),
6. The use of data normalization (Sect. 3.5),
7. The use of intermediate heatmap aggregation (Sect. 4.1) against last stage only heatmap prediction,
8. The inference of our model on marker data captured by 3 cameras only,
9. The inference of our model on marker data captured by 2 opposing cameras only.

The summary of this ablation following the same enumeration is shown in Table 6.

#1. Integral 3D regression versus CoM3D. One of the main contributions of our method is the introduction of the *CoM3D* fully differentiable module for 3D regression comprising a *zMean* layer followed by *Softmax* and a *CoM* layer. The main difference against other spatial regression approaches is the use of *zMean* for z-coordinate regression. In order to assess its value, we train our model substituting *CoM3D* module with integral pose regression proposed by Sun et al. (2018). As shown in Table 6, the use of integral pose regression module is not effective enough in our task resulting in greater errors than the original model (83.68 mm and 89.32 mm against 33.83 mm and 40.04 mm $\mathbf{M}_{PJPE}$ in validation and testing sets, respectively), while the inference time increases noticeably.

Number of rendering views. The input of our model is a pair of depth maps resulted by rendering the 3D positions of the reflective markers from two opposing viewpoints. Conceptually, we take advantage of the three dimensional information as well as its sparsity that allows us to "generate" numerous 2D inputs from one single 3D sample, claiming that multi-view input and supervision yields more robust inference. To assess this claim, we conduct two experiments tweaking the number of rendering views.

#2. 1- versus 2-view depth input. At first, we render only one single depth map and train the network with single-view input. As depicted in Table 6 (exp #2), this model showcases lower performance in comparison with the proposed dual-view approach across all metrics, validating that the multi-view concept drives the model to more accurate and robust predictions.

#3. 4- versus 2-view depth input. On top of that, another experiment with increased number of rendering views is conducted (Table 6 (exp #9)). We train and assess a model with a 4-view input showing favorable comparison against the proposed dual-view original model. In detail, the model performs similarly in the validation set, showing clear outperformance nevertheless in the testing set across all metrics, i.e. approximately 3 mm absolute improvement across all euclidean distance-based error metrics, 1.72% $\text{mAP}_{50\text{mm}}$ and 3.67° and 5.41° M_{PJAE} and RMS_{PJAE}, respectively, given the higher reliability of the results when trained on multiple inputs and based on higher number of multi-view estimates.

The conclusion for these experiments is threefold; firstly, despite the sparsity of the marker point cloud, single-view ambiguities still exist in challenging and complex body poses, resulting in locally dense marker subsets and potential marker occlusions, which multi-view rendering can overcome. Secondly, increasing the number of rendering views, the reliability and accuracy of the predictions is improved, validating our claims with respect to the contribution of multi-view supervision. Finally, increasing the number of rendered depth inputs linearly increases the computational complexity and performance costs of the model. We find the use of two opposing depth renderings ideal as a trade-off between effectiveness and efficiency for deployment, given their comparable results.

#4. High- *versus* low-resolution rendering. We build DeMoCap posing a purely 3D problem as a 3D regression

Fig. 10 Qualitative results of the ground-truth (blue) and predicted (yellow) poses in 3D. Our model regresses 3D poses comparable to ground-truth. In the last row, failure cases are illustrated, when the poses of the subjects are extremely challenging and fast

Table 6 Ablation results. We ablate the contributions of our model one by one to showcase their effectiveness and necessity

	Method \ Metrics (mm/ °)	Set	$M_{PJPE}\downarrow$	$RMS_{PJPE}\downarrow$	$M_{PMPE}\downarrow$	$RMS_{PMPE}\downarrow$	$mAP_{50mm}\uparrow$	$M_{PJAE}\downarrow$	$RMS_{PJAE}\downarrow$
#1.	Integral 3D regression *vs.* CoM3D	val	83.68	96.02	109.77	121.34	24.53%	29.88	33.66
#2.	1- *vs.* 2-view depth input		38.04	46.48	48.67	59.85	85.55%	19.57	22.83
#3.	4- *vs.* 2-view depth input		**33.81**	42.86	44.01	55.30	89.70%	19.48	25.62
#4.	High- *vs.* low-resolution rendering		34.91	43.71	43.69	53.44	89.44%	19.49	23.86
#5.	W/o *vs.* w/ data augmentation		64.82	90.94	85.53	115.81	50.42%	35.44	46.02
#6.	W/o *vs.* w/ data normalization		38.11	47.53	67.99	78.07	87.04%	20.56	23.75
#7.	W/o *vs.* w/ heatmap aggregation		34.73	43.25	48.52	57.74	88.33%	**17.48**	**20.49**
#8.	3 *vs.* 4 capturing depth sensors		46.81	53.50	57.90	66.40	77.17%	22.79	27.22
#9.	2 *vs.* 4 capturing depth sensors		59.86	87.50	73.73	100.88	64.66%	28.08	39.08
	DeMoCap		33.83	**42.65**	**42.33**	**51.74**	**90.41%**	18.66	22.47
#1.	Integral 3D regression *vs.* CoM3D	test	89.32	100.09	116.42	126.91	22.14%	30.89	36.29
#2.	1- *vs.* 2-view depth input		41.87	54.17	56.71	71.44	86.73%	18.98	24.65
#3.	4- *vs.* 2-view depth input		**37.13**	**48.14**	**49.64**	**63.22**	**89.72%**	**16.06**	**20.77**
#4.	High- *vs.* low-resolution rendering		41.50	53.01	53.49	67.06	87.76%	17.29	22.52
#5.	W/o *vs.* w/ data augmentation		63.12	79.83	80.45	100.82	61.36%	24.08	31.93
#6.	W/o *vs.* w/ data normalization		44.19	56.22	72.34	91.28	86.73%	18.64	23.86
#7.	W/o *vs.* w/ heatmap aggregation		40.56	51.97	58.09	72.21	87.76%	16.91	22.10
#8.	3 *vs.* 4 capturing depth sensors		47.58	60.02	61.33	76.30	81.88%	19.25	25.35
#9.	2 *vs.* 4 capturing depth sensors		59.03	74.46	76.73	95.21	67.97%	24.26	33.09
	DeMoCap		40.04	51.69	52.92	66.49	88.05%	19.73	26.18

from multiple 2.5D inputs mission, allowing the use of 2D fully convolutional architectures with proved and remarkable effectiveness in keypoint localization tasks. That way though, we pay the cost of encoding 3D non-quantized data to quantized 2D grids, leading to some loss of information. Specifically, the rendered depth maps based on quantized coordinates are not totally accurate leading to sub-optimal supervision and degraded model performance. We limit the quantization error and information loss by rendering the depth images in high pixel resolution, i.e. 800 × 800, and linearly interpolate them to our input size, i.e. 160 × 160.

In Table 6 (exp #4), we train and assess the model with data directly rendered to low-resolution, thereby encoding higher quantization errors. This models shows lower performance, validating the consideration that low-resolution rendering leads to biased coordinate regression. It is worth mentioning that the errors are not extremely higher than ours, assuming that the multi-view supervision can better handle this bias as well as the sub-pixel coordinate regression characteristics of *CoM3D* heatmap coordinate decoding module.

#5. W/o *versus* w/ data augmentation. To demonstrate the contribution of 3D rotational augmentation of our data (Sect. 3.5), we train our model by excluding it (Table 6 (exp #5)). Our sparse point cloud input provides us the privilege to actually rotate it before rendering, tremendously increasing the amount of new depth map inputs during training with significant effect as figured in the results, an important difference in comparison with the limitations of pseudo-rotational augmentation applied on 2D visual data.

#6. W/o *versus* w/ data normalization. In Table 6 (exp #6), we evaluate the contribution of the volumetric scale and translation normalization transform $\mathcal{T}_{\mathcal{N}}$ we perform to the marker point cloud to occupy equal volume in the normalized voxel-grid for all samples. We observe that this normalization boosts the performance of our model across all metrics. Our consideration is that this transform leads to high variance with respect to the human body structures, allowing the model to learn how to directly reconstruct the scale normalized absolute 3D poses.

#7. W/o *versus* w/ heatmap aggregation. In our approach, instead of supervising heatmaps as originally proposed for the architectures we build upon, we supervise only the aggregated heatmaps. This aggregation scheme drives our model to faster convergence and slightly better results, especially for the marker estimation, as shown in Table 6 (exp #7), in comparison with last stage only supervision (HRNetV1), as proposed for HRNET in the respective work (Wang et al. (2020)). Number of sensors. Finally, we conduct experiments of our model on the validation and testing sets taking into account the marker observations only from 3 and 2 sensors (Table 6 (exp #8 and #9)), respectively, instead of the full 4-sensor setup with which we trained DeMoCap. We present this experiment to assess the bias of our model on the training set and the generalization capabilities, as well as its sensitivity to sensor decrease.

#8. 3 *versus* 4 capturing depth sensors. As expected, the accuracy is lower in relation to the 4-sensor captured data, nevertheless, the results are still better than the compared methods, showing lower M_{PJPE} and RMS_{PJPE} errors and higher $\mathrm{mAP}_{50\mathrm{mm}}$ accuracy (46.81 mm, 53.50 mm and 77.17% in the validation and 47.58 mm, 60.02 mm and 81.88 mm in the testing set, respectively).

#9. 2 *versus* 4 capturing depth sensors. On the 2-sensor assessment, the performance is further decreased, however, the results can be considered fair enough given the lack of information. Our conclusion from this experiment is that DeMoCap follows a reasonable dependency on the number of sensors that capture the markers, as the high-end MoCap systems do with their specialized cameras.

5.5 Study on Clean MoCap data

5.5.1 On DeMoCap Dataset

We further benchmark our model by training and assessing it using as input the post-processed, clean marker data from VICON used as ground-truth, under the same learning configurations. These experiments showcase the behaviour of the model in ideal conditions where the marker data are totally clean and highly precise, without the noise present in unclean optical marker data either captured with low-cost depth sensors or high-end MoCap systems before post-processing. In Table 7, we present results of DeMoCap and DeMoCap trained with VICON data ($\mathrm{DeMoCap}_{vicon}$) assessed both on clean VICON and noisy data from consumer-grade depth sensors (RS).

As expected, $\mathrm{DeMoCap}_{vicon}$ achieves significantly high accuracy both on the validation and testing VICON sets, achieving M_{PJPE} and M_{PMPE} lower than 3 cm and $\mathrm{mAP}_{50\mathrm{mm}}$ 99.81% and 94.07% in each set, respectively. On the other hand, $\mathrm{DeMoCap}_{vicon}$ showcases low performance on noisy RS data exceeding 6 cm for absolute distance errors and $\mathrm{mAP}_{50\mathrm{mm}}$ performance lower than 56%, even lower than DeMoCap assessed on 2-viewpoint data only, given the dissimilarity of the evaluation sets in comparison to the training set.

Results of particular interest are presented by our model when assessed on the clean validation and testing set from VICON. DeMoCap demonstrates significantly better performance in VICON than RS data, though trained on the latter, letting us consider that the model generalizes well without bias on the systemic camera noise, poses or pinhole parameters. The model is trained to handle noisy and clean data, showing that increasing the accuracy of the marker capturing leads to more reliable inference.

Table 7 Results on DeMoCap clean MoCap data. We train DeMoCap with clean MoCap data to assess the performance of the models in various combinations between training and validation/testing sets

Method \ Metrics (mm)	Set	M_{PJPE}	RMS_{PJPE}	M_{PMPE}	RMS_{PMPE}	mAP_{50mm}	M_{PJAE}	RMS_{PJAE}
DeMoCap on VICON data	val	29.64	34.58	36.61	42.95	96.11%	16.28	18.60
$DeMoCap_{vicon}$ on RS data		72.76	108.94	94.75	135.48	45.11%	31.42	41.00
$DeMoCap_{vicon}$ on VICON data		**21.04**	**24.24**	**19.09**	**24.49**	**99.81%**	**11.18**	**13.52**
DeMoCap		33.83	42.65	42.33	51.74	90.41%	18.66	22.47
DeMoCap on VICON data	test	37.28	47.63	47.80	59.79	89.86%	14.52	19.40
$DeMoCap_{vicon}$ on RS data		62.81	87.50	90.78	122.57	55.09%	28.34	35.88
$DeMoCap_{vicon}$ on VICON data		**25.16**	**32.98**	**27.44**	**37.30**	**94.07%**	**10.04**	**15.28**
DeMoCap		40.04	51.69	52.92	66.49	88.05%	19.73	26.18

Bold indicates the results of the best performing methods

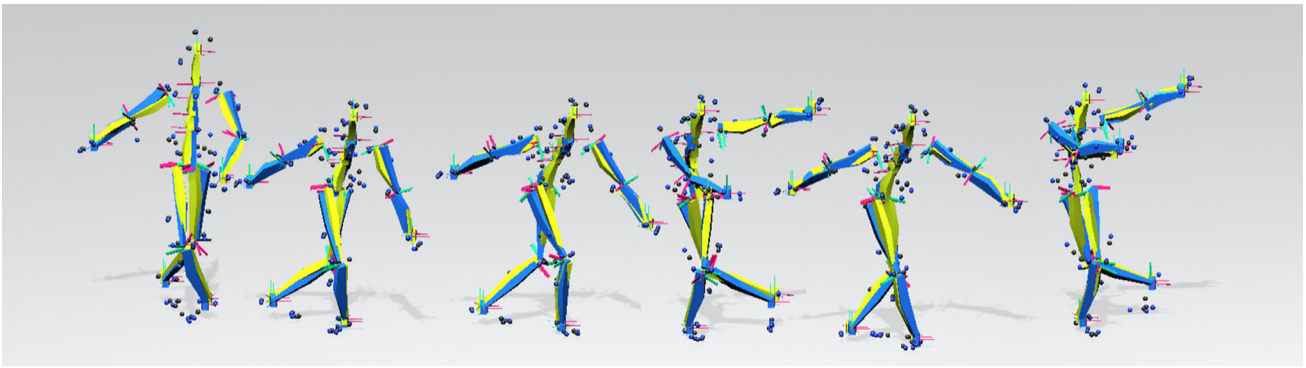

Fig. 11 Qualitative results of various frames illustrated on the same scene (locomotion) from *DanceTurns002* sequence of SFU Dataset (Ying (2011)), on totally unseen body structures and activities. The yellow poses represent the predictions of DeMoCap, while the blue ones the ground-truth data of the dataset

5.5.2 On SFU Dataset

We also evaluate the performance of our models, DeMoCap and $DeMoCap_{vicon}$ on a public MoCap dataset with a relatively similar marker configuration and pose structure with 53 markers and 30 joints, SFU Motion Capture Database by Ying (2011). Indicatively, in our experiments, we include two challenging activities, *DanceTurns002*[7] and *HopOverObstacle001*.[8] The quantitative results for 575 samples in total are shown in Table 8. Visually, the models showcase comparable results, as illustrated in Fig. 11, numerically though, only $DeMoCap_{vicon}$ reaches high scores, while for DeMoCap, the task is proved more challenging. It is worth highlighting the spatial offsets existing between different body structures for the various datasets that insert a constant error in the measurements, as discussed in Sect. 6.

[7] http://mocap.cs.sfu.ca/index154af.html?id=0018_DanceTurns002.bvh

[8] http://mocap.cs.sfu.ca/index1fe61.html?id=0015_HopOverObstacle001.bvh.

Table 8 Results on SFU (Ying (2011)) clean MoCap data. We train DeMoCap with clean MoCap data to assess the performance of the models in various combinations between training and validation/testing sets

Model \ Metrics (mm)	M_{PJPE}	RMS_{PJPE}	mAP_{50mm}
DeMoCap	58.28	69.38	48.95%
$DeMoCap_{vicon}$	**45.28**	**54.95**	**75.46%**

Bold indicates the results of the best performing methods

6 Discussion

In this section, we present a summary of our observations, discussing the pros and cons of the various motion capture solutions, in relation to our approach and beyond.

6.1 Strengths

For decades, marker-based motion capture has been the gold-standard for high-fidelity motion capturing and tracking. Nevertheless, despite its sub-millimeter accuracy on marker tracking, the use of marker-based systems is globally limited given the high costs of the setups, the software licenses, the

maintenance and more. To the best of our knowledge, DeMoCap is the first computer vision method that enables the use of low-cost equipment for marker-based motion capture with comparable results to high-end MoCap systems, to the extent the hardware and data limitations allow for.

DeMoCap is one of the pioneering methods in deep marker-based motion capture that allows one-shot regression of pose from a sparse set of 3D points. The method performs better than recent state-of-the-art color-based methods (i.e. LT and 4DA) despite the use of highly erroneous depth estimates from low-cost sensors (depth error higher than 3 cm in 1.5 m distance from the camera), while the mean average per joint error drops under 2.5 cm when trained and assessed on clean data. DeMoCap generalizes well even with the use of low number of cameras (2 or 3 sensors, Sect. 5.4), showcasing increased stability in comparison with methods based on potentially erroneous partial view detections (e.g. 2D pose detectors). The model is driven to reject outliers and detect missing markers at the first stage (marker inference), allowing for pose estimation from refined prior marker information.

DeMoCap focuses exclusively on the information that solves the pose, i.e. the markers attached on the body, without interference from background context as color or dense depth data do. Finally, DeMoCap performs better when noise is reduced (as assessed in Sect. 5.5), despite the existence of systematic noise of the depth sensors in the training set, showcasing that our model, when the marker configuration is the same, is affected mostly by the quality of the marker tracking, as all marker-based motion capture system do.

6.2 Weaknesses

Nonetheless, DeMoCap still presents weaknesses in comparison with traditional high-end marker-based systems and markerless methods based on dense visual data. The current consumer-grade sensors used to trade the high costs of the specialized MoCap cameras are limited with regards to the capturing frequency (30 Hz vs 120/240 Hz or higher) and depth-sensing range (up to 4 m keeping acceptable accuracy). To this end, contrary to professional marker-based solutions or methods applied on dense visual data applicable in large scale capturing areas, e.g. arenas, sport fields or stadiums, DeMoCap is particularly limited with regards to the capturing space volume, at least based on the existing consumer-grade depth and infrared sensing technologies.

Furthermore, similarly to all data-driven statistical models, DeMoCap is trained on a special dataset captured with a specific 53-marker configuration placement for human motion capture. That fact will lead DeMoCap in erroneous predictions in the appearance of different marker configurations or skeleton structures, requiring re-training on data captured with the settings. Contrarily, traditional MoCap can be applied on a variety of moving entities, from humans to animals and objects, where data-driven models lag behind with regards to this flexibility of the gold-standard marker-based MoCap solutions. Traditional MoCap also requires new configuration for marker labeling and skeleton tracking, however, it is "cheaper" due to the shorter time and less effort needed for its completion, without the need for dataset creation. In other words, for all marker-based solutions, the placement of markers is a strong prior for their operation, however, this prior is even stronger for DeMoCap due to its data-driven modeling.

DeMoCap provides one-shot inference for markers and pose 3D regression avoiding the possibility for marker swapping, a common case in MoCap tracking when markers are getting very close to each other. The temporal aspect though can extremely eliminate potential errors, being a strong driver for correct predictions. DeMoCap's inference is instantaneous, without considering the temporal aspect of the marker trajectories. This constitutes a limitation for DeMoCap in comparison with marker-based solutions designed for out-of-the-box marker tracking of high stability and precision.

7 Conclusions

In this paper, we introduced DeMoCap, a low-cost lightweight data-driven model for marker-based motion capture with the use of spatio-temporally aligned infrared- and depth-sensing streams acquired with consumer-grade devices. We train our model on noisy optical marker data captured with a low-cost multi-view system to accurately regress marker and joint 3D coordinates by staging a smooth representation transition from markers to 3D pose to learn the underlying structural relation between human body and marker configuration placement in an end-to-end, scale- and translation-invariant manner. Learning upon it, the model overcomes bias on our relatively limited training data and generalizes well. Technically, our method is the first that introduces the use of fully convolutional networks to be applied on extremely sparse depth maps efficiently regressing marker and joint 3D coordinates by posing their estimation as a joint 2D localization and regression objective within a normalized 3D space to embed the z-dimension indirectly with the introduction of a new fully differentiable module for 3D regression. The special dataset we created to drive our model is publicly available, containing inter- and intra-system spatio-temporally aligned infrared-depth and motion capture data.

We focus our future work to overcome the aforementioned limitations of our method. Our approach is limited to regress markers and pose of one single person in the capturing space. That is due to use of spatial regression that limit us to regress one single coordinate per latent heatmap layer. We aim to conduct research on that challenging task to

enable multi-person motion capture. Even though we apply our method on temporally continuous 3D data, DeMoCap is not designed to maintain internal memory for sequential data processing, instead, the inference is single-shot without considering the previous predictions. On the one hand, the discrete per frame inference allows us to skip issues that tracking techniques can cause such as marker swaping, however, considering temporal information can lead to higher motion capture accuracy and robustness. Hence, we will explore potential techniques that will allow us to introduce temporal features in our work for the development of more efficient and effective deep learning models for motion capture. Finally, given the regressed labeled marker data, soft inverse kinematics solvers can be explored to result in joint

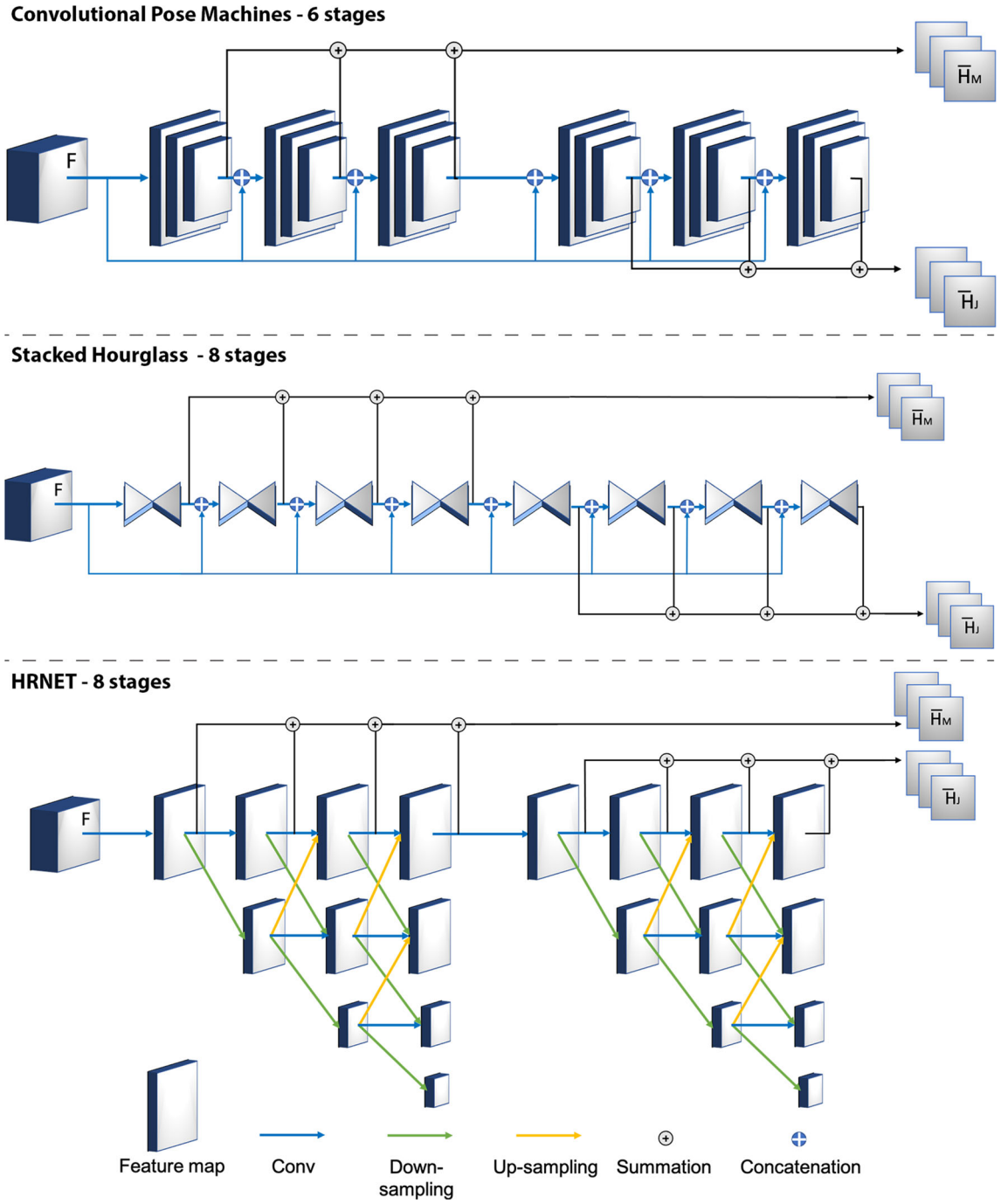

Fig. 12 Markers-to-pose multi-stage FCN Architectures. We illustate in high level the architectures we used to train DeMoCap. In all of them, we follow the same concept where the first stages predict $\mathbf{H}_{M,1...K}$, aggregated to $\bar{\mathbf{H}}_M$, while the last stages predict $\mathbf{H}_{J,K+1,...,2K}$ aggregated to $\bar{\mathbf{H}}_J$. The predictions of each stage and the feature maps $\mathbf{F}$ are concatenated for each subsequent stage

transformation solving similar to professional motion capture solutions that will allow the regression of bone orientation in a data-driven end-to-end manner.

Appendix A Network Architecture Details

For all architectures, we follow the same schema. All our networks consist of 2 super-stages, meaning groups of stages as defined in multi-stage and multi-branch FCN architectures. The first super-stage predicts $\mathbf{H}_{M,1,\ldots,K}$ heatmaps which are aggregated to the final marker heatmaps $\bar{\mathbf{H}}_M$, while the second super-stage predicts heatmaps $\mathbf{H}_{J,K+1,\ldots,2K}$ which are aggregated to the final joint heatmaps $\bar{\mathbf{H}}_J$. The predictions of each stage and their corresponding image features are concatenated for each subsequent stage. High level designs of the various architectures are illustrated in Fig. 12. We discuss each network details below.

A.1 Convolutional Pose Machines (CPM)

Following the original work by Wei et al. (2016), we stack 6 stages in total, separated in the two super-stages of 3 stages each. However, we reduce the number of *MaxPooling* layers to 2 instead of 3 by removing the third one. This results into a higher resolution feature map $\mathbf{F}$, (i.e. of 2D spatial size 40×40) leading to an increased heatmap resolution, similar to the rest of the networks.

Subsequently, we follow the stage architecture as originally proposed in the CPM network, i.e. the first stage consists of one 9×9 followed by two 1×1 convolutional layers, whilst every next stage is composed of 5 convolutional layers ($3 \cdot 11 \times 11 - 2 \cdot 1 \times 1$). All these stages are fed with the concatenation of $\mathbf{F}$ and the output of the previous stage, except for *Stage1* which is only fed with $\mathbf{F}$ alone.

A.2 Stacked Hourglass (SH)

We build a 8-stage Stacked Hourglass (Newell et al. (2016)) based on the configuration of the original work, selecting 4 stages per super-stage. Starting from a pre-processing module, a feature map $\mathbf{F}$ is extracted which, similarly for all networks, we concatenate with the intermediate feature map outputs of each stage. We use hourglass modules with depth equal to 2, reaching to heatmaps of 2D spatial size equal to 40×40.

A.3 High Resolution Network (HRNET)

Following the configuration of the original work (Wang et al. (2020)), we build a HRNET-based network by staging two 4-stage HRNET architectures, one per super-stage. Due to the multi-stage and multi-branch design of HRNET, we select to build the second super-stage as a 4-stage HRNET instead of a 8-stage and 8-branch model. Similarly, we feed the second super-stage with the feature maps $\mathbf{F}$ concatenated with the heatmap outputs. The configuration of each super-stage module is similar to the one proposed in the original work. The initial stage contains 4 residual units formed by a bottleneck with width equal to 64, followed by one 3×3 convolution reducing the width of feature maps to 40×40. We follow the same schema with the original work where the second, the third and four stages of each super-stage consist of 1, 4, 3 exchange blocks, correspondingly.

References

Alexanderson, S., O'Sullivan, C., & Beskow, J. (2017). Real-time labeling of non-rigid motion capture marker sets. *Computers & Graphics, 69*, 59–67.

Bascones, J. L. J. (2019). Cloud point labelling in optical motion capture systems. Ph.D. thesis, Universidad del País Vasco-Euskal Herriko Unibertsitatea.

Bekhtaoui, W., Sa, R., Teixeira, B., Singh, V., Kirchberg, K., Yj, Chang, & Kapoor, A. (2020). *View invariant human body detection and pose estimation from multiple depth sensors*. arXiv preprint arXiv:2005.04258.

Buhrmester, V., Münch, D., Bulatov, D., & Arens, M. (2019). Evaluating the impact of color information in deep neural networks. In *Iberian conference on pattern recognition and image analysis* (pp. 302–316). Springer.

Burenius, M., Sullivan, J., & Carlsson, S. (2013). 3D pictorial structures for multiple view articulated pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3618–3625).

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291–7299).

Chatzitofis, A., Zarpalas, D., Kollias, S., & Daras, P. (2019). Deepmocap: Deep optical motion capture using multiple depth sensors and retro-reflectors. *Sensors, 19*(2), 282.

Chatzitofis, A., Saroglou, L., Boutis, P., Drakoulis, P., Zioulis, N., Subramanyam, S., Kevelham, B., Charbonnier, C., Cesar, P., Zarpalas, D., et al. (2020). Human4d: A human-centric multimodal dataset for motions and immersive media. *IEEE Access, 8*, 176241–176262.

Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., Zhang, L. (2019). Bottom-up higher-resolution networks for multi-person pose estimation. arXiv preprint arXiv:1908.10357.

Doosti, B., Naha, S., Mirbagheri, M., & Crandall, D. J. (2020). Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6608–6617).

Elhayek, A., de Aguiar, E., Jain, A., Tompson, J., Pishchulin, L., Andriluka, M., Bregler, C., Schiele, B., & Theobalt, C. (2015). Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3810–3818).

Feng, Z. H., Kittler, J., Awais, M., Huber, P., & Wu, X. J. (2018). Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2235–2245).

Fuglede, B., Topsoe, F. (2004). Jensen-shannon divergence and hilbert space embedding. In *International symposium on information theory, 2004. ISIT 2004. Proceedings* (p. 31). IEEE.

Gao, H., & Ji, S. (2019). Graph u-nets. In *International conference on machine learning, PMLR* (pp. 2083–2092).

Gaschler, A. (2011). Real-time marker-based motion tracking: Application to kinematic model estimation of a humanoid robot. Thesis

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th international conference on artificial intelligence and statistics* (pp. 249–256).

Guler, R. A., & Kokkinos, I. (2019). Holopose: Holistic 3D human reconstruction in-the-wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 10884–10894).

Han, S., Liu, B., Wang, R., Ye, Y., Twigg, C. D., & Kin, K. (2018). Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics (TOG), 37*(4), 166.

Haque, A., Peng, B., Luo, Z., Alahi, A., Yeung, S., & Fei-Fei, L. (2016). Towards viewpoint invariant 3D human pose estimation. In *European conference on computer vision* (pp. 160–177). Springer

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

Holden, D. (2018). Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG), 37*(4), 1–12.

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2013). Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(7), 1325–1339.

Iskakov, K., Burkov, E., Lempitsky, V., & Malkov, Y. (2019). Learnable triangulation of human pose. In *Proceedings of the IEEE international conference on computer vision* (pp. 7718–7727).

Joo, H., Simon, T., & Sheikh, Y. (2018). Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8320–8329).

Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., & Bhowmik, A. (2017). Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1–10).

Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Li, S., Zhang, W., & Chan, A. B. (2015). Maximum-margin structured learning with deep networks for 3D human pose estimation. In *Proceedings of the IEEE international conference on computer vision (ICCV)*.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755). Springer.

Loper, M., Mahmood, N., & Black, M. J. (2014). Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG), 33*(6), 220.

Luvizon, D. C., Tabia, H., & Picard, D. (2019). Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics, 85*, 15–22.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., Black, M. J. (2019). Amass: Archive of motion capture as surface shapes. arXiv preprint arXiv:1904.03278.

Martínez-González, A., Villamizar, M., Canévet, O., Odobez, J. M. (2018a). Investigating depth domain adaptation for efficient human pose estimation. In *2018 European conference on computer vision—workshops, ECCV 2018*.

Martínez-González, A., Villamizar, M., Canévet, O., & Odobez, J. M. (2018b). Real-time convolutional networks for depth-based human pose estimation. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 41–47). https://doi.org/10.1109/IROS.2018.8593383.

Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H. P., Xu, W., Casas, D., & Theobalt, C. (2017). Vnect: Real-time 3D human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG), 36*(4), 1–14.

Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Elgharib, M., Fua, P., Seidel, H. P., Rhodin, H., Pons-Moll, G., Theobalt, C. (2019). Xnect: Real-time multi-person 3D human pose estimation with a single RGB camera. arXiv preprint arXiv:1907.00837.

moai, . (2021). *moai*: Accelerating modern data-driven workflows. https://github.com/ai-in-motion/moai.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European conference on computer vision* (pp. 483–499). Springer.

Nibali, A., He, Z., Morgan, S., Prendergast, L. (2018). Numerical coordinate regression with convolutional neural networks. arXiv preprint arXiv:1801.07372.

Park, S., Yong Chang, J., Jeong, H., Lee, J. H., & Park, J. Y. (2017). Accurate and efficient 3D human pose estimation algorithm using single depth images for pose analysis in golf. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 49–57).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. de-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates Inc.

Pavllo, D., Porssut, T., Herbelin, B., & Boulic, R. (2018). Real-time finger tracking using active motion capture: A neural network approach robust to occlusions. In *Proceedings of the 11th annual international conference on motion, interaction, and games* (pp. 1–10).

Perepichka, M., Holden, D., Mudur, S. P., & Popa, T. (2019). Robust marker trajectory repair for mocap using kinematic reference. In *Motion, interaction and games* (pp. 1–10). Ernst & Sohn.

Qi, C. R., Yi, L., Su, H., Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems* (pp. 5099–5108).

Qiu, H., Wang, C., Wang, J., Wang, N., & Zeng, W. (2019). Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision* (pp. 4342–4351).

Rhodin, H., Salzmann, M., & Fua, P. (2018). Unsupervised geometry-aware representation for 3D human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 750–767).

Riegler, G., Osman Ulusoy, A., & Geiger, A. (2017). Octnet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3577–3586).

Rüegg, N., Lassner, C., Black, M. J., Schindler, K. (2020). Chained representation cycling: Learning to estimate 3D human pose and shape by cycling between representations. arXiv preprint arXiv:2001.01613.

Sigal, L., Isard, M., Haussecker, H., & Black, M. J. (2012). Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision, 98*(1), 15–48.

Sterzentsenko, V., Karakottas, A., Papachristou, A., Zioulis, N., Doumanoglou, A., Zarpalas, D., & Daras, P. (2018). A low-cost, flexible and portable volumetric capturing system. In *2018 14th international conference on signal-image technology & internet-based systems (SITIS)* (pp. 200–207). IEEE.

Sun, X., Xiao, B., Wei, F., Liang, S., & Wei, Y. (2018). Integral human pose regression. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 529–545).

Tensmeyer, C., Martinez, T. (2019). Robust keypoint detection. In *2019 international conference on document analysis and recognition workshops (ICDARW)* (Vol. 5, pp. 1–7). IEEE.

Tompson, J.J., Jain, A., LeCun, Y., Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems* (pp. 1799–1807).

Toshev, A., & Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1653–1660).

Tu, H., Wang, C., Zeng, W. (2020). Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *Computer Vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16* (pp. 197–212). Springer.

VICON L. (1984). Vicon systems ltd. https://www.vicon.com/

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. (2020). Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wei, S. E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4724–4732).

Yang, Y., Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011* (pp. 1385–1392). IEEE.

Ying, K. Y. G. J. (2011). Sfu motion capture database. http://mocap.cs.sfu.ca/

Zanfir, A., Marinoiu, E., Zanfir, M., Popa, A. I., & Sminchisescu, C. (2018). Deep network for the integrated 3D sensing of multiple people in natural images. *Advances in Neural Information Processing Systems, 31*, 8410–8419.

Zhang, F., Zhu, X., Dai, H., Ye, M., & Zhu, C. (2020a). Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7093–7102).

Zhang, Y., An, L., Yu, T., Li, X., Li, K., & Liu, Y. (2020b). 4D association graph for realtime multi-person motion capture using multiple video cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1324–1333).

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE Multimedia, 19*(2), 4–10.