CVM-Net: Motion Reconstruction from a Single RGB Camera with a Fully Supervised DCNN

Mansour Tchenegnon, Thibaut Le Naour, Sylvie Gibet

Journées Françaises de l'Informatique Graphique 2021 / R.Vergne and N.Mellado.
(Paper Chairs)

CVM-Net: Motion Reconstruction from a Single RGB Camera with a Fully Supervised DCNN

Mansour Tchenegnon[†1] and Thibaut Le Naour[‡1] and Sylvie Gibet[§1]

[1]IRISA, Université Bretagne Sud

Abstract

Many solutions have been proposed for 3D human pose estimation from video data. However, only a few of them take into account temporal features. In this article, we present a method focusing on this temporal aspect and show promising results. Our approach consists of two parts. The first one concerns the creation of a dataset that contains a variety of motion features. Based on this dataset, the second one deals with the training of a DCNN-based model, which takes as input the 2D pose estimations directly computed from videos. Here we present the first training tasks and results obtained using our deep neural network model to directly estimate 3D poses. Three models were trained using the same architecture applied on several configurations of our dataset. Using a small benchmark, we evaluate our network architecture.

CCS Concepts

•***Computing methodologies*** → *Motion processing; Neural Networks;*

1. Introduction

Human pose estimation is a very competitive research subject in Computer Vision and Computer Graphics. The greatest progress has been made in 2D pose estimation [CHS*18, SXLW19]. Recently, researchers have focused on estimating human poses in 3D, which is a major challenge. Achieving high accuracy in this task may open new perspectives in different application fields, such as computer vision, computer animation, robotics, etc.With the new neural network techniques, many solutions are proposed.

To estimate 3D human poses, there are mainly two approaches. One way to solve this task is to use the best solution in 2D pose estimation [CHS*18, SXLW19]. The estimated 2D pose can then be lifted with a specific algorithm, using for example a regression-based neural network [CR17, BSGB19]. Another way is to first detect features for the 3D pose, such as heat maps, depth, camera focal length, and then compute the 3D pose [LPT18, HXM*19, MCL19]. The challenge becomes more difficult depending on the type of data considered (images or videos), as videos require, compared to images, to take into account the temporal context. First work that deals with temporal features uses as input a set of frames [MSS*17, CGL*19, WYXL20]. Others propose to use kinematic methods after 3D estimation to improve the results [SUF*20]. All this was made possible thanks to the large-scale datasets available for this task: Human3.6M [IPOS14], MPI-INF-3DHP [MRC*17], HumanEva [SBB10].

In this paper, we choose to address the issue of reconstructing 3D motion from videos, taking into account temporal features. We adopt a two-step approach, which consists of:

- **Building a dataset**. The goal of this part is to build a dataset with a variety of activities where we have control over its content. While existing data sets contain mainly images, positions and angles data, it is almost impossible to extract new features from them, we believe that building our own data set will allow us to have full control over it and be able to add, compute and test as many features as necessary.
- **Training and evaluating a model architecture**. This part is the main objective of the study. We propose a fully supervised DCNN solution using temporal information to reconstruct smooth and plausible motions from videos.

The paper is organized as follows. A brief state of the art is given in Section 2. In Section 3, we describe the details of our approach. The results of our experiments in training and evaluation are then presented in Section 4, before concluding.

2. Related works

Recent advances in deep neural network techniques brought a significant improvement in human pose estimation, both in 2D or 3D. The greatest improvement was in the 2D human pose estimation task. Features such as heat maps can be computed using multi-layer CNNs (Convolutional Neural Networks), and 2D poses can then

† PhD Student mansour.tchenegnon@univ-ubs.fr
‡ Co-supervisor
§ Supervisor

be estimated with a better accuracy [CHS*18, SXLW19, GKSB19, FXT*17]. Recent work focuses on 3D human pose estimation and motion reconstruction tasks that we describe below.

2.1. 3D human pose estimation

We can group 3D human pose estimators into two main categories according to the type of data used as input.

The first category uses images as input and directly estimates 3D poses. Methods in this category compute 2D and 3D features, such as heat maps and other features (camera focal length, depth information) to estimate the final 3D poses. These methods generally involve two stages: a features detection stage followed by a 3D pose estimation [MN17, LPT18, HXM*19, MCL19]. Yang et al. [YOW*18] use a 3D estimation pose network and a pose discriminator to ensure that the estimated poses are plausible. Wei et al. [WWTL21] use a framework to generate heat maps and bone maps in order to extract 2D pose hypotheses. A pose regressor or a selection-based algorithm uses these hypotheses to compute the final 3D pose.

The second category starts from the estimated 2D poses and, through various methods, estimates the corresponding 3D poses. The main advantage of this approach is that it is more efficient on videos in the wild, due to the use of state-of-the-art 2D estimators. Some researchers propose lifting models [CR17, BSGB19]. Chen et al. [CTA*19] present an unsupervised algorithm that lifts 2D joints to 3D skeletons. They show that adding random 2D projections and an adversarial network, allow the training process to be self supervised using geometric consistency. Martinez et al. [MHRL17] propose an approach of consecutive linear layers to perform a 2D-to-3D keypoints regression.

The main problem with these models is that, while working efficiently with static data such as a single image or pose, they fail in processing dynamic data (videos or sequences of poses), where we can observe some jittering.

2.2. Motion reconstruction

Many 3D pose estimators are currently proposed in the literature, most of them work on a single image at the time. When receiving a video as input, they estimate the pose at each frame, and then directly concatenate the outputs. This way of reconstructing a motion does not take into account the temporal characteristics of motion. This leads to a lot of jittering in the results, and very few approaches have considered these effects [MSS*17, YAC19, PFGA19, CGL*19, SUF*20, LLD*21]. Among them, Metha et al. [MSS*17] choose to infer the pose at time $t-1$ to estimate the pose at time t. Another solution was proposed by Yiannakides et al. [YAC19]. They defined a database where the 2D joint positions obtained by projections are associated with the 3D skeleton poses extracted from CMU Mocap Database [Car]. Then they can extract the 2D poses from a video and determine the closest 2D projections from the database. The associated 3D poses are then selected taking into account the temporal consistency. Wang et al. [WYXL20] represent the 2D skeleton input as a spatio-temporal graph and propose a loss function (motion loss) and a Graph Convolution Networks to predict 3D poses. Xu et al. [XYN*20] choose to estimate 3D poses first and then use a trajectory completion framework to correct the sequence. More recently, Shi et al. proposed MotioNet [SUF*20]. Their solution is a forward-kinematic-based DCNN. Two independent CNN models are first in charge of estimating the sequence of rotations and the bones length. From these features, they apply the forward kinematics to obtain the sequence of 3D poses.

3. Our approach

Our approach is based on two main stages. First, from a Mocap dataset, we generate a new dataset with features specific to motion. Then, in a second stage, we create a deep neural network model and train it with the previous dataset.

3.1. Dataset

The first step in our approach consists in creating a suitable dataset for the motion reconstruction task. In order to create our own dataset, we start by choosing a Mocap Database among the available ones, such as CMU [Car] and HDM05 [MRC*07]. Each database has its own set of movements, ranging from basic activities to sport activities. As we chose to work with short length activities, we therefore selected HDM05 [MRC*07], which gives access to cuts of scenarios grouped in different motion classes.

HDM05 contains motion sequences performed by five non-professional actors. Each actor repeated the sequences several times according to the scenarios they were instructed. The motion clips were obtained by segmenting these sequences. There are about 1500 clips in the database with approximately 130 motion classes. We selected 309 motion clips, and for each motion category, we selected one clip per actor if available. In order to create the data for our training task, we used Unity3D engine to create a 3D scene with 5 avatars. We then generated the animations corresponding to each motion clip and avatar. From these animations, we retrieved and stored data for each sequence and each pose. Then, for each pose and articulation, we extracted the 2D and 3D positions, as well as the relative Euler angles and quaternions in the Unity camera space (See Figure 1).

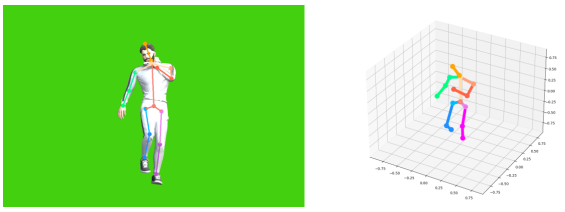

Figure 1: *Sample from our dataset. On the left is the image and the 2D pose drawn on top. On the right is the corresponding 3D pose.*

Initially, HDM05 motions have been captured with a frame rate of 120 fps. But while converting them to Unity, this frame rate was reduced to 30 fps, resulting in sequences lengths between 40 and 300 frames.

3.2. Neural network architecture

The second stage of our approach is to define a deep neural network model architecture in order to perform the reconstruction task. We begin with building a simple architecture to estimate a sequence of 3D poses from 2D poses.

3.2.1. Architecture

We consider a pose as a 1D-vector of skeleton joints positions. Let $P_{2D} \in \mathbb{R}^{Tx2J}$ and $P_{3D} \in \mathbb{R}^{Tx3J}$ represent two temporal sequences of respectively 2D and 3D poses from the same video. T is the temporal length of each sequence and $T \in [40, 300]$, J represents the number of joints of a skeleton.

Our 2D-to-3D poses model is based on 1D-convolution. As stated previously, the goal is to produce a sequence of 3D poses from a sequence of 2D poses. Our model is composed of two almost identical consecutive CNN blocks. The first block takes as input the sequence of 2D poses and outputs a sequence of 3D poses, whereas the second block takes as input the 3D poses and works as a refiner. The global architecture of the model and the implementation details of each blocks are shown in Figure 2.

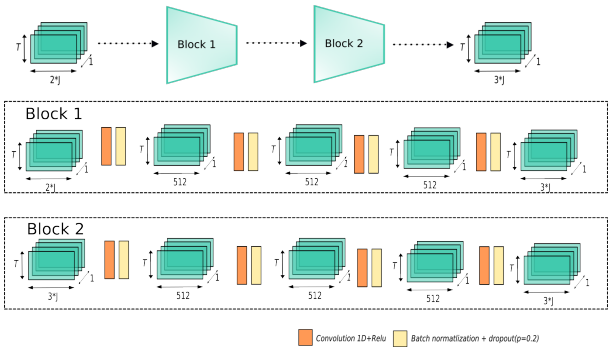

Figure 2: *Architecture of the model. On top we have the global structure of the model. In the middle are the implementation details of the first block, and at the bottom those of the second block. We can see that the main difference resides in the input of each block.*

3.2.2. Loss function

The model aims to predict the sequence of poses $P_{3D} \in \mathbb{R}^{Tx3J}$. For this particular task, we choose as our loss function the most widely used function in 3D pose estimation, namely the Mean Per Joint Position Error (MPJPE), which is the average of the Euclidean distance between the estimated and the ground truth positions on the joints:

$$MPJPE(\overline{p}_t, p_t) = \frac{1}{J}\sum_{j=1}^{J} ||\overline{p}_t^j - p_t^j||$$

where J represents the number of joints, $\overline{p}_t$ and p_t represent respectively the vectors of estimated joints positions and ground truth joints positions at frame t in the sequence.

The loss function we used is the mean of MPJPE over the whole sequence.

$$\ell = \frac{1}{T}\sum_{t=1}^{T} MPJPE(\overline{p}_t, p_t)$$

where T represents the total length of the sequence of poses.

4. Results

In order to evaluate our model and to check its robustness, we have implemented two types of experiments: 1) First, we train and test our model on different data sets extracted from HDM05; 2) Then we evaluate the ability of our model trained on HDM05 to reconstruct the motion poses extracted from the CMU database.

4.1. Training experiments

We set up various training/set sessions with different data sets for different tasks of increasing complexity. Each session trains a model over 50 epochs and at each epochs we compute loss value. For each configuration, we observe the loss history (MPJPE in meters) over the epochs. For each experiment, the train set contains approximatively 1200 samples and the test set (or validation set) around 300.

4.1.1. First task – Cross validation

In the first experimentation, which is a standard case of train-test task usually performed for cross-validation, the original dataset is randomly split into train (80%) and test data (20%). This is our first model (model A). We can observe in Figure 3 how the loss function is minimized over the epochs.

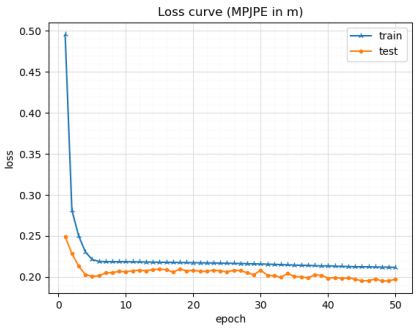

Figure 3: *MPJPE over training for model A. The original dataset has been randomly split into 80% training set and 20% test set.*

4.1.2. Second task – One actor data out as test set

This experiment focuses on the variability of the motion performed by different actors, due to different factors (morphological, gender, style variations, etc.). In order to test the robustness of our model to these variations, we split the dataset so that the motion performed by one of the five actors is left as the test set and the rest as the training set (model B). The MPJPE loss minimized during the training can be observed in Figure 4.

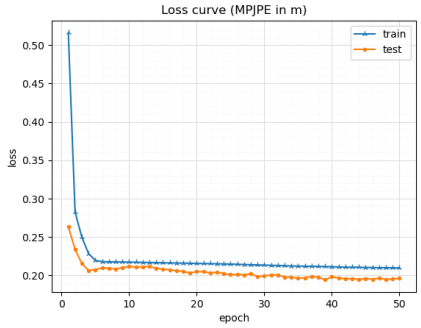

Figure 4: *MPJPE over training for model B. The motion performed by one of the actor in HDM05 dataset has been isolated as validation set.*

4.1.3. Third task – One HDM05 motion class out as test set

In this session, we isolate one class of motion as test set and train the model on all other classes of motion, in order to verify that our model can reconstruct an unknown motion from its training (model C). The MPJPE loss minimized during the training can be observed in Figure 5).

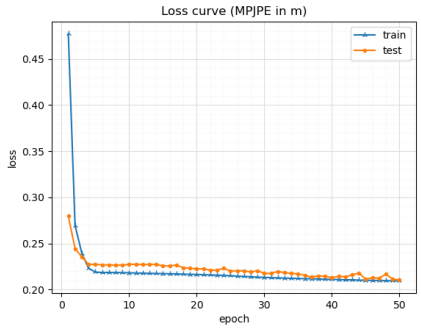

Figure 5: *MPJPE over training for model C. The test set contains only Hop (jump) (one leg and both) activities of HDM05 dataset.*

4.1.4. Observations and interpretations

The previous experiments results show that our model architecture is able to learn with an average error of about 200mm. These results show that our neural network architecture cannot achieve better results while trained on our dataset. The first training session is a standard cross-validation task. In the second session, we observe that the error is lesser on the test set than on the train set. However, in the third training task, whereas the test set contains some motion classes not represented in the training set, the final error is larger on the test set than on the training set. In order to show the efficiency of our DCNN architecture and to go further on its generalization capability, we train a new model on the whole HDM05 dataset and evaluate it on some samples extracted from the CMU dataset.

4.2. Evaluation on CMU samples

In this experiment we prepare a small benchmark based on CMU Mocap Dataset [Car]. We use the same process described in section 3.1 to generate this benchmark. Two activities are considered, basketball and break dance, with 30 clips perform by one avatar in Unity3D Game Engine. To compare the efficiency of the model we evaluate the MPJPE as well as the PCK (Percentage of Correct Keypoints) [DMSM20, MRC*17, TS14] with a threshold of 200 mm. Table 1 presents the PCK and the MPJPE achieved on dance and basketball activities from our samples. We obtain better results for basketball, probably because there are a few basketball throw clips in our dataset and no break dance clips.

	Basketball	Break dance	Average
MPJPE	206.6	264.6	235.6
3DPCK	54.59	45.54	50.07

Table 1: *MPJPE and the PCK on the 3D poses sequences grouped by motion category.*

We also gathered the different joints and computed the MPJPE values for each group. The group **Hip** is composed of the three joints (root, left hip and right hip), the group **Head** of the two neck joints, and the top of the head. The MPJPE statistics observed in the Table 2 shows that the current architecture estimates the best the hips positions (less than 150 mm error). This is more difficult for extreme points such as head, wrists and ankles. The further a joint is from the root in the skeletal hierarchy, the lower the precision of the estimator.

Hip	Shoulder	Head	Elbow	Ankle	Knee	Wrist
75.59	238.18	216.50	275.03	295.42	259.72	378.01

Table 2: *MPJPE for different joints groups.*

5. Conclusion and perspectives

In this paper, we proposed an approach that combines a specific dataset generated from motion capture and a DCNN-based model trained on this dataset. Our main contributions have been to create an original database, and to show that our DCNN model provides promising results. It should be noted that the architecture of our CVM-Net model is simple. The objective was not so much to compare our model to state-of-the-art methods or to use benchmarks, but to take into account the temporal characteristics by 1D convolution. This is therefore primarily a test phase of our dataset and a first step towards our future goals. From that point we can next propose a spatio-temporal DCNN-based method for 3D poses estimation and motion reconstruction. We will focus on a new approach by adding new information (orientations, trajectories) in the training sessions. We will have to adapt the neural network architecture to the new motion descriptors. We will also work on fixed-length data sequences instead of full data sequences of various sizes. We also plan to use the Laplacian representation to optimize local distortions on full sequences and therefore correct the movement.

References

[BSGB19] BISWAS S., SINHA S., GUPTA K., BHOWMICK B.: Lifting 2d Human Pose to 3d : A Weakly Supervised Approach. URL: http://arxiv.org/abs/1905.01047, arXiv:1905.01047. 1, 2

[Car] CARNEGIE MELLON UNIVERSITY: CMU Graphics Lab Motion Capture Database. URL: http://mocap.cs.cmu.edu/. 2, 4

[CGL*19] CAI Y., GE L., LIU J., CAI J., CHAM T. J., YUAN J., THALMANN N. M.: Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (oct 2019), vol. 2019-Octob, Institute of Electrical and Electronics Engineers Inc., pp. 2272–2281. doi:10.1109/ICCV.2019.00236. 1, 2

[CHS*18] CAO Z., HIDALGO G., SIMON T., WEI S.-E., SHEIKH Y.: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 2017-Janua* (dec 2018), 1302–1310. URL: http://arxiv.org/abs/1812.08008, arXiv:1812.08008. 1, 2

[CR17] CHEN C.-H., RAMANAN D.: 3D Human Pose Estimation = 2D Pose Estimation + Matching. *Proceedings ofthe IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)* (2017). arXiv:1612.06524v2. 1, 2

[CTA*19] CHEN C.-H., TYAGI A., AGRAWAL A., DROVER D., MV R., STOJANOV S., REHG J. M.: Unsupervised 3D Pose Estimation with Geometric Self-Supervision. *Computer Vision and Pattern Recognition (CVPR '19)* (apr 2019). URL: http://arxiv.org/abs/1904.04812, arXiv:1904.04812. 2

[DMSM20] DESMARAIS Y., MOTTET D., SLANGEN P., MONTESINOS P.: A review of 3D human pose estimation algorithms for markerless motion capture. 1–49. URL: http://arxiv.org/abs/2010.06449, arXiv:2010.06449. 4

[FXT*17] FANG H.-S., XIE S., TAI Y.-W., LU C., JIAO TONG UNIVERSITY S., YOUTU T.: RMPE: Regional Multi-Person Pose Estimation. *IEEE International Conference on Computer Vision (ICCV) 1* (2017), 2353–2362. URL: https://cvsjtu.wordpress.com/rmpe-regional-multi-person-pose-estimation/, arXiv:1612.00137v5. 2

[GKSB19] GOLDA T., KALB T., SCHUMANN A., BEYERER J.: Human Pose Estimation for Real-World Crowded Scenarios. *16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (jul 2019). URL: http://arxiv.org/abs/1907.06922, arXiv:1907.06922. 2

[HXM*19] HABIBIE I., XU W., MEHTA D., PONS-MOLL G., THEOBALT C.: In the Wild Human Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations. *Computer Vision and Pattern Recognition (CVPR '19)* (apr 2019). URL: http://arxiv.org/abs/1904.03289, arXiv:1904.03289. 1, 2

[IPOS14] IONESCU C., PAPAVA D., OLARU V., SMINCHISESCU C.: 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 (2014), 1325–1339. doi:10.1109/TPAMI.2013.248. 1

[LLD*21] LI W., LIU H., DING R., LIU M., WANG P., YANG W.: Exploiting Temporal Contexts with Strided Transformer for 3D Human Pose Estimation. 1–13. URL: http://arxiv.org/abs/2103.14304, arXiv:2103.14304. 2

[LPT18] LUVIZON D. C., PICARD D., TABIA H.: 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), 5137–5146. arXiv:1802.09232, doi:10.1109/CVPR.2018.00539. 1, 2

[MCL19] MOON G., CHANG J. Y., LEE K. M.: Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image. *International Conference on Computer Vision (ICCV)* (2019). URL: https://github.com/mks0601/3DMPPE_POSENET_, arXiv:1907.11346v2. 1, 2

[MHRL17] MARTINEZ J., HOSSAIN R., ROMERO J., LITTLE J. J.: A Simple Yet Effective Baseline for 3d Human Pose Estimation. *Proceedings of the IEEE International Conference on Computer Vision 2017-Octob* (2017), 2659–2668. arXiv:1705.03098, doi:10.1109/ICCV.2017.288. 2

[MN17] MORENO-NOGUER F.: *3D Human Pose Estimation from a Single Image via Distance Matrix Regression*. Tech. rep., 2017. arXiv:1611.09010v1. 2

[MRC*07] MÜLLER M., RÖDER T., CLAUSEN M., EBERHARDT B., KRÜGER B., WEBER A.: *Documentation Mocap Database HDM05*. Tech. Rep. CG-2007-2, Universität Bonn, jun 2007. 2

[MRC*17] MEHTA D., RHODIN H., CASAS D., FUA P., SOTNYCHENKO O., XU W., THEOBALT C.: Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. *International Conference on 3D Vision (3DV 2017)* (2017). arXiv:1611.09813v5. 1, 4

[MSS*17] MEHTA D., SRIDHAR S., SOTNYCHENKO O., RHODIN H., SHAFIEI M. M.-H. M. M.-H. M. M.-H., SEIDEL H.-P., CASAS D., THEOBALT C., SHAFIEI M. M.-H. M. M.-H. M. M.-H., SEIDEL H.-P., XU W.: VNect Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Transactions on Graphics (SIGGRAPH 2017) 36*, 14 (2017), 1–13. URL: https://arxiv.org/abs/1705.01583, arXiv:1705.01583. 1, 2

[PFGA19] PAVLLO D., FEICHTENHOFER C., GRANGIER D., AULI M.: 3D human pose estimation in video with temporal convolutions and semi-supervised training. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019-June* (2019), 7745–7754. arXiv:1811.11742, doi:10.1109/CVPR.2019.00794. 2

[SBB10] SIGAL L., BALAN A. O., BLACK M. J.: HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision 87*, 1-2 (2010), 4–27. doi:10.1007/s11263-009-0273-6. 1

[SUF*20] SHI M., UNIVERSITY S., FILM ACADEMY KFIR ABERMAN B., ABERMAN K., ARISTIDOU A., KOMURA T., LISCHINSKI D.: MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency. *ACM Transactions on Graphics 40* (2020). URL: https://doi.org/10.1145/3407659, arXiv:2006.12075v1, doi:10.1145/3407659. 1, 2

[SXLW19] SUN K., XIAO B., LIU D., WANG J.: Deep high-resolution representation learning for human pose estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019-June* (2019), 5686–5696. arXiv:1902.09212, doi:10.1109/CVPR.2019.00584. 1, 2

[TS14] TOSHEV A., SZEGEDY C.: DeepPose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (sep 2014), IEEE Computer Society, pp. 1653–1660. arXiv:1312.4659, doi:10.1109/CVPR.2014.214. 4

[WWTL21] WEI G., WU S., TANG K., LI G.: BoneNet: Real-time 3D Human Pose Estimation By Generating Multiple Hypotheses with Bone-map Representation. URL: https://doi.org/10.14733/cadaps.2021.1448-1465, doi:10.14733/cadaps.2021.1448-1465. 2

[WYXL20] WANG J., YAN S., XIONG Y., LIN D.: Motion Guided 3D Pose Estimation from Videos. *European Conference on Computer Vision (ECCV)* (2020). URL: https://www.youtube.com/watch?v=VHhsXG6OXnI&t=87s., arXiv:2004.13985v1. 1, 2

[XYN*20] XU J., YU Z., NI B., YANG J., YANG X., ZHANG W.: Deep Kinematics Analysis for Monocular 3D Human Pose Estimation. *Proceedings of the IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition (2020), 896–905. doi:10.1109/CVPR42600.2020.00098. 2

[YAC19] YIANNAKIDES A., ARISTIDOU A., CHRYSANTHOU Y.: Real-time 3D human pose and motion reconstruction from monocular RGB videos. *Computer Animation and Virtual Worlds 30*, 3-4 (may 2019). doi:10.1002/cav.1887. 2

[YOW*18] YANG W., OUYANG W., WANG X., REN J., LI H., WANG X.: 3D Human Pose Estimation in the Wild by Adversarial Learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (mar 2018). URL: http://arxiv.org/abs/1803.09722, arXiv:1803.09722. 2