From Synthetic to One-Shot Regression of Camera-Agnostic Human Performances

From Synthetic to One-shot Regression of Camera-Agnostic Human Performances *

Julian Habekost[1], Kunkun Pang[1], Takaaki Shiratori[2], and Taku Komura[1]

[1] School of Informatics, University of Edinburgh, Edinburgh, UK
{julian.habekost,k.pang,tkomura}@ed.ac.uk
[2] Facebook Reality Labs, USA
tshiratori@fb.com

Abstract. Capturing accurate 3D human performances in global space from a static monocular video is an ill-posed problem. It requires solving various depth ambiguities and information about the camera's intrinsics and extrinsics. Therefore, most methods either learn on given cameras or require to know the camera's parameters. We instead show that a camera's extrinsics and intrinsics can be regressed jointly with human's position in global space, joint angles and body shape only from long sequences of 2D motion estimates. We exploit a static camera's constant parameters by training a model that can be applied to sequences with arbitrary length with only a single forward pass while allowing full bidirectional information flow. We show that full temporal information flow is especially necessary when improving consistency through an adversarial network. Our training dataset is exclusively synthetic, and no domain adaptation is used. We achieve one of the best Human3.6M joint's error performances for models that do not use the Human3.6M training data.

Keywords: Human Performance · Monocular Video · Synthetic Data.

1 Introduction

3D human performance estimation from monocular videos is a challenging topic that attracts researchers attention from various areas such as computer animation, virtual reality, surveillance, health-care etc. One major problem is that this task is entangled with the camera: If a person is lying or standing straight on the floor can either be judged through a high level of visual understanding of the surroundings or other information about the camera angle and position. This is why the early human pose estimation task is only concerned with camera-relative body poses [15, 16]. But the difference between laying and standing matters for performance capture, so subsequent work started to assume the camera intrinsics and extrinsics as given [25, 27].

We instead propose a method that learns to regress the extrinsics and intrinsics implicitly and explicitly together with the 3D performance from 2D human

* Supported by Facebook Reality Labs

A. Scale based Depth Ambiguity B. FoV based Depth Ambiguity

Fig. 1. Depth ambiguities of unknown monocular cameras. For all the depicted cases the person appears similar sized at the camera's viewfinder. We cannot differentiate between smaller people close and taller people further away (A.). Second, the field of view (FoV) influences how near or far objects and people appear (B.).

motion. The camera is unknown but static throughout one motion sequence and the motion is performed on a ground plane. Apart from the illustrated rotation ambiguity, this also is supposed to solve the FoV (field of view) depth ambiguity depicted in Figure 1 B. That is possible just from 2D motion can be understood when imaging a person coming towards the camera with a constant walking speed: A large FoV will make the person's projected 2D size increase quicker. Further, we can learn the ground plane implicitly through foot contacts and other types of interaction of the subject with the ground plane. To achieve this Our method uses

- a synthetic dataset that renders minute-long videos with various settings of the body shape, camera parameters, occluders and backgrounds,
- DensePose [24] to obtain an intermediate 2D motion representation,
- and a model that is able to do one-shot regressions on arbitrarily large sequences with global temporal information flow.

We show that global motion estimation and explicit camera estimation is possible with our method. Further, our method is the only domain generalization approach to the popular Human3.6M [7] dataset known to us. We also beat all Human3.6M domain adaptation tasks in local pose performance.

2 Related Work

Local pose estimation. This is the task of estimating a human's pose rotated relative to the camera and with the hip centered at the origin. Martinez et al. [15] regress 2D keypoint detections [2, 18] to 3D joint positions. Zhou et al. [28] directly incorporate a depth regression into a formerly 2D-only keypoint regressor. They use an adversarial pose prior when only 2D supervision is available. Kanazawa et al. [8] estimate both 3D pose and body shape based on SMPL [12], a parametric model of human shape and pose. Pavllo et al. [20] introduce a temporal model with dilated convolutions.

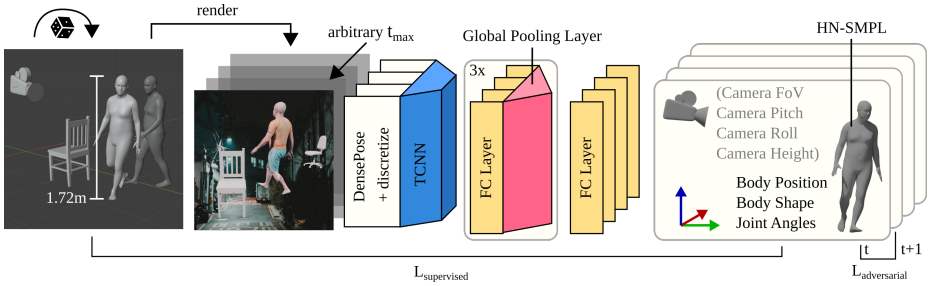

Fig. 2. We render differently body shaped subjects scaled to a body height of 172cm together with small occlusions in front of a random background. DensePose [24] gives us projected UV maps that we then discretize for training feasibility. The global pooling enables bidirectional information flow without temporal limits. The HN-SMPL is a height normalized PCA-based subspace of SMPL.

Apart from our focus on the harder task of estimating global motion rotated relative to the world, another contrast is that these methods assume the camera intrinsics or field of view as known; in most cases [8,10,15,16,19,20,28] implicitly through training and evaluating on the same cameras in Human3.6M [7].

Global pose estimation. Here the estimated human poses are rotated and translated relative to the world's ground plane. Mehta et al. [17] use procrustes alignment style post-processing on per frame estimations with a given camera to obtain a temporally consistent real time global pose from a local pose estimator. Shimada et al. [25] establish a baseline by optimizing 2D and 3D keypoints from a local pose estimator [9] to match a reprojection loss given the camera extrinsics and intrinsics. This shows that with known camera and 2D keypoints the task of local and global pose estimation are equivalent up to a classic test-time optimization problem. It has been extended [25,27] by including temporal physical simulations and constraints into this projection based optimization loop. Rempe et al. [22] use the same optimization loop with a motion VAE instead of a physics simulation. All of these methods need camera intrinsics and extrinsics to be known, which our work explicitly does not rely on. Further all of these global pose works are based on some kind of test-time optimization; only ours is a purely one shot regression.

Domain adaptation. Theoretically, training a deep model with fewer data could lead to worse generalisation performance [1, 5]. This problem becomes more prominent if there is insufficient data on the target domain. To overcome this, researchers proposed to reformalise the pose estimation as a domain adaptation problem. Chen et al. [3] adapted the trained model to Human3.6M by fine-tuning in a supervised manner, whereas Chen et al. [4] and Habekost et al. [6] applied domain adversarial learning without using ground truth. Although these models can learn the dataset-specific intrinsics (i.e. same for training and testing), they may not be suitable for the task without a training set. Apart from this,

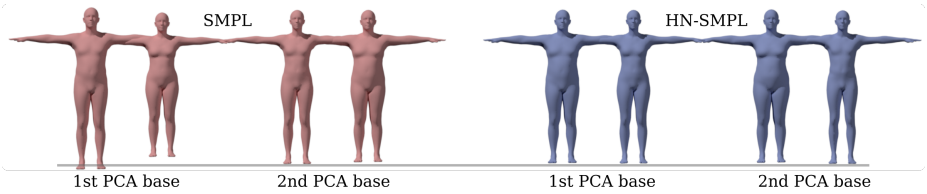

Fig. 3. HN-SMPL (Height-Normalized-SMPL) expresses all body shape variation except total body height through the bases of a PCA. The pairs show +1 and −1 of the noted PCA-basis. Note that even though less visible, height variation also does occur on other than the 1st SMPL PCA-base.

some works such as [9, 10, 23] use unpaired or unlabeled data, they use it to complement the supervised learning on Human3.6M.

Here, we go one step further and train the system without collecting the training data or any testing data except for the specific example of interest from the target dataset. The system will be trained on a synthetic dataset and generalize to an unseen Human3.6M dataset. This makes our work unique from existing works.

3 Synthetic Training Data Generation

Height Normalization. Due to the scale ambiguity of monocular video, it is impossible to estimate the height of a subject without reference or calibration (see figure 1A). We, therefore, assume that every subject has the height of 1.72 m, which is the average human height implied by the SMPL model. We circumvent the same scaling ambiguity for translation estimates by this assumption. Consequentially we predict the subject's body-height-normalized body shape and the subject's body-height-normalized translation. If the subject's real body height is known, the normalized translation can easily be scaled by the ratio between known and normalized body height to obtain a translation that adheres to scale.

The PCA-based SMPL body shape space implicitly and non-trivially embeds the total height. We propose a similar body shape space but with a fixed height. We sample the SMPL with random body shapes and T-pose angles, normalize the vertices by scaling the height difference between the top and bottom vertices to 1.72 m. These samples of scaled SMPL meshes are used to fit a new PCA, which then embeds a body space that is independent of the absolute body height (see 3). We only take the eight principal components. In order to be able to map back to the SMPL shape space, we fit the SMPL model to our height normalized body mesh by iterative gradient descent. We then learn a mapping from the new space to the SMPL body shape space with a simple neural network Θ with two layers and 16 hidden units each. To infer the mesh or joint positions from an HN-SMPL body shape β_{HN}, the SMPL body shape $\beta = \Theta(\beta^{HN})$ can be calculated.

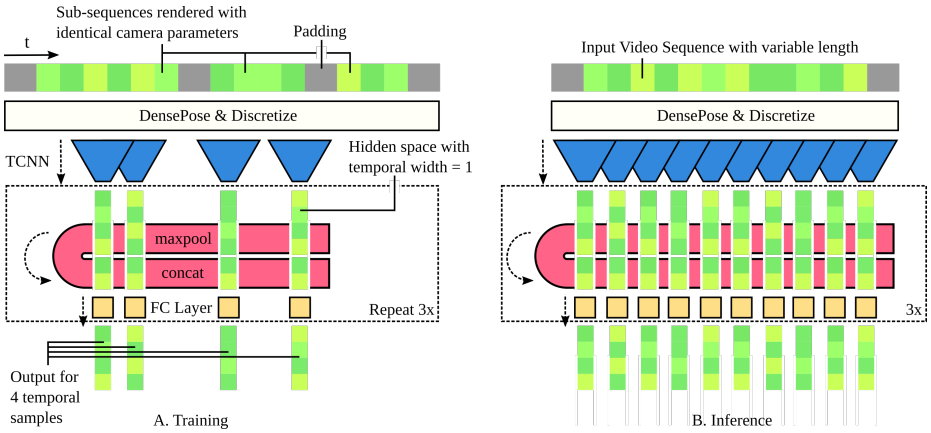

Fig. 4. The neural network regression model in detail. Input, output and hidden space are depicted in green. Processing steps and network functions have black borders around them. Similarly shaped and colored network functions share their weights. The forward pass direction is depicted through dotted lines.

Synthetic Sequence Rendering. For each video sequence S to render we sample a body shape $\beta_S^{HN} \sim \mathcal{N}(\mathbf{1}_{10}, \mathbf{0}_{10})$ and retarget all motions in S to this shape. We, therefore, randomly sample motion clips from the AMASS dataset. After retargeting, we randomly rotate the motion clip perpendicular to the ground plane and randomly translate it by $T_{xy} \sim \mathcal{N}(\mathbf{3}_2, \mathbf{0}_2)$ parallel to the ground plane. Each sequence has a uniquely random camera with a different pitch, roll, height and field of view. Only the yaw is kept such that the camera looks down the Z-axis because the camera's yaw is arbitrary and not inferable. Our random distributions cover most reasonable settings to capture human performances, from extremely close to exceptionally far away, from slightly up-looking to almost from above.

We simply concatenate the motion clips until a minimum of 60 seconds or 600 frames at 10fps is reached. Because we don't truncate sequences, the total length is variable and effectively a Poisson distribution starting at frame 600. Most sequences have a total length of around 700, but some rare examples of up to 2000 frame lengths exist.

Note that we do not transition between motion clips within a video sequence and account for this later in the model. If we only use long, continuous motion clips, the dataset size and variety would be reduced. Further, long motion clips often revolve around similar and repetitive actions, yielding global predictability that we want to avoid as our model could overfit it.

IUV Discretization We choose IUV-Maps obtained with DensePose as an intermediate representation. This aims to leave the domain gap between simulated and real data to DensePose. This also reduces the data dimensionality while

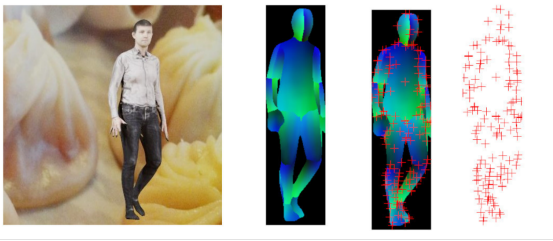

Fig. 5. The process of UV discretization visualized. DensePose [24] (B.) gives a dense map of the UV coordinates in image space. There are 178 discrete points (D.) in the final set.

keeping most of the useful information. We further reduce and adapt it to our sequence model by discretizing the dense IUV maps.

DensPose gives a three-channeled dense representation $D(x, y) \in \mathbb{N} \times \mathbb{R}^2$ containing the part index $D(x, y)_i$ with the highest posterior probability $P(i = p|x, y)$ and the part-specific projected U- and V-coordinates $D(x, y)_{uv}$ at the image pixels x, y. We heuristically choose a set of 178 IUV-coordinates based on their distance to each other. For each predefined IUV-coordinate C we exhaustively search in the image's pixel space x, y:

$$D(x, y)_i = C_i \tag{1}$$

$$d_{uv} = \|D(x, y)_{uv} - C_{uv}\|_2 \tag{2}$$

$$\min_{x,y} d_{uv}. \tag{3}$$

We don't specify a threshold for the part-specific UV-distance d_{uv}. As long as the part p is visible, all $C, C_i = p$ for that part will be found. As a result some C lie on the boundary of the part's segmentation mask, which gives a rough indication of the body part's shape.

4 Model

Network design. Similar to [6], our model Φ is composed of a temporal convolutional neural network (TCNN). The TCNN aggregates local features of a 32 frame wide neighborhood. Each 1D convolutional layer down-samples the temporal space by half with stride 2. Five convolutional layers are necessary to arrive at a hidden space with a temporal width of one. In figure 4,the max-pooling is applied on the hidden units over the entire temporal channel so that the model can extract a sequence-level information flow. This method is inspired by PointNet [21] and allows scaling the amount of temporal samples to an arbitrary length in a single regression. The max-pooling results are concatenated and a feed-forward layer is applied. These steps of max-pooling, concatenation, and fully-connected layer are repeated three times. The last fully-connected layer

produces the model's output for every temporal step that the regression has originally been applied to. We use a fixed number of temporal neighborhoods sampled from the sequence in training. In inference, our model can regress the whole input sequence in one shot.

Network loss. For each time step t, our network Φ maps the discretized IUV coordinates $c = \{\{c_{xi}, c_{yi}\}_1, ...\{c_{xi}, c_{yi}\}_{178}\} \in \mathbb{R}^{356}$ to a HN-SMPL body shape $\beta^{HN} \in \mathbb{R}^8$, 24 joint angles $\theta \in \mathbb{R}^{72}$, a 3D translation $T \in \mathbb{R}^3$ and a camera view $V = \{pitch, roll, height, fov\} \in \mathbb{R}^4$ (see figure 2). The objective function consists of an adversarial loss, and a weighted L1-loss between those mentioned components $X = \{\beta^{HN}, \theta, T, V\}$ and the network prediction $\Phi(c)$:

$$\ell = \sum_t (\sum_X l_X \cdot L_1(X, \Phi(c_t)) + L_{adversarial} \tag{4}$$

Table 1. Errors of camera estimation on the CP dataset, compared to methods that use images for camera parameter estimation.

Method	**FoV (°)**	**Pitch (°)**	**Height (mm)**
ScaleNet [29]	3.63	2.11	-
CamCalib [11]	3.14	1.80	-
Ours (linear)	3.84	2.95	32.8
Ours (adversarial + pooling)	3.92	3.21	36.1

5 Experiments

Datasets. Our synthetic training dataset uses motion sequences from **AMASS** [14] rendered with textures from **SURREAL** [26] and random CC0-licensed images as background. The horizontal field of view is drawn from $[80°, 34°]$ for each sequence. We chose square 1000×1000 pixel images, which results in the vertical field of view being the same.

For evaluation we use subjects 9 and 11 of **Human3.6M** [7]. The other subjects are not used in this work. Note that the Human3.6M field of view is 46.4°. We report the mean per joint error (MPJE) and procrustes-aligned mean per joint error (PA-MPJE). We also make our own evaluation dataset where people walk in circles from different perspectives (**CP**, see figure 8). Each of the 25 sequences produced by one of the five different subjects has a different camera angle and focal length in a distribution similar to the training dataset. We only obtain the motion starting position as the ground truth and only report the translation error to this starting position.

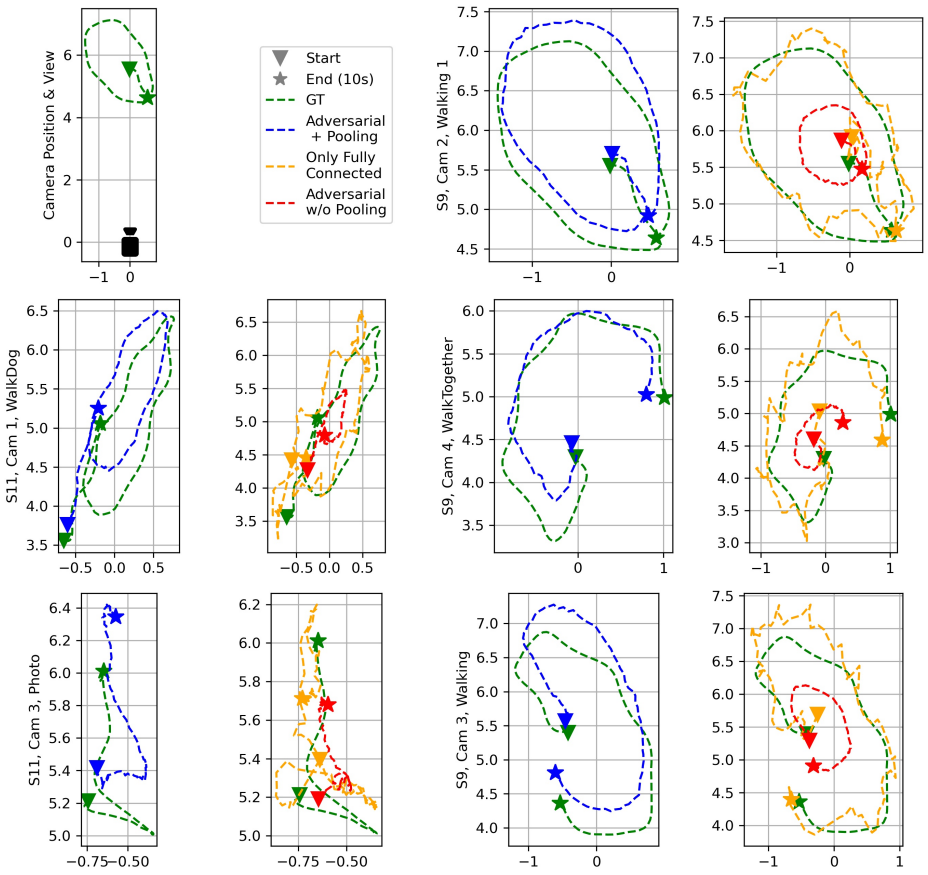

Fig. 6. Translation results on the first 10 seconds of Human3.6M sequences viewed from above. After the overview and legend, each two plots next to each other belong to the same sequence.

Training. We train on synthetic dataset sizes of 10k, 20k and 70k sequences until the performance of the identical 1k validation sequences cannot be further improved. We report our model's results with the 70k training sample size, except when stated otherwise. In training, we sample 16 temporal neighborhoods from each sequence. The neural network has 6 linear layers with 2048 hidden units and concatenate 512 max-pooled hidden units, Adam [13] as the optimizer and a learning rate of $1e^{-4}$. We set $l_\theta = 1000, l_\beta = 10, l_T = 100$ and $l_V = 1$. We report results with the 70k training set model, except when stated otherwise.

Results. Table 3 shows our local pose performance on Human3.6M. We are only beaten by methods that integrate supervised training into their model. Besides, we beat all domain adaptation methods on Human3.6M. This result is reasonable. The supervised learning approaches learn the dataset-specific camera

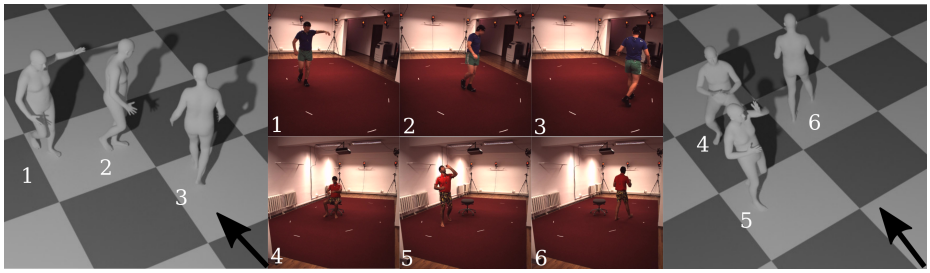

Fig. 7. Qualitative results of two Human3.6m test sequences. Both the pose and the translations are depicted. The camera is out of the rendered frame but its direction is shown with a black arrow.

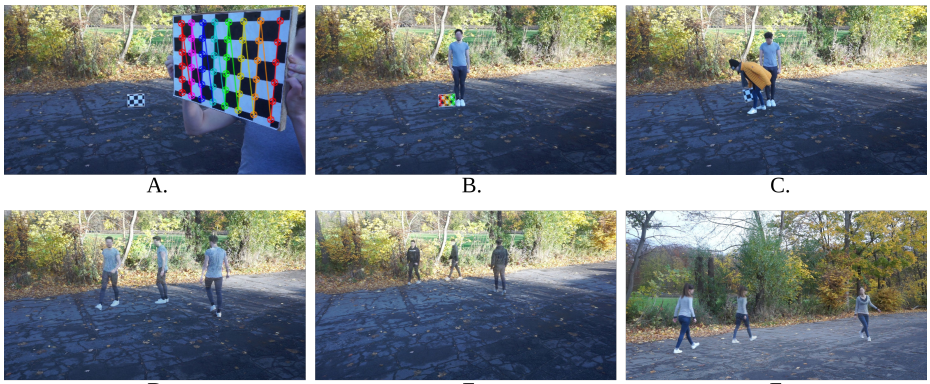

Fig. 8. Our **CP** evaluation dataset. For different field of views, the camera is first calibrated (A). A marker on the floor (B) is used to infer the ground truth motion starting position.

intrinsics and have no domain gap, whereas our approach does not have domain knowledge and intrinsics.

Figure 6 shows a qualitative evaluation of the translation. It shows that the adversarial network pushes the model towards a more stable, smoother and more realistic overall motion shape. But without global pooling, the adversarial network's regime seems too strong; the model cannot infer the overall motion shape with only local temporal knowledge and collapses into a smaller motion.

As shown in the table 2, the fully connected layer with the TCNN seems to deliver a slightly better performance than our adversarial approach on the 70k dataset. The gap becomes more extensive when a smaller training dataset is used, which is also reflected in the differences between training and validation data. This gap cannot be explained via a domain gap but simply because the training set is too small for enabling the model to learn the synthetic domain perfectly and not overfit to the training set. This is the reason why we believe

Table 2. Translation errors (in mm, non-procrustes) for different training set sizes. Both train and val are synthetic datasets.

Method	10k				20k				70k			
	train	**val**	**H36**	**CP**	**train**	**val**	**H36**	**CP**	**train**	**val**	**H36**	**CP**
Only FC	248	482	500	843	293	395	426	683	294	380	406	531
Advers. + Pool.	308	694	915	1072	323	476	721	882	315	414	421	539

Table 3. Evaluation results on Human3.6M sorted by MPJPE. Domain adaptation (DA) methods use Human3.6M training videos together with a non-GT based supervision. Supervised (S) papers directly train on the Human3.6M training set. Methods with implicit camera have seen the Human3.6M cameras, which are identical for training and testing. Domain generalization (DG) methods have not seen any images, cameras or motion from Human3.6M.

Method	**MPJPE↓**	**PA-MPJPE(↓)**	**Task**	**Camera**
Chen et al. [3]	136.1	-	DA	implicit
Habekost et al. [6]	118.2	-	DA	given
Chen et al. [4]	-	68.0	DA	implicit
Kanazawa et al. [8] unpaired	106.8	66.5	DA	implicit
Shimada et al. [25]	97.4	65.1	S	given
Kanazawa et al. [8] paired	88.0	56.8	S	implicit
Ours	87.9	66.9	**DG**	**unseen**
Rhodin et al. [23]	66.8	51.6	S	implicit
Kocabas et al. [10]	65.6	41.4	S	implicit
Martinez et al. [15]	62.9	47.7	S	implicit

there is still room for improving the proposed method's translation accuracy by generating an even larger dataset.

There are methods [25, 27] that report the Human3.6m translation error we report in table 2 and they are significantly better. This is simply due to the methods assuming known camera parameters. But when camera parameters are unknown, like for the CP dataset, these do not work. Hence it is impossible to infer the subject's translation in our CP dataset, which we also report in table 2. We argue that implicit camera estimation is half of the work necessary for out model to infer the subject's translation and a comparison would be unfair.

Table 1 shows our camera estimation results compared to other deep learning methods on image data. We do not expect to outperform these models but merely show that the results are sensible and can be regressed from only 2D human motion. Also, only our method can infer the camera's height (inversely scaled to the human's size in the video). Note that our method can regress the camera parameters in one shot. We show this to convince our reader that our model can implicitly learn the camera estimation necessary for global pose estimation.

6 Conclusion

By using a large scale synthetic dataset coupled with DensePose [24] as a 2D human motion extractor and an one-shot regression network that encourages global information flow, we show that human performance capture is possible without externally calibrated cameras or visual understanding of the surroundings. We demonstrate that the camera's intrinsics and extrinsics can even be estimated explicitly. To our knowledge, by doing so we have created the first domain generalization approach for the popular Human3.6M [7] dataset. Our result is comparable with some supervised learning approaches and outperforms all domain adaptation based methods on Human3.6m's local pose performance.

References

1. Arora, S., Ge, R., Neyshabur, B., Zhang, Y.: Stronger generalization bounds for deep nets via a compression approach. In: ICML (2018)
2. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017)
3. Chen, C.H., Ramanan, D.: 3d human pose estimation = 2d pose estimation + matching. In: CVPR (2017)
4. Chen, C.H., Tyagi, A., Agrawal, A., Drover, D., MV, R., Stojanov, S., Rehg, J.M.: Unsupervised 3d pose estimation with geometric self-supervision. In: CVPR (2019)
5. Fengxiang He, D.T.: Recent advances in deep learning theory. CoRR **abs/2012.10931** (2020)
6. Habekost, J., Shiratori, T., Ye, Y., Komura, T.: Learning 3d global human motion estimation from unpaired, disjoint datasets. In: BMVC (2020)
7. 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 (2014)
8. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR (2018)
9. Kanazawa, A., Zhang, J.Y., Felsen, P., Malik, J.: Learning 3d human dynamics from video. In: CVPR (2019)
10. Kocabas, M., Athanasiou, N., Black, M.J.: Vibe: Video inference for human body pose and shape estimation. arXiv:1912.05656 (2019)
11. Kocabas, M., Huang, C.H.P., Tesch, J., Müller, L., Hilliges, O., Black, M.J.: SPEC: Seeing people in the wild with an estimated camera. In: ICCV (2021)
12. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. SIGGRAPH Asia (2015)
13. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam. CoRR **abs/1711.05101** (2017)
14. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: ICCV (2019)
15. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: ICCV (2017)
16. 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. In: 3DV (2017)

17. Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.P., Xu, W., Casas, D., Theobalt, C.: Vnect: Real-time 3d human pose estimation with a single rgb camera. ACM Transactions on Graphics (TOG) (2017)
18. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV (2016)
19. Pavlakos, G., Zhu, L., Zhou, X., Daniilidis, K.: Learning to estimate 3D human pose and shape from a single color image. In: CVPR (2018)
20. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. arXiv **abs/1811.11742** (2018)
21. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv preprint arXiv:1612.00593 (2016)
22. Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., Guibas, L.J.: Humor: 3d human motion model for robust pose estimation. In: ICCV (2021)
23. Rhodin, H., Sporri, J., Katircioglu, I., Constantin, V., Meyer, F., Erich Muller, M.S., Fua, P.: Learning monocular 3d human pose estimation from multi-view images. In: CVPR (2020)
24. Riza Alp Guler, Natalia Neverova, I.K.: Densepose: Dense human pose estimation in the wild. In: CVPR (2018)
25. Shimada, S., Golyanik, V., Xu, W., Theobalt, C.: Physcap: Physically plausible monocular 3d motion capture in real time. ACM Transactions on Graphics (TOG) (2020)
26. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: CVPR (2017)
27. Xie, K., Wang, T., Iqbal, U., Guo, Y., Fidler, S., Shkurti, F.: Physics-based human motion estimation and synthesis from videos. In: ICCV (2021)
28. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Towards 3d human pose estimation in the wild: A weakly-supervised approach. In: ICCV (2017)
29. Zhu, R., Yang, X., Hold-Geoffroy, Y., Perazzi, F., Eisenmann, J., Sunkavalli, K., Chandraker, M.: Single view metrology in the wild. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) ECCV (2020)