TexturePose: Supervising Human Mesh Estimation with Texture Consistency **Supplementary Material**

Georgios Pavlakos*, Nikos Kolotouros*, Kostas Daniilidis
University of Pennsylvania

In this Supplementary Material, we will provide more details that were not included in the main manuscript due to space constraints. Section 1 provides more quantitative results for our approach using additional datasets for training and evaluation. Section 2 extends the qualitative evaluation of the main manuscript, providing more example reconstructions, including both success and failure cases. Then, Section 3 clarifies the settings of the empirical evaluation, while Section 4 presents more details about the training procedure. Finally, in Section 5, we define the evaluation metrics employed in the quantitative evaluation.

1. Further quantitative evaluation

For the additional quantitative evaluation, we present results using also the VLOG-People and InstaVariety datasets [5] for training. The joints for these datasets are provided automatically using OpenPose [1, 2, 8, 9], which means that we leverage videos that provide only pseudo-annotations for 2D joints. Similarly to our training in the main manuscript, we use groups of five frames, where only one contains 2D joints annotations, while for the rest we only enforce the texture consistency loss. The data we used before (i.e., Human3.6M and MPII video) is still used here. We train two models, one with and one without the texture consistency loss. We present results for different datasets, i.e., Human3.6M (Table 1), LSP (Table 2) and 3DPW (Table 3). Similarly to our findings in the main manuscript, the version trained with the texture consistency loss is consistently outperforming the vanilla model without this loss.

2. Further qualitative evaluation

In this Section we extend the qualitative evaluation of Subsection 4.3 of the main manuscript, always employing the network that is used to report results in Tables 3 and 4 of the main manuscript. In Figure 1 we provide additional successful reconstructions including novel viewpoints, which are typically useful to assess the reconstruction quality of a monocular approach.

Moreover, in Figure 2 and Figure 3 we provide examples where our approach fails to recover a correct shape estimate.

	P1	P2
Ours + data of [5]	51.5	49.2
Ours + data of [5] + texture	48.9	46.1

Table 1: Evaluation on the Human3.6M dataset (Protocols 1 & 2), using the additional data from VLOG-People and InstaVariety. The numbers are mean reconstruction errors. We evaluate our models trained with and without the use of texture consistency.

	FB Seg.		Part Seg.	
	acc.	f1	acc.	f1
Ours + data of [5]	91.75	0.87	88.83	0.66
Ours + data of [5] + texture	92.12	0.88	89.26	0.67

Table 2: Evaluation on foreground-background and six-part segmentation on the LSP test set, using the additional data from VLOG-People and InstaVariety. We evaluate our models trained with and without the use of texture consistency.

	Absolute	Procrustes
Ours + data of [5]	157.0	107.5
Ours + data of [5] + texture	142.5	101.2

Table 3: Evaluation on the 3DPW dataset, using the additional data from VLOG-People and InstaVariety. The numbers are mean per-vertex errors without and with Procrustes alignment ('Absolute' and 'Procrustes' respectively). We evaluate our models trained with and without the use of texture consistency.

These failures can give us intuition on the failure modes of our approach and the ways we can improve upon it. Typical failures include cases with interactions between multiple people, poses with increased level of self-occlusions, or ordinal depth ambiguities.

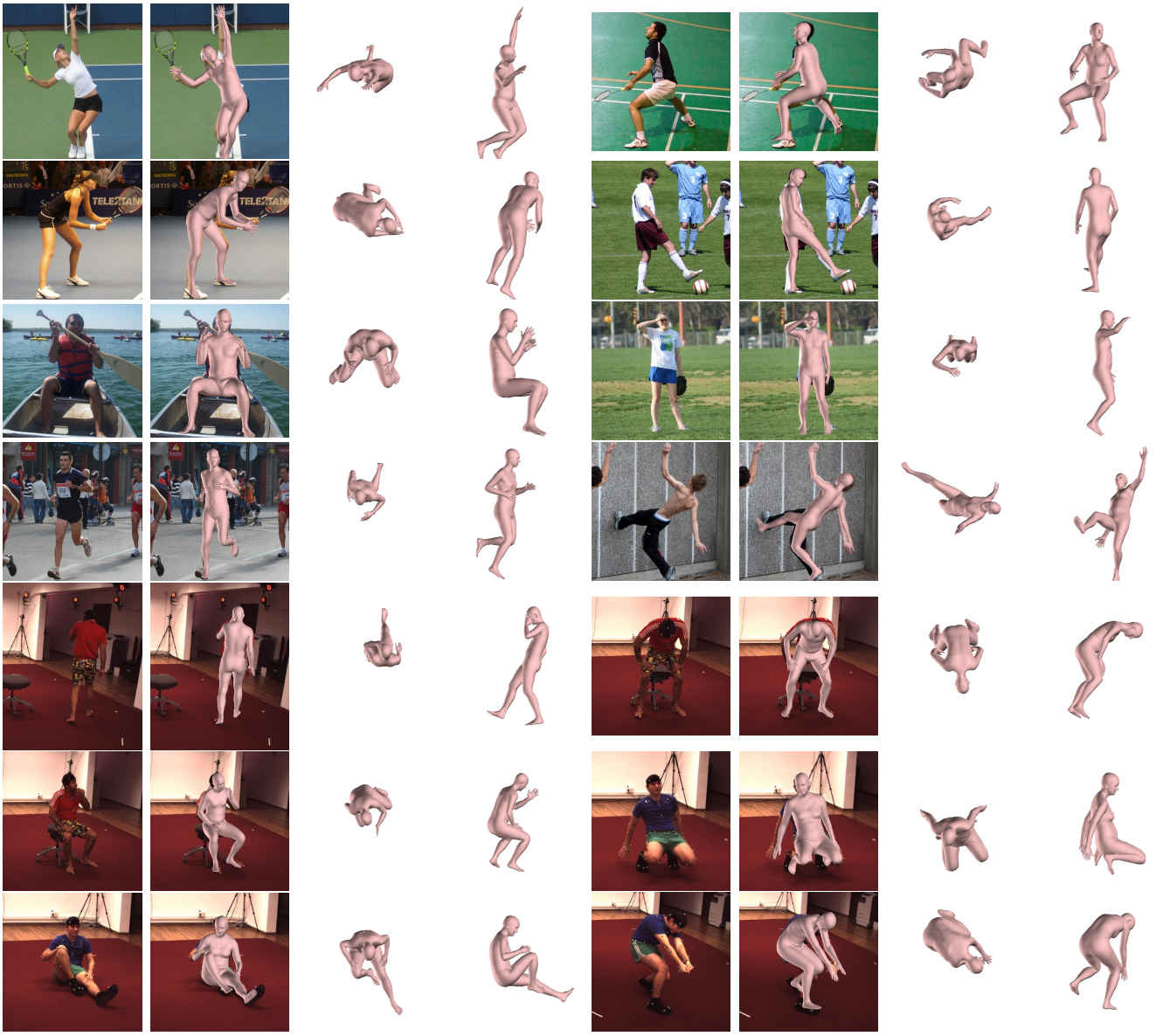

Figure 1: Successful reconstructions of our approach (with the network that is used to report results in Tables 3 and 4 of the main manuscript). For each example from left to right: Image, Our reconstruction result, Our reconstruction result from a novel viewpoint (top), our reconstruction result from a novel viewpoint (side).

3. Experimental settings

In this Section we provide more details about our training and testing settings. Although we stressed it in the main manuscript as well, we re-iterate here that although training happens with group of images, i.e., with frames from a video or from different viewpoints of a time-synchronized multi-view setup, *at test time, our network takes a single image* and predicts the 3D body pose and shape for the person in the image. In the following paragraphs, we clarify some of the details for the different settings involved in our experimental evaluation.

Human3.6M (monocular): This corresponds to the results reported in Table 1 of the main manuscript. The training includes sets of five consecutive frames from the Human3.6M dataset, as presented in Figure 4. The only difference for the various settings concerns the level of annotations that each setting has available, or the level of supervision (loss terms) that we enforce. For the first row of Table 1, we handle each frame independently, applying a L_{2D} loss

Image Result Image Result

Figure 2: Typical failure cases of our approach. Failures include wrong orientation of the head, confusion because of multiple people in the scene, and mis-alignment between the model and image, particularly for the extremities.

Image Result Novel view

Figure 3: Further failure cases of our approach that require visualization from a novel viewpoint to properly identify the error in the reconstruction. Ordinal depth ambiguities and self-occlusions are typically responsible for these failures.

and $L_{\text{adv prior}}$ loss. This is equivalent to the "unpaired" setting of [4], where we have no frame with 3D ground truth and we have access only to 2D keypoints and an independent set of 3D pose/shape parameters that we use to learn an (adversarial) prior. For the second row, we keep the L_{2D} loss and $L_{\text{adv prior}}$ terms, but we also add the texture consistency term $L_{\text{texture cons}}$ between all pairs of images. We clarify that $L_{\text{shape cons}}$ is always used when we apply $L_{\text{texture cons}}$, so we do not mention it separately from now on. We also stress that no additional annotation is available for this second row, and we simply leverage a property of natural images to improve our supervision. This auxiliary supervision provides further constraints for the recovered pose and improves learning compared to a weak prior that only informs us whether the recovered pose is valid or not. Finally, for the third row, we use full 3D ground truth for supervision, L_{3D}, which is a parameter loss on the pose and shape parameters, as well as a per-vertex loss on the recovered mesh. Since this is the most informative form of supervision, the result of this setting acts as a lower limit of the performance we can hope we can achieve with any form of auxiliary supervision. Despite texture consistency comes effectively for free from videos, we are able to improve significantly over the initial baseline, and in fact it comes very close to the lower limit set by full 3D supervision.

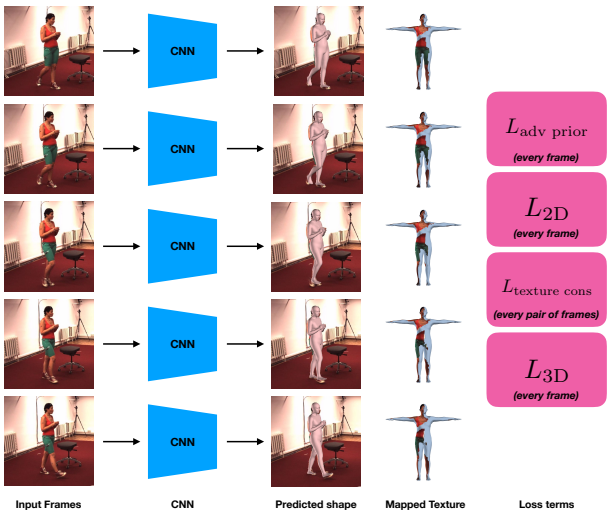

Figure 4: Training procedure using the data from Human3.6M to train with monocular video sequences. The figure corresponds to the experimental setting of Table 1 of the main manuscript. For each experiment of the ablative we provide different forms of supervision to the network, i.e., only specific loss terms are active. The details are clarified in the text (Section 3).

Human3.6M + MPII (monocular): For this second experiment, presented in Table 2 of the main manuscript, we keep images from Human3.6M using full 3D ground truth, L_{3D}, for supervision, but we also include in-the-wild images from MPII in our training, with weaker annotations/supervision. This means that we follow a mixed training strategy, where our batches include images from both datasets. In Figure 5, we have focused particularly on the different settings we use the in-the-wild MPII images, considering that this is the main factor changing in this ablative experiment. Initially (row 1), we use only images from Human3.6M with full 3D pose and shape ground truth as supervision. This is the same experiment as the last row

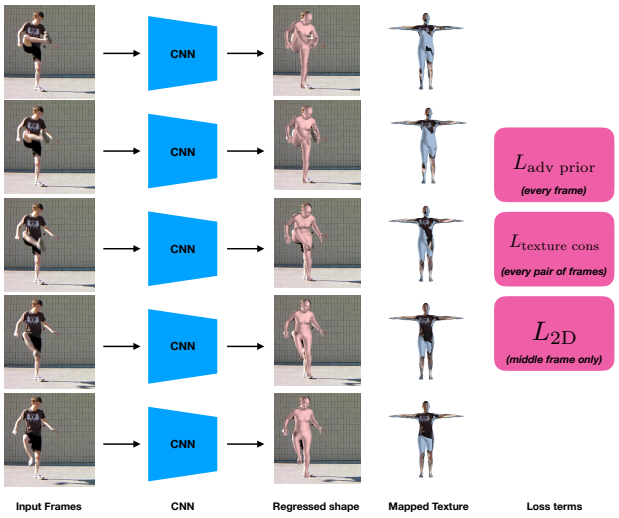

Figure 5: High level figure of the training procedure using the data from MPII video. The figure corresponds to the experimental setting of Table 2 of the main manuscript. For each experiment of the ablative we provide different forms of supervision to the network, i.e., only specific loss terms are active. The details are clarified in the text (Section 3).

of Table 1, and serves as our initial baseline. For the setting of row 2, we add images from MPII videos, but we do not give access to any 2D keypoint ground truth. As a result, only $L_{\text{adv prior}}$ is active, which forces the network to still produce valid poses, along with our texture consistency $L_{\text{texture cons}}$ that adds more constraints to the output. As we highlight in the main manuscript, this improves performance for Human3.6M, but does not give us satisfying results for in-the-wild images. For the next setting (row 3), we ignore video (i.e., remove consistency losses), and instead we provide additional supervision through a keypoint reprojection loss, L_{2D}, where the annotation is available *only for one frame (middle) of the short video*. This performs better than the previous setting, which is expected, since we explicitly added more annotations in our training. Finally (row 4), on top of the keypoint loss for the middle frame, we also add the consistency losses $L_{\text{texture cons}}$ for all pairs of frames. Despite the fact that we only use unlabeled data (i.e., neighboring frames with no annotations whatsoever), this setting further improves performance compared to the previous experiment, since we capitalize on the texture consistency of the subjects.

Human3.6M (multi-view): For the multi-view setting, which refers to Table 5 of the main manuscript, the training procedure is presented in Figure 6. This Figure visualizes the training for a specific time instance, where we have access to four viewpoints of the same subject. With the exception of a small part of the training data (i.e., images of subject S1), in this setting we use no annotations for the four views, other than the extrinsic calibration of the multi-view system. Besides a pose prior loss $L_{\text{adv prior}}$, we only enforce texture consistency through $L_{\text{texture cons}}$ and mesh consistency through $L_{\text{mesh cons}}$, for all pairs of frames.

4. Training details

Our model follows the architecture of Kanazawa *et al*. [4], where we only change the form of the output such that we regress 3D rotations (for the pose parameters) using the representation of Zhou *et al*. [10], instead of the axis-angle representation employed by [4]. Our model is trained using the Adam optimizer, with the learning rate initially set to $3e-4$, and reducing it by a factor of 0.9 every $100k$ iterations. The batch size is set to 60, sufficient to accommodate 12 sets of short-sequences with five frames (temporal), or equivalently 15 time instances captured by four viewpoints (multi-view). Training with data from Human3.6M lasts for $400k$ iterations, while when training with data from Human3.6M and MPII, we increase the number of iterations to $600k$. When using the VLOG-People and InstaVariety data as well, we further increase the iterations to $900k$.

During training, the most computationally intensive procedure is visibility computation for all surface points of the regressed mesh (or equivalently all texels of the texture map). On a GeForce 2080 Ti GPU, a forward/backward computation lasts for 3s when we need to compute visibility. The current implementation for visibility computation relies on an efficient CPU implementation [6]. We experimented with a GPU implementation based on raycasting, but it was particularly memory intensive. A potential implementation of the efficient algorithm of [6] on GPU could further accelerate computation. Since the visibility computation can be slow, we found it practical to activate the texture consistency loss in the middle of the training procedure, so that we can accelerate experimentation. Regardless of the slow training time, at test time, a single forward pass is very efficient, requiring less than 40ms.

5. Evaluation metrics

In this Section we discuss in more detail the evaluation metrics used to report results in the main manuscript. Since the segmentation metrics (reported in Table 4 of the main manuscript) are quite common in the literature, while the 3D pose metrics (reported in all the other Tables) typically incur more confusion, here we focus on the metrics that evaluate 3D pose.

MPJPE: MPJPE stands for Mean Per-Joint Position Error and is proposed by Ionescu *et al*. [3]. Given a predicted 3D pose $\hat{X} \in \mathbb{R}^{3\times k}$ and a ground truth 3D pose $X \in \mathbb{R}^{3\times k}$, MPJPE involves computation of the average Euclidean distance over all the joints, *after* aligning the root joint (typi-

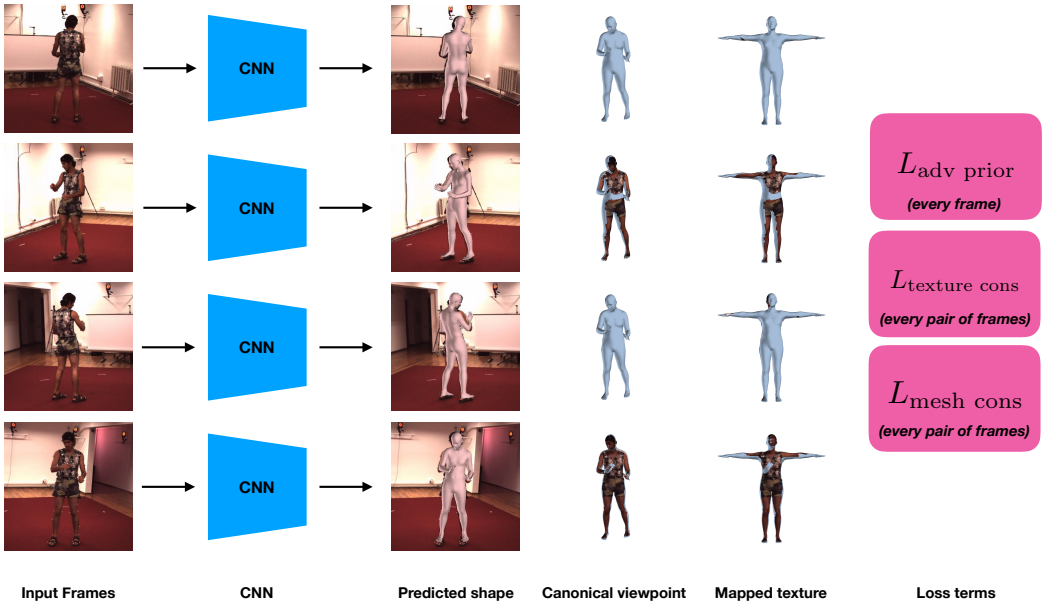

Figure 6: High level representation of the training procedure using the data from Human3.6M to train with multi-view images. This Figure corresponds to the experimental setting of Table 5 of the main manuscript. The details are clarified in the text (Section 3). Here, we visualize the regressed mesh in the camera view and in the canonical orientation (assuming the same global orientation for all viewpoints), along with the recovered texture mapped on the template shape. The missing (non-visible) texture for some viewpoints indicates that we have recovered the texture from the back side of the subject.

cally the pelvis) of the predicted 3D pose with the root joint of the ground truth 3D pose.

NMPJPE: NMPJPE or Normalized MPJPE is proposed by Rhodin *et al*. [7] and is a normalized version of the previous metric, which allows us to align the scale of our prediction with the scale of the ground truth 3D pose. This typically allows us to ignore differences in the size of the estimated skeleton, which are impossible to resolve anyway for a monocular method.

Rec. Error: Reconstruction Error or PMPJPE (Procrustes MPJPE) allows us to perform a Procrustes alignment with the ground truth skeleton before estimating the per-joint error. This means that we can also ignore errors in the global orientation of our estimate.

References

[1] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018. 1

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017. 1

[3] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014. 4

[4] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 3, 4

[5] Angjoo Kanazawa, Jason Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *CVPR*, 2019. 1

[6] MPI-IS. Mesh processing library. https://github.com/MPI-IS/mesh. 4

[7] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3D human pose estimation from multi-view images. In *CVPR*, 2018. 5

[8] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 1

[9] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 1

[10] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 4