Learning to Estimate 3D Human Pose and Shape from a Single Color Image Supplementary material

Georgios Pavlakos[1], Luyang Zhu[2], Xiaowei Zhou[3], Kostas Daniilidis[1]
[1] University of Pennsylvania [2] Peking University [3] State Key Lab of CAD&CG, Zhejiang University

This supplementary material provides additional details for our approach that were not included in the main manuscript due to space constraints. Sections 1 to 6 present additional qualitative results of our approach. Section 7 demonstrates a qualitative evaluation of the proposed per-vertex loss. Section 8 includes the exact description of the different architectures employed in our experiments. Section 9 provides further details regarding our implementation. Section 10 includes details relevant to our training procedure. Finally, Section 11 focuses on extensive qualitative evaluation and provides additional examples of the proposed approach on images from the selected datasets.

1. Individual benefit for shape/pose

Most of the results from the main manuscript focus on joint evaluation of pose and shape. To provide more insight into which component is influenced more from our proposed losses, we analyze further the results of Table 1 of the main manuscript. Focusing on the outputs using parameter loss and our proposed per-vertex loss, we provide further evaluations, using a) the predicted pose parameters and ground truth shape parameters, and b) the predicted shape parameters and ground truth pose parameters. The detailed results are presented in Table 1. From these results we can easily infer that the benefit comes primarily from predicting a more accurate pose. On the other hand, the shape influence in the final error is rather small and shape prediction is only marginally improved when we adopt the per-vertex loss. This demonstrates that 3D pose prediction is a very challenging problem and one that is properly addressed with the introduction of our per-vertex loss.

2. Additional ablatives for decision choices

To clarify and motivate some of our decision choices, we provide here the results of additional ablative experiments. Focusing on the setting of Table 1 of the main manuscript, we do further exploration by:

- training without parameter loss and keeping only the per-vertex loss active.
- ignoring the keypoint detection confidences of *Human2D*, and not using them as input to the *PosePrior* network, as we currently do.
- using a single network for pose and shape prediction, where the input is the channel-wise concatenation of the heatmaps and the masks from *Human2D*.
- using ground truth 2D keypoints and masks as input to the *Priors* networks.

The complete results for these experiments are presented in Table 2. The results of the first three cases justify some of our design choices, while the last experiment demonstrates the potential upper bound for our approach.

	Pred	GT shape	GT pose
Parameter loss (rot matrix)	140.7	133.6	30.3
+ Per-vertex loss	120.7	114.9	29.9

Table 1: Effect of the proposed per-vertex loss on pose and shape prediction individually. Most of the benefit comes from better pose prediction, while the shape has only small improvement. The numbers are mean per-vertex errors (mm). The results are reported using UP-3D and correspond to the setting of Table 1 of the main manuscript.

Alternative models	Avg Error
Our model	117.7
- no parameter loss	125.9
- no keypoint confidence	124.7
- combined *Pose/ShapePrior*	156.1
- GT 2D input	79.9

Table 2: Additional ablative studies motivating some of our decision choices. The numbers are mean per-vertex errors (mm). The results are reported using UP-3D and correspond to the setting of Table 1 of the main manuscript.

3. Detailed results on SURREAL

In Table 3 we provide the results for all the actions of the Human3.6M part of the SURREAL dataset [10]. This is the extended version of Table 3 of the main manuscript. Our approach outperforms the other baselines across the majority of actions and on average.

	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SitingD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Lassner *et al.* [5] (GT shape)	145.4	118.1	166.0	152.9	175.2	213.8	150.1	288.7	271.2	446.9	194.7	172.1	267.9	154.4	171.1	200.5
Bogo *et al.* [2] (GT shape)	129.0	**103.1**	148.3	**137.8**	153.8	191.8	140.0	251.2	237.9	392.4	176.2	154.6	238.3	134.1	137.7	177.2
Ours (GT shape)	**120.9**	111.1	**132.1**	138.1	**137.8**	**154.1**	**137.3**	**166.4**	**200.0**	**252.5**	**167.8**	**151.5**	**150.3**	**131.1**	**129.3**	**151.5**
Bogo *et al.* [2]	155.6	129.9	172.0	165.4	180.0	214.6	166.6	273.5	265.3	409.3	204.7	179.0	259.6	157.1	161.5	202.0
Ours	**126.1**	**118.1**	**136.0**	**143.4**	**140.9**	**156.8**	**141.2**	**170.9**	**200.9**	**253.4**	**171.6**	**156.5**	**154.6**	**137.7**	**133.7**	**155.5**

Table 3: Detailed results on the Human3.6M part of SURREAL [10]. Numbers are mean per vertex errors (mm). For the first three rows, the shape coefficients are known, for the last two rows they are predicted.

4. Detailed results on Human3.6M

In Table 4 we provide the results for all the actions of Human3.6M dataset [3]. This is the extended version of Table 4 of the main manuscript. Our approach outperforms the other baselines across the majority of actions and on average.

5. Boosting SMPLify

The motivation for Section 5.5 of the main manuscript, was to demonstrate the benefit of using our direct predictions as an initialization and an anchor of iterative optimization methods to accelerate them and improve their results. Here, we provide more details on this aspect, by evaluating the performance of our anchor itself, and exploring using another direct prediction approach as an optimization anchor. For our experiments, we employed the direct prediction approach of Lassner *et al.* [5]. Using this anchor lead to slightly worse, yet comparable results with out approach. However, the optimization became 10% slower, compared to using our anchor, which indicates that our anchor was more accurate to begin with. Indeed, by evaluating the anchors themselves on the same task of person and part segmentation, we achieve significantly better performance with our anchor. The results of our direct prediction are worse than the ones obtained by SMPLify, since SMPLify explicitly optimizes for the 2D-3D consistency. However, this sometimes comes in the expense of obtaining a non-coherent 3D pose (check also sample reconstructions of Figure 10).

6. Keypoint localization

Following the evaluation of the previous section, focusing on the 2D aspect of our predictions, we also evaluate our predictions on the task of the 2D keypoint localization on the UP-3D dataset [5]. The complete results expressed in PCKh are presented in Table 6. The comparison includes the accuracy of our keypoint predictions from *Human2D*, as well as the 2D keypoints resulting from projecting the 3D joints of the body model on the 2D plane. Although the projection of the 3D joints is quite accurate, we observe that it still falls slightly behind the network trained exclusively for 2D keypoint localization. This observation is consistent with other approaches that are trained for 3D pose consistency in expense of highly accurate keypoint localization (e.g. [9]). We also observed that the reprojections of the 3D joints can perform more accurate 2D localization when we train longer using the reprojection losses, but typically this can start having a negative effect on the 3D reconstruction itself.

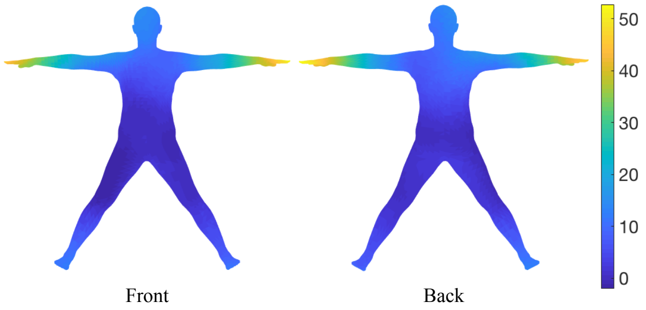

Figure 1: Qualitative evaluation of the benefit of the per-vertex loss. A human body model is visualized, where the color of each vertex corresponds to the improvement of the mean error of that vertex, after using a per-vertex loss for training (compared to using only a vanilla parameter loss). For the evaluation, we used all test examples of UP-3D. The numbers are expressed in mm.

7. Qualitative benefit of per-vertex loss

For a qualitative evaluation of the proposed per-vertex loss in comparison to simply using a loss only on the parameters, we visualize the parts of the human body that are mostly influenced by this improvement. Starting with the parameter loss as a baseline, in Figure 1 we visualize a human body model such that the color of each vertex represents the improvement of the mean error for this vertex after we impose our per-vertex loss for training. For this evaluation, we used all test examples of UP-3D. From the visualization, we infer that the per-vertex loss has benefited mostly the extremities of the human body (arms, legs, head), and in particular the lower arms, where the decrease of the mean error can even exceed the 50mm.

8. Architecture

The individual components of our architecture (i.e., *Human2D*, *PosePrior*, *ShapePrior*), are designed following best practices from the literature, as we described in the main manuscript. For completeness, here we present all the details concerning the exact architectures used in our experiments. In Figure 2, we present the basic architecture for

	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SitingD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Akhter & Black [1]*	199.2	177.6	161.8	197.8	176.2	186.5	195.4	167.3	160.7	173.7	177.8	181.9	176.2	198.6	192.7	181.1
Ramakrishna *et al*. [8]*	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6	175.6	160.4	161.7	150.0	174.8	150.2	157.3
Zhou *et al*. [12]*	99.7	95.8	87.9	116.8	108.3	107.3	93.5	95.3	109.1	137.5	106.0	102.2	106.5	110.4	115.2	106.7
Bogo *et al*. [2]	62.0	**60.2**	**67.8**	76.5	92.1	**77.0**	73.0	75.3	100.3	137.3	83.4	77.3	86.8	79.7	87.7	82.3
Lassner *et al*. [5] (direct)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	93.9
Lassner *et al*. [5] (optimization)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	80.7
Ours	**59.3**	61.8	70.6	**68.0**	**91.0**	80.1	**59.7**	**64.2**	**89.1**	**124.4**	**77.3**	**68.8**	**73.0**	**70.5**	**71.6**	**75.9**

Table 4: Detailed results on Human3.6M [3]. Numbers are reconstruction errors (mm). The numbers are taken from the respective papers, except for (*), which were obtained from [2].

	FB Seg.		Part Seg.	
	acc.	f1	acc.	f1
SMPLify on GT	92.17	88.23	88.82	67.03
SMPLify	91.89	88.07	87.71	63.98
SMPLify + Lassner *et al*. anchor [5]	92.13	88.35	88.18	64.54
SMPLify + our anchor	**92.17**	**88.38**	**88.24**	**64.62**
Lassner *et al*. [5] anchor	86.66	79.93	82.32	51.02
Our anchor	**89.51**	**84.57**	**84.68**	**55.47**

Table 5: Accuracy and f1 scores for foreground-background and six-part segmentation on LSP test set for different versions of SMPLify, the direct prediction of Lassner *et al*. [5], and our direct prediction. The numbers for the first, second and fifth rows are taken from [5]. Our approach performs better than the direct prediction method of Lassner *et al*. [5] and leads to faster convergence of the iterative optimization when the two are employed as anchors for SMPLify.

	PCKh
Human2D	94.6
3D joints projection	89.5

Table 6: Evaluation of 2D keypoint localization on the test set of UP-3D. The numbers are PCKh scores. Since the *Human2D* network has been explicitly trained for the keypoint detection, it performs better on this task, compared to projecting the 3D joint predictions on the 2D plane.

the hourglass module [7], while in Figure 3 we present *Human2D*, an adaptation of the stacked hourglass network, that predicts both heatmaps for the joints and silhouette segmentation. Regarding the *Priors* networks, Figure 4 illustrates our *PosePrior* network, which was used for the prediction of the pose parameters and includes two bilinear modules [6]. The *ShapePrior* network for shape parameter estimation is presented in Figure 5 and consists of five convolutional layers and one bilinear module in the end.

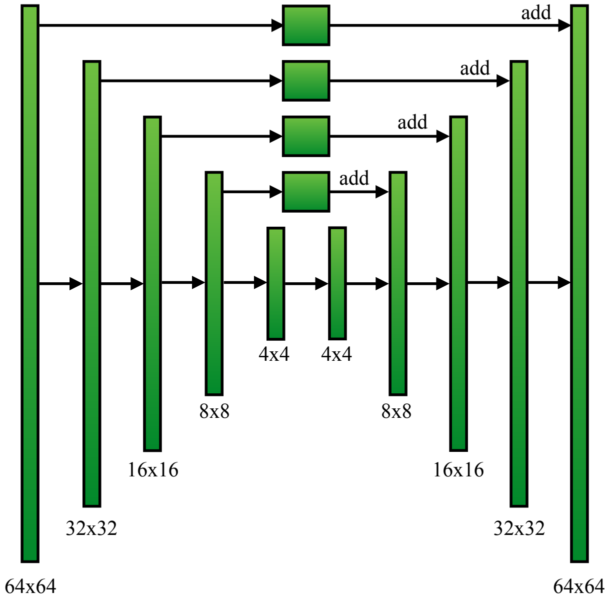

Figure 2: Schematic representation of the hourglass component. Each column corresponds to three consecutive residual modules. The convolutions are implemented with 3×3 kernels, and the number of channels remains equal to 256 across the hourglass. ReLU is the activation function, while batch normalization is also used. The numbers at the bottom of each column indicate the spatial resolution of each level of the hourglass. At the encoding part of the hourglass, max pooling is used to decrease the resolution, while at the decoding part, nearest neighbor upsampling is used to increase the resolution. The skip connections also contain residual modules, and their output is added element-wise to the feature map of the decoding part after the nearest neighbor upsampling procedure.

9. Implementation details

Regarding the transition from the *Human2D* network to the *PosePrior* component, we need to transform the N heatmaps (where $N = 16$) to the input vector of $3N$ values. To get the pixel position of the joints, we use an *argsoftmax* operation [11] on the heatmaps. Then, we shift the joint coordinates, such that the root joint (pelvis) is on the location

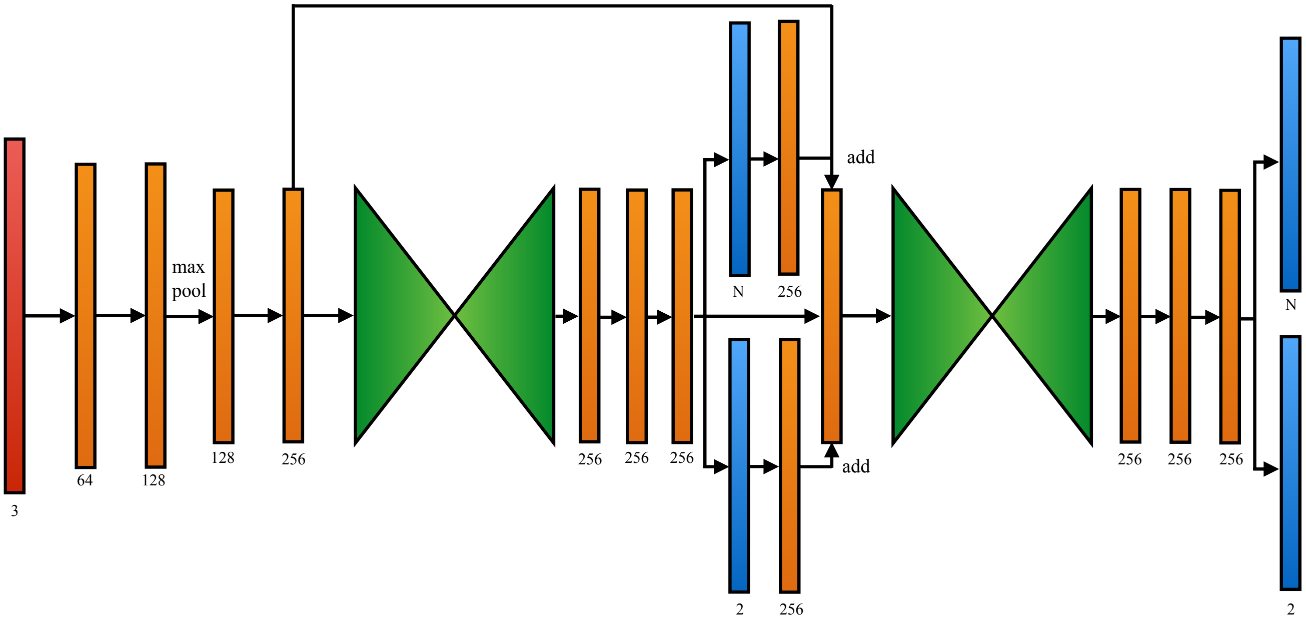

Figure 3: The complete architecture for *Human2D*, including the hourglass components. The orange columns are convolutional layers, the red column corresponds to the input image, while the blue columns indicate the outputs (intermediate and final) for heatmaps (corresponding to the N joints) and silhouettes (body and background channels). The green modules correspond to the hourglass design from Figure 2. The numbers at the bottom of the columns indicate the number of channels for each feature map. ReLU is the activation function, while batch normalization is also used. All layers are implemented as residual modules with kernels of size 3×3. The only exception is the first layer, which is a 7×7 convolution, and the layers that produce and post-process the outputs, which implement 1×1 convolutions. The spatial resolution starts at 256×256, drops at 128×128 after the first layer (which uses stride equal to two), and then drops again to 64×64 after the second module with a max pooling operation. This resolution remains constant until the end of the network (with the exception of the interiors of the hourglasses).

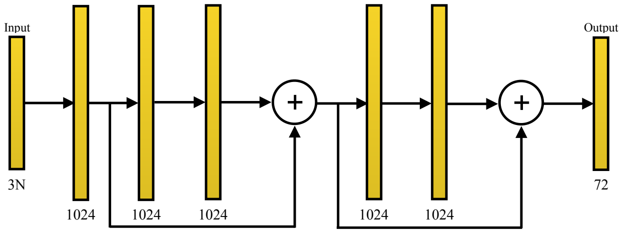

Figure 4: Detailed architecture for the *PosePrior* component. The input is a vector of size $3N$, containing the coordinates of the 2D joint locations and the confidences of the detections (realized by the maximum values of the heatmaps). Then a fully connected layer brings the dimensionality to 1024. After that, two bilinear modules [6] follow. The architecture of each bilinear module includes two fully connected (linear) layers of size 1024 with a skip residual connection from the input to the output of the module. Finally, an additional fully connected layer brings the dimensionality to 72, which is the output of the *PosePrior* component. After each fully connected layer of size 1024, we use batch normalization, ReLU, and Dropout.

$(0, 0)$, and we scale them by dividing with the $\max$ absolute value across all the x-y coordinates, such that the coordinates are in the $[-1, 1]$ interval. The $\max$ value of each heatmap (which realizes an evidence for the confidence of the detection) is also concatenated to these $2N$ values.

Concerning the differentiable rendering, we use a perspective projection model. For the supervision based on 2D annotations from in-the-wild images (described in Section 4.3 of the main manuscript), the focal length is typically not known and it is hard to be precisely estimated. For our implementation, we use a standard value $f = 5000$. During training, for a predicted 3D shape, we use this value to estimate the global translation vector (camera extrinsics) which asserts that our projected 3D shape will have the same vertical extend as the annotated silhouette. Provided with these parameters (focal length, global translation, 3D shape), the renderer can handle the rest of the projection procedure.

10. Training details

For the training of the *Priors* networks, it is crucial to augment the training with noisy inputs, to anticipate noisy 2D predictions from the *Human2D* network. This is a form of data augmentation, which is typically used in the form

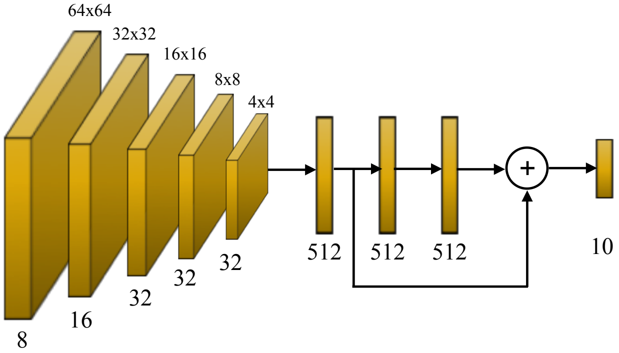

Figure 5: Detailed architecture of the *ShapePrior* component. Initially, the body segmentation is filtered with five convolutional layers which are implemented as residual modules. The kernels are of size 3×3. The activation is ReLU, while batch normalization is also used. Max pooling is used after every module to reduce the spatial resolution. The numbers at the bottom of each module indicate the number of channels for the feature maps, while the numbers at the top indicate the spatial resolution. After the last convolutional layer, a fully connected layer brings the dimensionality to 512. Consequently, one bilinear module [6] follows. The bilinear module includes two fully connected layers with size 512 and a skip connection from the input to the output. A final fully connected layer brings the dimensionality to 10, which is the output of the *ShapePrior* component. After each fully connected layer of size 512, we use batch normalization, ReLU, and Dropout.

of pixel-wise noise when training CovNets with images as input. For the *PosePrior* component we add noise to each coordinate (after centering and rescaling in $[-1, 1]$), sampling from a Gaussian with $\mu = 0$ and $\sigma = 0.05$. Given the distance of each noisy keypoint from its original position, we scale the input confidence value as well (where we assign $conf = 1$ if the distance is 0, $conf = 0$ if the distance is 0.4 or greater, and we use linear scaling to assign confidences for distance values within this interval). For the *ShapePrior* network we add the more traditional pixelwise noise to the input silhouette.

11. Qualitative evaluation

Because of space constraints, the main manuscript included only a small sample of reconstruction examples for our approach. Here we provide a more detailed qualitative evaluation on the various datasets. Figure 6 collects successful reconstructions of our approach on UP-3D [5] (effectively extending Figure 3 of the main manuscript). In Figure 7, we compare the results of our approach with output meshes from the direct prediction approach of Lassner *et al.* [5] on the same dataset (extending Figure 4 of the main manuscript). Concluding the evaluation on UP-3D, in Figure 8 we present some erroneous reconstructions made by our network, which summarize the failure modes of our approach. Regarding the Human3.6M part of SURREAL [10], we have collected a variety of outputs in Figure 9. The first seven rows consist of successful reconstructions, while the last row includes some error cases. In general, the low lighting and the challenging backgrounds are the cause of the most common failures for this dataset. For Human3.6M [3], we have collected in Figure 10 a subset of the failure cases for the iterative optimization approach of Bogo *et al.* [2] and compare it with our predictions for the same input images. In the majority of these examples, the projected 2D pose for [2] is correct, so the errors occur at the reconstruction step (optimization). In contrast, our direct approach which is discriminatively trained, predicts more faithful reconstructions, which even if they do not match exactly with the image evidence, are in accordance with the 3D pose of the person. Finally, in Figure 11, we provide additional examples for the scenario (Section 5.5 of the main manuscript) that our direct predictions are used as an initialization and an anchor for the iterative optimization approach of Bogo *et al.* [2] (SMPLify), leading to more accurate pose and shape reconstructions (extending Figure 5 of the main manuscript).

References

[1] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *CVPR*, 2015. 3

[2] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016. 2, 3, 5, 8

[3] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014. 2, 3, 5, 8

[4] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 8

[5] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *CVPR*, 2017. 2, 3, 5, 6

[6] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3D human pose estimation. In *ICCV*, 2017. 3, 4, 5

[7] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 3

[8] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3D human pose from 2D image landmarks. In *ECCV*, 2012. 3

[9] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. In *ICCV*, 2017. 2

[10] G. Varol, J. Romero, X. Martin, N. Mahmood, M. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *CVPR*, 2017. 1, 2, 5, 7

[11] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016. 3

[12] X. Zhou, M. Zhu, S. Leonardos, and K. Daniilidis. Sparse representation for 3D shape estimation: A convex relaxation approach. *PAMI*, 2016. 3

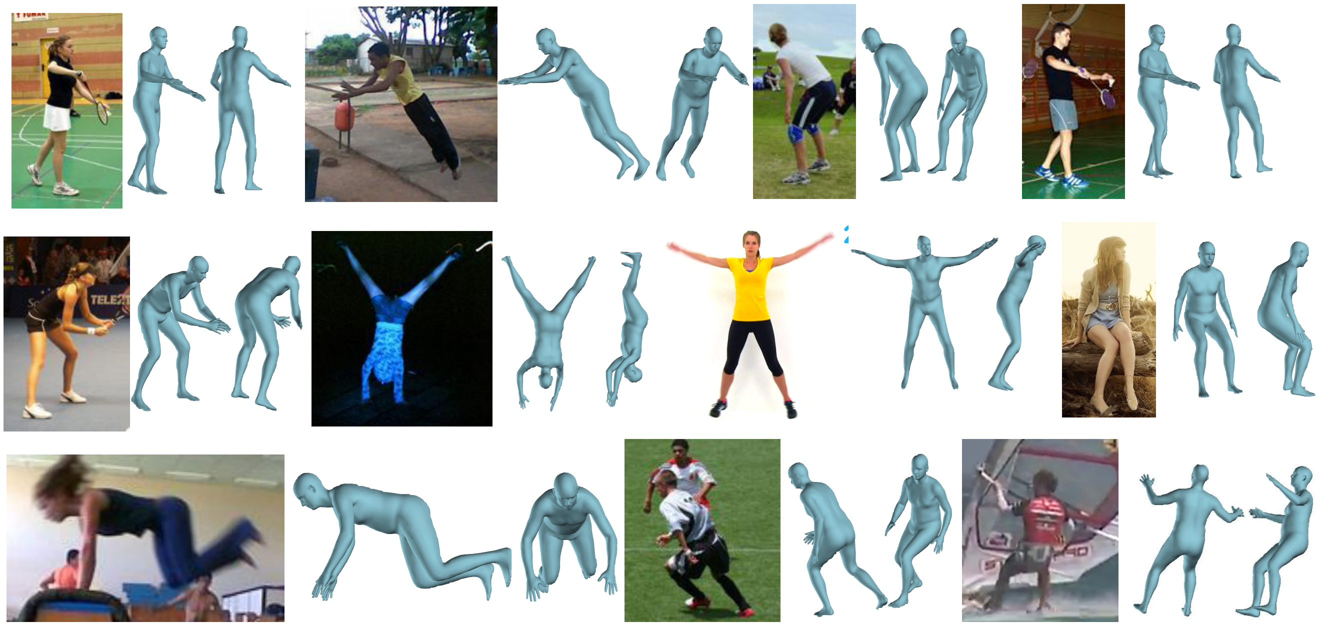

Figure 6: Successful reconstructions of our aproach on UP-3D [5] (extension of Figure 3 of the main manuscript).

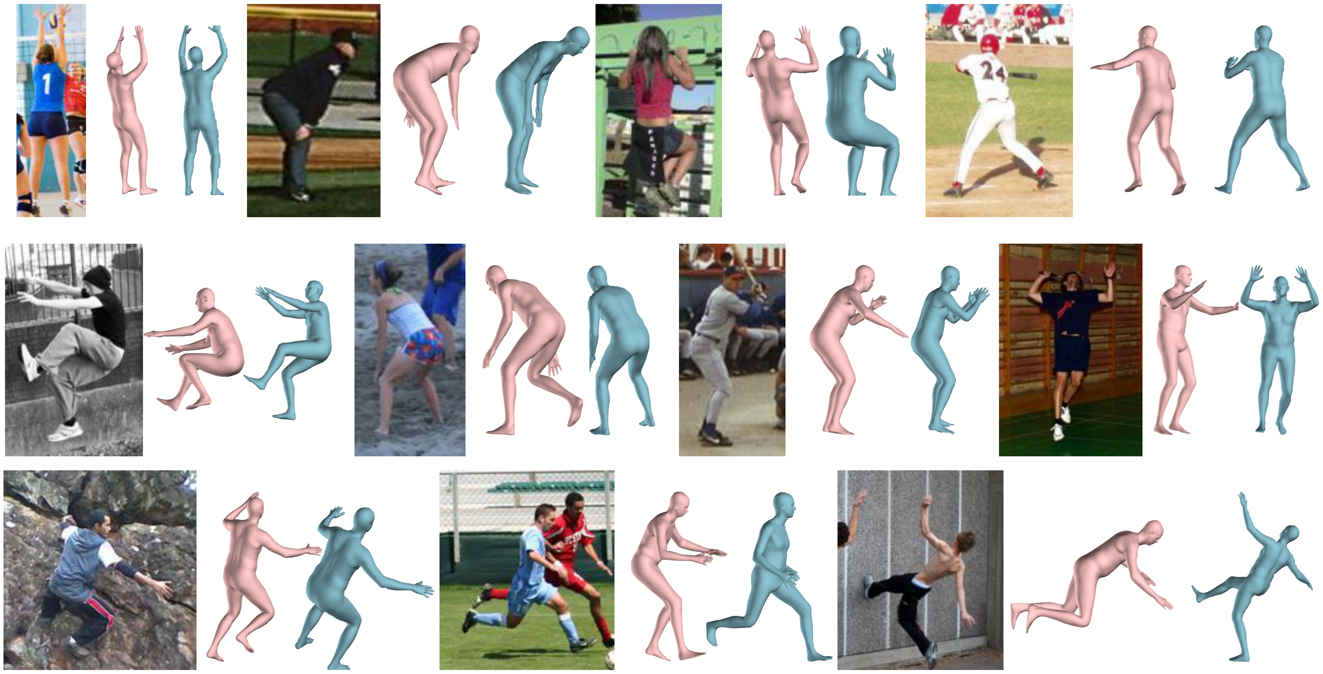

Figure 7: Reconstructions on UP-3D [5] using our approach (blue) and the direct prediction approach of [5] (pink). Our approach typically leads to more accurate and faithful reconstructions (extension of Figure 4 of the main manuscript).

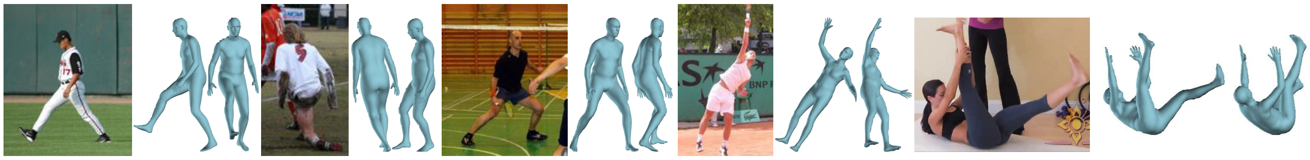

Figure 8: A set of typical failure cases on UP-3D [5] for our approach. Reconstructions that do not match exactly to the 2D image evidence, retrieval of a more regular pose, ignoring small scale details, errors in the global rotation, and failures because of particularly challenging poses (e.g., with self-occlusions), are the main sources of error for our approach.

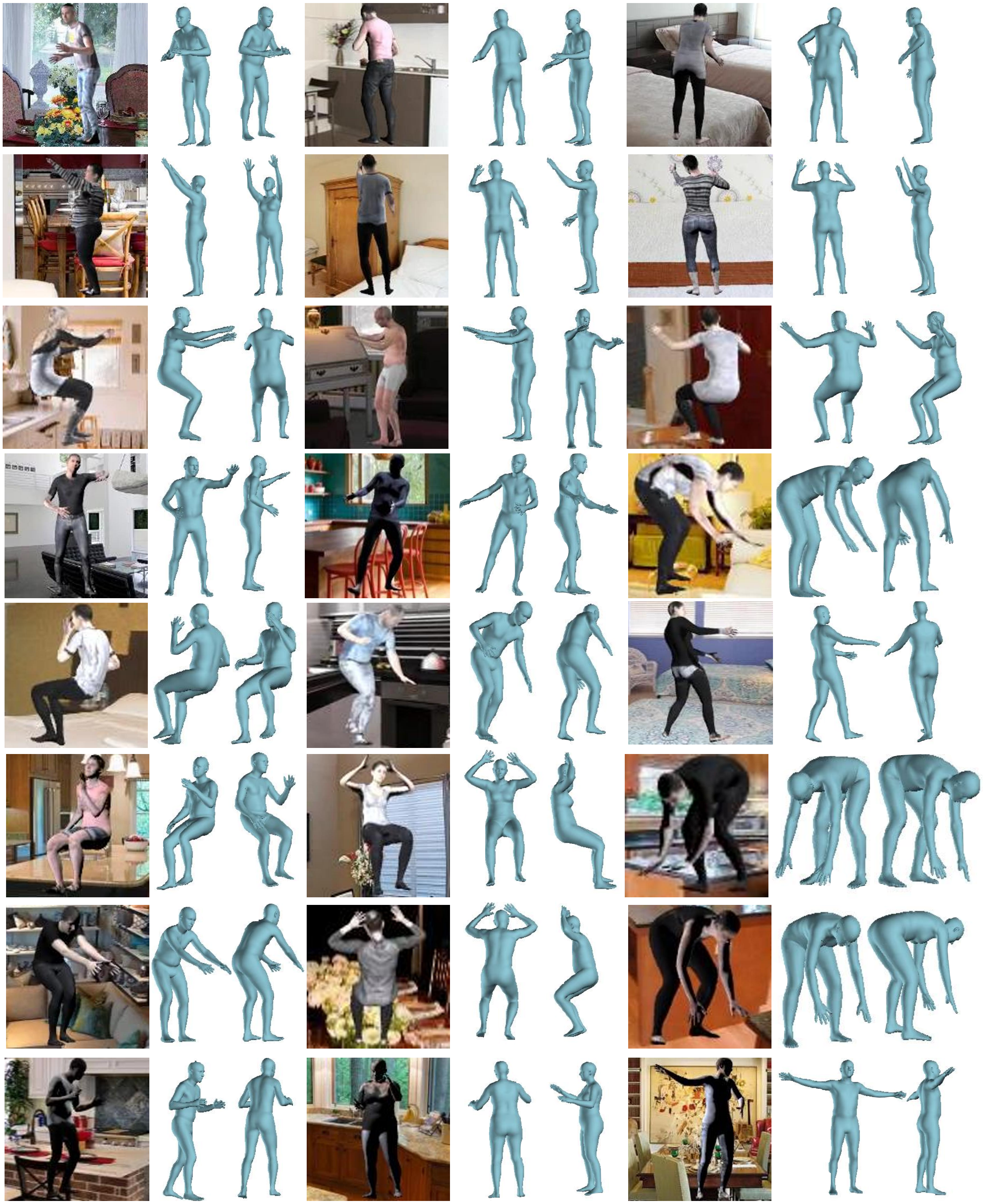

Figure 9: Reconstructions of our approach on the Human3.6M part of SURREAL [10]. The last row corresponds to failure cases, which are usually attributed to the low lighting conditions and challenging backgrounds.

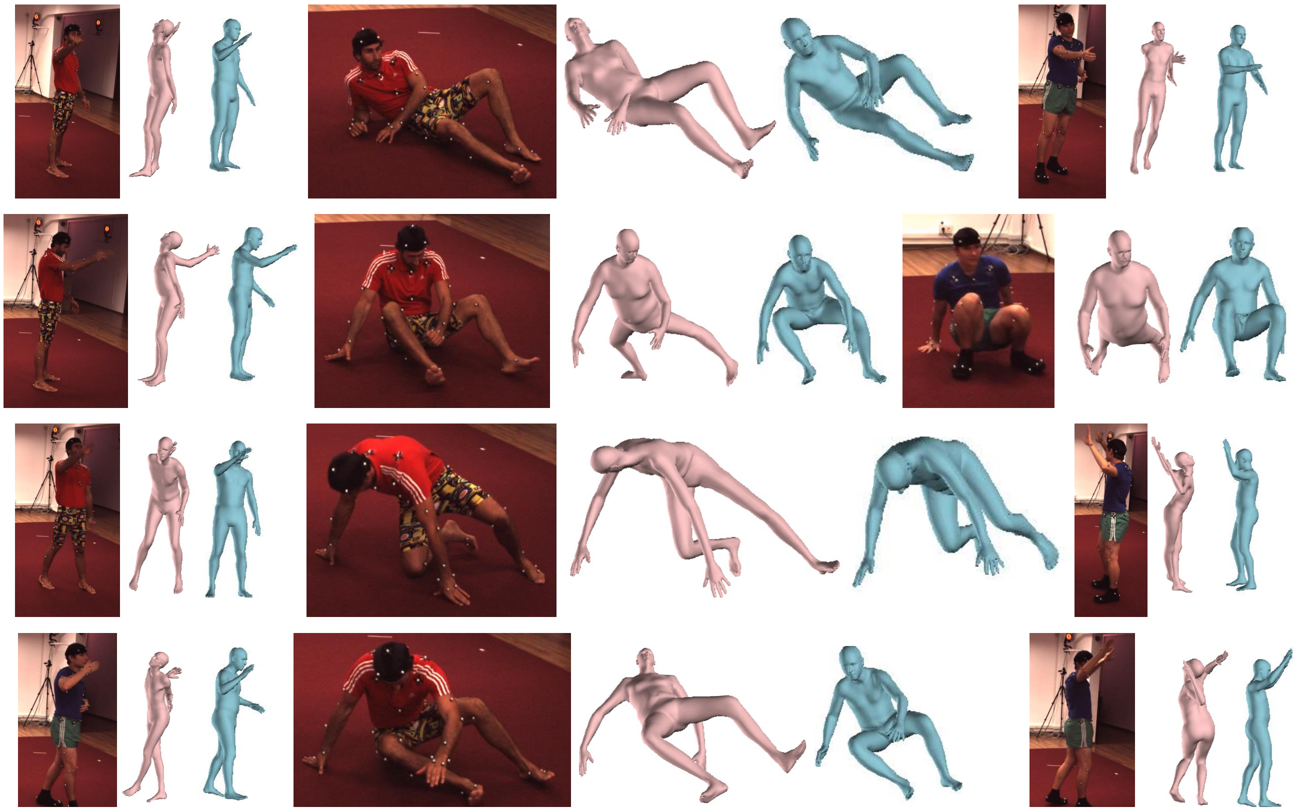

Figure 10: Reconstructions on Human3.6M [3] for our approach (blue) and the iterative optimization approach of Bogo *et al*. [2] (pink). The optimization solution relies heavily on the 2D detections. Even small errors for the detected 2D locations can lead to erroneous reconstructions (including flipping, interpenetration, failures because of depth ambiguity, etc). In these cases our approach typically provides more reasonable results, even if they are not completely faithful to the image evidence.

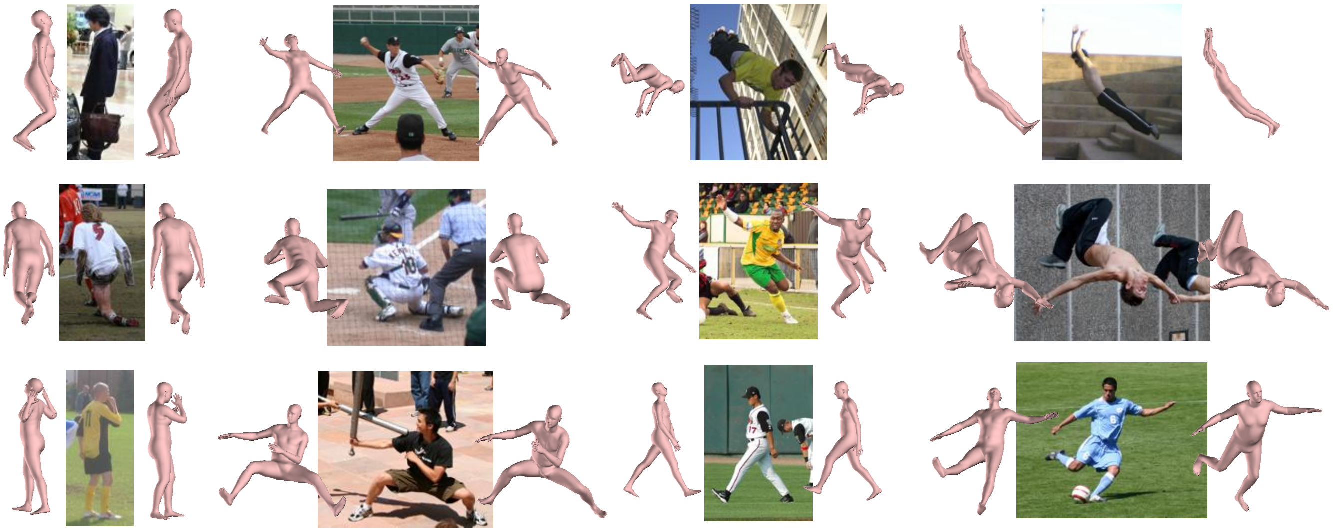

Figure 11: Reconstruction on the test set of LSP [4] using vanilla SMPLify [2] (left of each image), and our anchored version of the same algorithm (right of each image). Using our direct prediction as initialization and as anchor can help to get reasonable 3D reconstructions and avoid unsatisfying failures (extension of Figure 5 of the main manuscript).