Delving Deep into Pixel Alignment Feature for Accurate Multi-view Human Mesh Recovery (Supplementary Document)

Kai Jia, Hongwen Zhang*, Liang An, Yebin Liu*

[1]Association for the Advancement of Artificial Intelligence
Department of Automation, Tsinghua University, Beijing, China
kajia@umich.edu, zhanghongwen@mail.tsinghua.edu.cn, al17@mails.tsinghua.edu.cn, liuyebin@mail.tsinghua.edu.cn

More Implementation Details

Pytorch (Paszke et al. 2019) is used to implement our Pixel-aligned Feedback Fusion (PaFF) model. The training is performed in a single Quadro RTX 8000 GPU. The model runs at 15fps for four input views.

Pixel Alignment Feedback (PaF) Feature Extractor We build the Pixel Alignment Feedback (PaF) feature extractor upon ResNet-50 (He et al. 2016), which produces image features with 7×7 resolution. We then upsample the features with deconvolution layers to construct a feature pyramid with three feature maps, the resolution of which are 14×14, 28×28, and 56×56. The grid sampling utilizes 21×21 sampling points, and the mesh vertex sampling downsamples the SMPL mesh vertices to a number of 431 following Kolotouros, Pavlakos, and Daniilidis (2019). After the sampling, 1D convolutional layers with LeakyReLU activation are used to downsize the sampled features to 2155 (431 x 5). By appending the decoder structures (the same as HMR (Kanazawa et al. 2018), except for not using dropout operations to maintain temporal stability) to the PaF feature extractor, we pre-train the feature extractor on the monocular datasets, as mentioned in Sect. 'Training'.

Calibration-free PaFF Achitecture To construct a calibration-free PaFF (Calib-free PaFF), we apply individual regression networks as the decoders to predict the orientation and translation. The regression networks are similar to HMR (Kanazawa et al. 2018) except that the dropout layers are removed due to their bad effect on stability for temporal prediction. The temporal stability of our model is shown in the supplementary video. Note that the focal length used in calibration-free PaFF is a default value of 5000. to adapt to different cameras while the PaFF with calibration uses real focal lengths of cameras.

Multi-view Feedback Fusion Module All of the feedback fusion tasks in the paper use the same PaF feature. In the Multi-view Pose & Shape Feedback Fusion module, the PaF feature is mapped to a feature dimension of 1024 using fully connected layers. In the Global Orientation Estimator, the PaF feature is mapped to 64 before being fed into the transformer fusion module. In the Global Translation Estimator, the PaF feature is mapped to 1024 and goes through an individual fullly-connected layer regressor same as the Calib-free PaFF.

*Corresponding Authors

More Training Details

As mentioned in Sect. 'Training', there are two stages of training. The first stage is to pre-train the Pixel-alignment Feedback Extraction Model. The pre-training setting is the same with (Zhang et al. 2021) except for our structure using 4 iterations instead of 3 iterations for the regression task and our decoder structure does not contain dropout, which is the reason for doing the pre-training. Note that the goal of the pre-training is not to get an accurate monocular estimator but to train the feature encoder to extract meaningful image features for multi-view estimation. For the second stage for training multi-view fusion modules, we fix the Pixel-alignment Feedback Extraction Model since it is trained with more diverse human instances compared with multi-view datasets, in which the extracted feature can generalize well.

For the first stage, we use the same kinds of losses in (Zhang et al. 2021), which are supervisions from ground truth 2D keypoints, 3D keypoints, SMPL human body parameters (10 shape and 24 pose (including root joint which is the body orientation)), auxiliary pixel-wise supervision given by UV map, and regularization of camera estimated body scale. For the second stage, we did not train the feature encoder. So, the auxiliary pixel-wise supervision is removed from training losses. And in the training of one dataset - MTC (Xiang, Joo, and Sheikh 2019), we find the model would converge to a thin-long shape even with mix-training in Human 3.6M. The reason is the limited data for supervising shape, which we have mentioned in Sect. Conclusions. We utilize shape regularization to deal with abnormal shape prediction, which is an L1 Norm to penalize large shape components. In all, the losses used in the second stage are given in Equ. 1.

$$\begin{aligned}\mathcal{L}_{second} = &\lambda_{2d}||K_{2d} - \hat{K_{2d}}|| + \lambda_{3d}||K_{3d} - \hat{K_{3d}}|| \\ &+ \lambda_{pose}||\theta - \hat{\theta}|| + \lambda_{shape}||\beta - \hat{\beta}|| \\ &+ \lambda_{shape-regu}|\beta| + \lambda_{scale}Exp\_Term(s),\end{aligned} \tag{1}$$

Algorithm 1: Global Orientation Aligner Algorithm

Input: $\{O_t^v, R_{cam}^v, v = 0, 1, ..., N\}$

$O_g^{pre} = Mean(\{(R_{cam}^v)^T O_t^v, v = 0, 1, ..., N\})$

$v_f = \text{argmax}_v \, abs((R_{cam}^v)^T O_t^v - O_g^{pre})$

$O_g = Mean(\{(R_{cam}^v)^T O_t^v, v = 0, 1, ..., N, v \neq v_f\})$

$\widehat{O_t^v} = (R_{cam}^v) O_g$

Output: $\{\widehat{O_t^v}, v = 0, 1, ..., N\}, O_g$

where $||\cdot||$ is the squared L2 norm and $|\cdot|$ is the L1 norm, $\hat{K_{2d}}$, $\hat{K_{3d}}$, $\hat{\theta}$, $\hat{\beta}$ denote the ground truth 2D keypoints, 3D joints, and SMPL pose and shape parameters, respectively. s is the estimated body scale with the orthogonal camera. The Exp_Term is to prevent it from becoming zero by s exponential function $(e^{-x})^2$. Note that the orientation estimation O^v is contained in the parameter θ but needs to be treated independently for each view image (we use a simplified formula here).

Derivations

Here, we present the mathematical derivations of the algorithms mentioned in Sect. 'Global Orientation, Translation Estimator'.

Global Orientation Aligner Algorithm

The Global Orientation Aligner Algorithm is used to align the relative orientation estimations of human in multi-view cameras, as illustrated in Alg. 1. Specifically, given the camera rotation parameters $\{R^v, v = 0, 1, ..., N\}$ and the relative orientation estimations $\{O_t^v, v = 0, 1, ..., N\}$ for one iteration t, we first estimate a global orientation O_g^{pre} from the mean of global orientation estimations for each view. Then by using O_g^{pre}, we filter out one most skewed orientation estimation and update the global orientation estimation as O_g. Finally, we update the relative orientation estimation $\widehat{O_t^v}$ for each view from O_g.

Global Translation Estimator Algorithm

As mentioned in Sect. 'Global Orientation, Translation Estimator', estimating global translation using camera parameters is helpful for removing scale ambiguity. Specifically, we apply the Global Translation Estimator Algorithm to estimate the global translation by assuming the estimated human body is aligned after independent orthogonal camera prediction. We give the derivation for the algorithm here.

Specifically, we first estimate global translation for the pelvis of the human body Tp_{global}. By assuming the estimated body is aligned for each view, the estimated pelvis 2D location and the global location of the 3D pelvis lie in the same camera ray for each view. Further, by linking the relative translation estimation with a global estimation using camera parameters, we solve an adaptive scale to remove the scale ambiguity. The derivations are shown below. Note that only the 3D/2D locations of the estimated pelvis are used, which are regressed from the estimated SMPL vertices (Loper et al. 2015) using pre-trained joints regressor.

The global position of pelvis T_{global} can be projected to the image in 2D location P^v using camera parameters $K_{cam}^v, R = R_{cam}^v, T_{cam}^v, v = 1, .., N$ and local image transformation parameter K_{cs}^v, as shown in Equ. 2.

$$
\begin{aligned}
&P^v = \prod(T_{global}, K = K_{cs}^v K_{cam}^v, R = R_{cam}^v, T = T_{cam}^v) \\
&T_{global}^v = [\ X \quad Y \quad Z\ ] \\
&K_{cs}^v = \begin{bmatrix} f_{cs}^v & 0 & Cs_x^v \\ 0 & f_{cs}^v & Cs_y^v \\ 0 & 0 & 1 \end{bmatrix}, \\
&K_{cam}^v = \begin{bmatrix} f_x^v & 0 & C_x^v \\ 0 & f_y^v & C_y^v \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2}
$$

where $\prod$ is the perspective projection operation, K_{cs}^v is used to transform the original image to a local image given by the bounding box of the human body, which can be seen as an additional intrinsic transformation.

The translation prediction network predicts an orthogonal camera with a relative 2D translation $t_o \in \mathbb{R}^2$ and scale $Scale_o \in \mathbb{R}$, which is the same with HMR (Kanazawa et al. 2018). The projection of a 3D point X is $\hat{x}$ for that representation in Equ. 3.

$$
\hat{x} = Scale_o \times \Pi(X) + t_o, \tag{3}
$$

where Π is an orthographic projection. The reason for using the orthogonal camera is that the 2D image feature has scale ambiguity and is easier to adapt the orthogonal camera.

A relative translation TO^v for each view can be inferred from t_o^v and $Scale_o^v$ in Equ 4.

$$
\begin{aligned}
&X_o^v, Y_o^v = t_o^v[0], t_o^v[1] \\
&Z_o^v = f_{net}/(H_{res}/2 \times Scale_o^v) \\
&TO^v = [\ X_o^v \quad Y_o^v \quad Z_o^v\ ],
\end{aligned}
\tag{4}
$$

where f_{net} is the focal length for the regression task and H_{res} is the resolution of input images. When it is Calib-free PaFF, $f_{net} = 5000$.. When it is PaFF with calibrated cameras, f_{net} becomes view-dependent and $f_{net}^v = f_{cs}^v \times mean(f_x^v, f_y^v)$. The reason for not using default focal length here is that it is easier for the network to converge to a good TO^v with the real focal length given by the multiply of K_{cam}^v and K_{cs}^v.

$$
f_{net}^v = f_{cs}^v \times mean(f_x^v, f_y^v) \tag{5}
$$

Then the relative position of pelvis TO^v for view v can be projected to the image as a 2D location PO^v in Equ. 6.

$$
\begin{aligned}
&PO^v = \prod(TO^v, K = K_{net}, R = I, T = O) \\
&K_{net}^v = \begin{bmatrix} f_{net} & 0 & H_{res}/2 \\ 0 & f_{net} & H_{res}/2 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{6}
$$

where K_{net} is the intrinsic matrix used in the projection of the neural network model. I is an identity matrix, and O is a zero translation matrix.

Since the estimated scales of the human body are the same given by one unique shape parameter across multi-view, we apply a Global Translation Estimator Algorithm 2 to solve an adaptive scale $scale_g$ and a global translation of the 3D pelvis T_{global} in the first stage and solve an updated estimated relative translation $\widehat{TO^v}$ with updated scale $\widehat{scale_o^v}$ for each view in the second stage.

After performing extrinsic transformation as Equ. 7, the global translation of pelvis T_{global} is transformed to a relative translation T^v_{global} for each view. Since the scale of the body is consistent among different views, we have $Z^v = scale_g \times Z^v_o$ (the body might be farther or closer scaled with the adaptive scale). By assuming 2D pelvis is aligned, we have $PO^v = P^v$. With these equations, we transfer the estimation of $scale_g$ and T_{global} to solving a linear equation in Equ. 10.

$$\begin{aligned} T^v_{global} &= [\ X^v \quad Y^v \quad Z^v\ ] \\ &= \Gamma(T_{global}, R = R^v_{cam}, T = T^v_{cam}) \end{aligned} \tag{7}$$

where Γ is the extrinsic transformation operation.

$$P^v_{2D} = PO^v_{2D} \tag{8}$$

$$Z^v = scale_g \times Z^v_o \tag{9}$$

$$\begin{aligned} &T^v_{cam} = [\ I \quad -R^v_{cam}T^v_o\ ] \times [\ T_{global} \quad scale'_g\ ]^T, v = 0, ..., N \\ &T^v_o = \begin{bmatrix} f_{net}/(f^v_{cs}fx^v)X^v_o + P^v_x Z^v_o \\ f_{net}/(f^v_{cs}fy^v)Y^v_o + P^v_y Z^v_o \\ Z^v_o \end{bmatrix} \\ &P^v_x = (H_{res}/2 - Cs^v_x - f^v_{cs}C^v_x)/(f^v_{cs}fx^v), \\ &P^v_y = (H_{res}/2 - Cs^v_y - f^v_{cs}C^v_y)/(f^v_{cs}fy^v) \\ &scale_g = f_{net}/(f^v_{cs}fx^v) \times scale'_g \end{aligned} \tag{10}$$

Note that f_{net} is not expanded as Equ. 5. A default f_{net} can also generalize in the solution.

After estimating T_{global} and $scale_g$, in the second stage of Alg. 2, we derive an updated relative translation $\widehat{TO^v}$ and thus an updated $\{\widehat{t^v_o}, \widehat{scale^v_o}\}$ for each view using $PO^v = P^v$ (updating the 2D Pelvis location in each view) and Equ. 4. In the second stage, there is a scale assumption that the depth Z^v reflects the real depth of the human body but with an adaption for the assumed focal length f_{net} and $scale_g$. The multiply of $scale_g$ is to keep the continuity for the vertices projection (not to move the body closer or farther).

The updated $\{\widehat{t^v_o}, \widehat{scale^v_o}\}$ can be passed to the next iteration to correct the estimation of single view translations after combining the multi-view translation estimations. However, we found that the expected improvement with the better translation alignment does not happen in MPJPE, PA-MPJPE, and PVE, which might be due to the discontinuity induced by the updated $\{\widehat{t^v_o}, \widehat{scale^v_o}\}$. It will be studied in one additional ablation study.

Algorithm 2: Global Translation Estimator Algorithm

Input: $\{PO^v, K^v_{cs/cam}, K^v_{cam}, , K_{net}, R^v_{cam}, T^v_{cam}, TO^v, v = 0, ..., N\}$

Stage 1: Solve T_{global} **and** $Scale_g$

1: Assumption: $PO^v = P^v$, $Z^v = scale_g \times Z^v_o$

2: Solve T_{global}, $scale_g$ from a Linear Equation 10

***Stage 2*: Solve** $\{\widehat{TO^v}, \widehat{t^v_o}, \widehat{scale^v_o}, v = 0, 1, ..., N\}$

3: Derive $\{T^v, v = 0, 1, ..., N\}$ from T_{global}

4: Scale Assumption: $f_{net} \times Z^v = f^v_{cs}fx^v \times \widehat{Z^v_o} \times Scale_g$

5: Derive $\{\widehat{TO^v}, \widehat{t^v_o}, \widehat{scale^v_o}, v = 0, 1, ..., N\}$ from $\{T^v, v = 0, 1, ..., N\}$ using Equ. 8 and Equ. 4

Output: $T_{global}, scale_g, \{\widehat{TO^v}, \widehat{t^v_o}, \widehat{scale^v_o}, v = 0, 1, ..., N\}$

More Experiment Details

Datasets

Here, we provide additional details for the datasets we use for training and evaluation.

To pre-train the PaF feature extractor, we adopt monocular datasets - COCO (Lin et al. 2014), MPII (Andriluka et al. 2014), LSP (Johnson and Everingham 2010), and LSP Extended (Andriluka et al. 2014). Here, we give more details for those datasets.

COCO COCO (Lin et al. 2014) is a large-scale dataset with human keypoints annotations. Following Zhang et al. (2021), we only use the samples with at least 12 keypoints for training.

MPII MPII (Andriluka et al. 2014) is a dataset with diverse human keypoints annotations, which are collected from Youtube. We only use the samples with the complete keypoints annotation for training.

LSP and LSP-Extended LSP (Johnson and Everingham 2010) and LSP Extended (Andriluka et al. 2014) are two 2D human pose estimation datasets, which comprise 2D human keypoints coming from sports scenes. We keep the samples with at least 14 keypoints annotations for training.

MTC MTC (Xiang, Joo, and Sheikh 2019) is a dataset captured by Panoptic Studio (Joo et al. 2015), which captures 40 diverse subjects in multi-view 31 cameras. It contains annotations for the whole body. We use it to demonstrate our PaFF's generalization ability for different camera views by mix-training it with Human3.6M using the 3D keypoints ground truth and the projected 2D keypoints ground truth. See the results in the supplementary video.

Evaluation Metrics

During the evaluation on Human3.6M (Ionescu et al. 2013), we adopt MPJPE, PA-MPJPE, and PVE as the metrics. MPJPE is the Mean Per Joint Position Error which reflects the mean absolute location error for the joints prediction. PA-MPJPE is MPJPE after rigid alignment between the prediction and the ground truth using Procrustes Analysis (PA), which can reflect the accuracy of the relative position of joints. PVE is the mean Per-vertex Error, which is defined as

Iter Number	MPJPE	PA-MPJPE	PVE	O Err
0	42.9	31.7	64.3	6.4°
1	36.6	28.8	52.8	5.6°
2	34.2	27.7	50.3	5.4°
3 (Ours)	**33.0**	**26.9**	**48.9**	**5.1**$^\circ$

Table 1: Iterative Performance of PaFF on Human 3.6M. '0' '3' denotes the feedback fusion iteration numbers in PaFF. It is clear that the performance is improved from iteration to iteration. O Err is in degrees

the average Euclidean distance between the estimated mesh vertices and the ground truth vertices. PVE can reflect the shape estimation accuracy of human bodies.

During the evaluation on MPI-INF-3DHP (Mehta et al. 2017), another two metrics, PCK and AUC, are employed (the same as (Liang and Lin 2019)). PCK denotes 'Percentage of Correct Keypoints' with a threshold of 150mm, which gives the percentage of 'good' estimated keypoints within an error threshold. AUC denotes 'Area Under the Curve' with a threshold range of 0-150mm.

For evaluating rotation estimation the Global Orientation Estimator, we adopt a simple rotation angle error named after 'O Err' in the main text, which is the angle by which the predicted body orientation needs to be rotated to get to the ground truth orientation (check Equ. 11).

$$OErr = \arccos((Tr(O_{pred}^{T} O_{gt}) - 1)/2) \qquad (11)$$

Tr is to calculate the trace of the matrix. We also evaluate the quality of global translation by comparing with ground truth global translation given by triangulation of ground truth Pelvis 2D locations in images, which is denoted as 'T Err' in 'mm'.

More Ablation Studies

The additional ablation studies are to answer the following questions: (i) What is the benefit of doing iterative refinement? (ii) The additional analysis related to O,T estimations (Whether to do initialization from single-view estimators? How it will influence if pass the updated orthogonal camera projection with Global Translation Estimator?). (iii) Can PaFF generalize to 2/3 views multi-view estimations?

Feedback Iteration Numbers In PaFF, except for the first initial iteration, there are 3 iterations for refining the estimation with feedback fusion. We test three additional options, '0', '1' and '2', by removing the first 3, 2, or 1 iterations (all options would utilize the largest feature map). Table 1 shows that 3 iterations (our choice) perform better than other options. This indicates that more feedback fusion iterations are helpful yet less feedback fusion iterations could still have a good performance, demonstrating the efficiency of feedback fusion itself. Besides, the decressing O angular errors from '0' to '1' also show the benefit of the multi-step refinement process of the body orientation estimation.

Additional O/T Settings We initialize the body orientation O^v and translation for each view TO^v using the single view estimator - the Pixel Alignment Feedback Feature Extraction Model to align the O, and T estimations faster then make the rest local joint angle refinement easier. However, there is an option to estimate O^v and TO^v using the grid sampled point features, which is denoted as 'Grid Init' in Table. 2. By comparing 'Grid Init' and 'Ours', we can find that 'Grid Init' has a worse body orientation estimation and the global translation estimation along with worse MPJPE and PA-MPJPE.

O/T Setting	MPJPE	PA-MPJPE	O Err	T Err
Grid Init	34.6	27.8	5.4°	83.9
Pass New T	34.3	27.6	**5.1**$^\circ$	**80.9**
Ours	**33.0**	**26.9**	**5.1**$^\circ$	82.2

Table 2: Additional Options for O/T estimations of PaFF on Human 3.6M. 'Grid Init' does not initialize single-view O,T estimations from single-view estimators but init in the first iteration using grid sampled point features. 'Pass New T' is to pass the estimated translation parameters from global T estimation to the next iteration. 'T Err' is in 'mm'.

View Number	MPJPE	PA-MPJPE	PVE	O Err
2	33.8	27.6	50.0	5.2°
3	33.5	27.5	49.7	**5.1**$^\circ$
4 (Ours)	**33.0**	**26.9**	**48.9**	**5.1**$^\circ$

Table 3: Different View Numbers of PaFF on Human 3.6M. The results are averaged. '2' means trained with 2 fixed views. '3' means trained with 3 fixed views.

As mentioned in Sect. Global Translation Estimator, we can choose to update the translation estimation $\{\hat{t_o^v}, \widehat{scale_o^v}\}$ with the belief of correcting the single-view relative translation to drag the final prediction to a better place. However, as shown in Table 2 by comparing 'Pass New T' and 'Ours', the option of 'Pass New T' has a worse MPJPE and PA-MPJPE although it has a better global translation estimation. The reason might be the discontinuity given by the translation update (a move that happens in 2D image plane) actually make the feedback-update process become worse.

Different View Numbers To show the generalization ability of PaFF on view numbers, we additionally train and validate PaFF in 6 sets of 2-view combinations and 4 sets of 3-view combinations in Human 3.6M (has 4 cameras). The result is averaged among the different sets and shown in Table. 3. The table shows that the performance of PaFF 2/3/4 views (MPJPE, PA-MPJPE, PVE, O Err) does not change drastically and the performance is improved as more views are given. We also shows the performance distribution of different views in Fig. 1. It can be seen that the samples lie tightly in the center in most of the metrics (stability) and more views would bring the distribution of the metrics to a better place.

More Qualitative Results

To demonstrate the stability and effectiveness of PaFF, we show a video demonstration in different capture scenes in

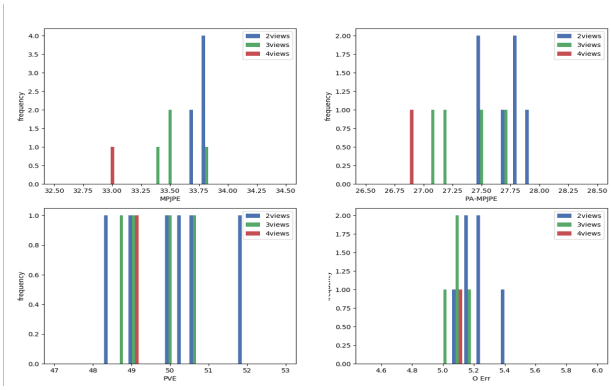

Figure 1: The distribution of evaluation results for 2/3/4 number of views on Human 3.6M dataset (4 cameras): We show results of MPJPE, PA-MPJPE, MVE and O Err. Note that there are 1 view-combination of 4-view, 4 view-combination of 3-view, and 6 view-combination of 2-view. It is clear that more views would improve these metrics.

Human 3.6M, MPI-INF-3DHP, and MTC. In the video, there is also a comparison between the calibration-free PaFF and a state-of-art calibration-free multi-view method Shape-aware (Liang and Lin 2019).

In Fig. 2 and Fig. 3, we demonstrate successful predictions of our PaFF on Human3.6M and MPI-INF-3DHP with challenges such as self-occlusion, object-occlusion, close-contact, and unusual poses. To show the generalization ability of our PaFF, we show the results from MTC (Xiang, Joo, and Sheikh 2019) in Fig. 4. We also show some failure cases of our PaFF in Fig. 5, which shows the limitations of PaFF in some scenes with self-occlusion, close contact, and unusual poses.

Figure 2: Successful Predictions of Our PaFF Estimations on Human3.6M, which showcase normal human pose and other challenging reconstruction scenes with self-occlusion, object-occlusion, close contact, and unusual poses. Each row means a different view. More examples can be seen in the video demo.

Figure 3: Successful Predictions of Our PaFF Estimations on MPI-INF-3DHP, which showcase normal human pose and other challenging reconstruction scenes with self-occlusion, object-occlusion, and unusual poses. Each row means a different view. More examples can be seen in the video demo.

Figure 4: Predictions of Our PaFF Estimations on MTC (Xiang, Joo, and Sheikh 2019), which shows the generalization ability for our method. The results are from a test set of MTC after a mix-training on Human3.6M and MTC. The third example shows feeding one repetitive image (inside the pink box). The fourth example shows changing one viewpoint (inside the blue box). Both of the results show the robustness of our PaFF in viewpoint changing. More examples can be seen in the video demo.

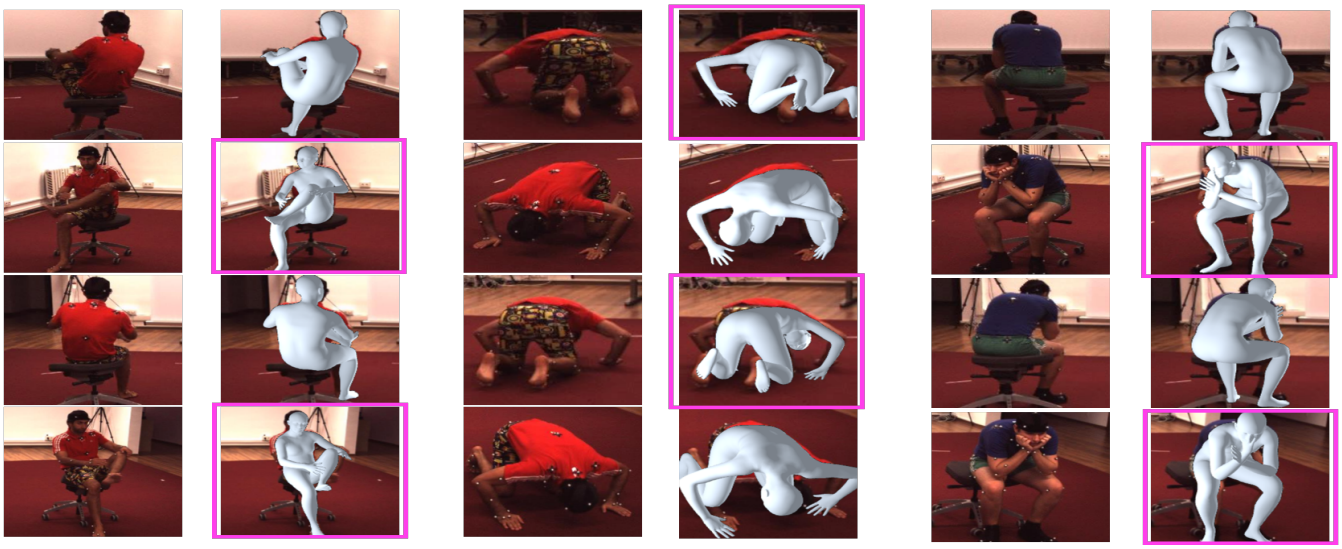

Figure 5: Failure Cases of Our PaFF Estimations on Human3.6M. Each row means a different view. Clear erroneous predictions can be seen in the estimation images inside the pink boxes. These examples show the PaFF's limitations in some scenes with self-occlusion, unusual poses, and self-contact. The third example shows that self-contact can result in a mesh piercing.

References

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B. 2014. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 3686–3693.

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

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7): 1325–1339.

Johnson, S.; and Everingham, M. 2010. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *bmvc*, volume 2, 5.

Joo, H.; Liu, H.; Tan, L.; Gui, L.; Nabbe, B.; Matthews, I.; Kanade, T.; Nobuhara, S.; and Sheikh, Y. 2015. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, 3334–3342.

Kanazawa, A.; Black, M. J.; Jacobs, D. W.; and Malik, J. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7122–7131.

Kolotouros, N.; Pavlakos, G.; and Daniilidis, K. 2019. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4501–4510.

Liang, J.; and Lin, M. C. 2019. Shape-aware human pose and shape reconstruction using multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4352–4362.

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

Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6): 1–16.

Mehta, D.; Rhodin, H.; Casas, D.; Fua, P.; Sotnychenko, O.; Xu, W.; and Theobalt, C. 2017. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 international conference on 3D vision (3DV)*, 506–516. IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Xiang, D.; Joo, H.; and Sheikh, Y. 2019. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10965–10974.

Zhang, H.; Tian, Y.; Zhou, X.; Ouyang, W.; Liu, Y.; Wang, L.; and Sun, Z. 2021. PyMAF: 3D human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11446–11456.