KITRO: Refining Human Mesh by 2D Clues and Kinematic-tree Rotation

Supplementary Material

A. Details on Swing-Twist Decomposition

Swing-twist decomposition for a rotation is a fundamental technique in computer graphics [1, 5, 6] and was first introduced to the human mesh recovery field by HybrIK [9]. As shown in Fig. 1, for any given joint, its rotation $\boldsymbol{\theta}_R \in \mathbb{SO}(3)$ can be decomposed into a 2 degree-of-freedom (DoF) swing rotation R_{sw} and a 1 DoF twist rotation R_{tw}, such that $\boldsymbol{\theta}_R = R_{sw}R_{tw}$.

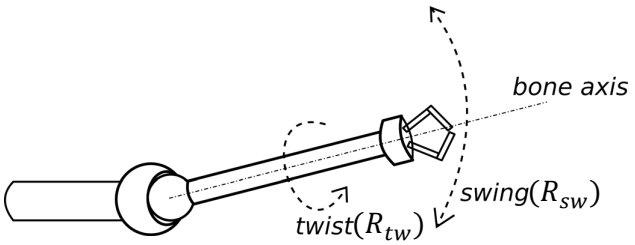


Figure 1. Illustration of the swing-twist decomposition [5].

Swing Rotation. R_{sw} is to rotate the bone from the template relative pose $\vec{t_r}$ to the target relative orientation $\vec{p_r}$. This rotation occurs around axis $\vec{n}$, defined as:

$$\vec{n} = \frac{\vec{t_r} \times \vec{p_r}}{\|\vec{t_r} \times \vec{p_r}\|}, \tag{1}$$

which is orthogonal to both $\vec{t_r}$ and $\vec{p_r}$. The rotation magnitude, denoted as γ, is the angle subtended between $\vec{t_r}$ and $\vec{p_r}$:

$$\cos\gamma = \frac{\vec{t_r} \cdot \vec{p_r}}{\|\vec{t_r}\|\|\vec{p_r}\|}, \quad \sin\gamma = \frac{\|\vec{t_r} \times \vec{p_r}\|}{\|\vec{t_r}\|\|\vec{p_r}\|} \tag{2}$$

Employing Rodrigues' rotation formula, the swing rotation R_{sw} is expressed in closed form as:

$$R_{sw} = \mathcal{I} + \sin\gamma[\vec{n}]_\times + (1 - \cos\gamma)[\vec{n}]_\times^2, \tag{3}$$

where $\mathcal{I}$ represents the 3×3 identity matrix and $[\vec{n}]_\times$ denotes the skew-symmetric matrix of $\vec{n}$.

Twist Rotation. Then R_{tw} is to rotate the bone around bone axis $\vec{t_r}$ itself. Let φ represent the rotation angle. According to Rodrigues' formula, the twist rotation is given by:

$$R_{tw} = \mathcal{I} + \sin\varphi\frac{[\vec{t_r}]_\times}{||\vec{t_r}||} + (1 - \cos\varphi)\frac{[\vec{t_r}]_\times^2}{||\vec{t_r}||^2}, \tag{4}$$

where $[\vec{t_r}]_\times$ is the skew-symmetric matrix of $\vec{t_r}$.

In our work, we focus on refining the swing rotation R_{sw} while preserving the initial twist rotation R_{tw} estimates. This is motivated by the limited variability in twist angles φ due to human physiological constraints, as evidenced by HybrIK's empirical studies [9]. In contrast, the swing rotation R_{sw} exhibits a more significant range of motion, necessitating a more detailed and accurate refinement. Thus, in our approach, we explicitly formulate the bone directions in closed form in order to refine the swing rotation R_{sw}.

B. Proof-of-concept for Solution Selection

In this section, we conduct empirical studies to validate the assumptions discussed in Sec. 4.3 of the main paper. As shown by the red bars in Fig. 2, 87% of bones in the initial HMR estimates are correctly identified as pointing towards or away from the camera. When a 10° margin of error is tolerated in ambiguous cases where only 10 degrees separate two solutions, the accuracy increases to 93% as illustrated by the green bars in Fig. 2. These results affirm the effectiveness of the original HMR model in determining bone direction towards or away from the camera, providing a reliable prior for the decision tree formulation of our method.

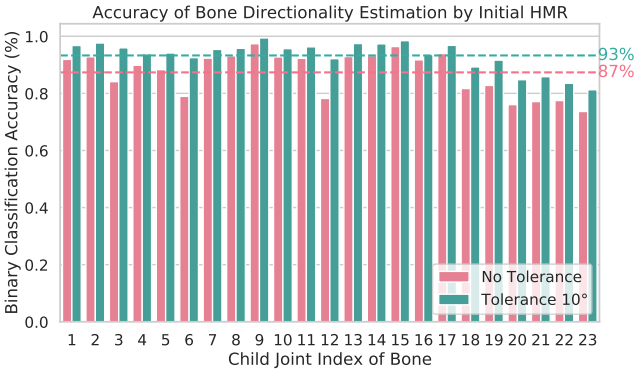


Figure 2. Classification accuracy of bone pointing towards or away from the camera using the HMR model. Red bars indicate 87% accuracy without tolerance, while green bars show improved accuracy up to 93% when a 10° error margin is allowed. These results highlight the HMR model's effectiveness in coarse-grained directionality estimation of bones.

C. Correctness Proof for θ update

In this section, we validate the correctness of the pose update equation (*i.e.*, Eq. 19 in Sec. 4.3 of the main paper).

Proof. Assuming the refinement from the root joint up to the parent joint $\tilde{p}$ of joint p is already done, the current focus is updating rotation $\boldsymbol{\theta}_R^p$ for joint p. The objective is to verify that updating the joint rotation from $\boldsymbol{\theta}_R^p$ to $\boldsymbol{\theta}_R^{p\prime}$ correctly rotate the bone direction of bone (p, c) from $\vec{\mathbf{b}}^{(p,c)}$ to $\vec{\mathbf{b}}_{new}^{(p,c)}$. Considering the joint rotations in $\boldsymbol{\theta}$ are all relative to

each parent joint's coordinate system, the bone direction in the absolute coordinate system is derived as the product of relative rotations from the root joint to the parent joint along the kinematic chain. Hence, the global rotation for joint p in the absolute coordinate system before refinement is:

$$R^p_{abs} = \prod_{i \in KC(p)} \boldsymbol{\theta}^i_R, \tag{5}$$

where $\prod_{i \in KC(p)}$ denotes the matrix product of rotation matrices from the root joint to joint p, with $KC(p)$ representing the kinematic chain. Considering the template relative pose (T pose) for bone (p, c) denoted as $\vec{t}^p_r$, when applied with absolute rotation from Eq. 5, yields the absolute bone direction:

$$\vec{\mathbf{b}}^{(p,c)} = R^p_{abs} \cdot \vec{t}^p_r = \prod_{i \in KC(p)} \boldsymbol{\theta}^i_R \cdot \vec{t}^p_r, \tag{6}$$

As mentioned in the main paper, $R^{(p,c)}_{sw}$, computed via Rodrigues' formula, is the rotation matrix that rotates $\vec{\mathbf{b}}^{(p,c)}$ to $\vec{\mathbf{b}}^{(p,c)}_{new}$ in the absolute coordinate system:

$$\vec{\mathbf{b}}^{(p,c)}_{new} = R^{(p,c)}_{sw} \cdot \vec{\mathbf{b}}^{(p,c)}. \tag{7}$$

Now we verify the updated absolute rotation for joint p after the refinement of Eq. 19:

$$R^p_{abs}{}' = \prod_{i \in KC(\tilde{p})} \boldsymbol{\theta}^i_R \cdot \boldsymbol{\theta}^{p\,\prime}_R, \tag{8}$$

applying this new rotation to the T pose for bone (p, c) results in:

$$\begin{aligned}
R^p_{abs}{}' \cdot \vec{t}^p_r &= \prod_{i \in KC(\tilde{p})} \boldsymbol{\theta}^i_R \cdot \boldsymbol{\theta}^{p\,\prime}_R \cdot \vec{t}^p_r && \text{(from Eq. 8)} \\
&= \prod_{i \in KC(\tilde{p})} \boldsymbol{\theta}^i_R \cdot \Big(\prod_{i \in KC(\tilde{p})} \boldsymbol{\theta}^i_R \Big)^T \\
&\quad \cdot R^{(p,c)}_{sw} \cdot \prod_{i \in KC(p)} \boldsymbol{\theta}^i_R \cdot \vec{t}^p_r && \text{(Main's Eq. 19)} \\
&= R^{(p,c)}_{sw} \cdot \prod_{i \in KC(p)} \boldsymbol{\theta}^i_R \cdot \vec{t}^p_r && \text{(SymMat property)} \\
&= R^{(p,c)}_{sw} \cdot \vec{\mathbf{b}}^{(p,c)} && \text{(from Eq. 6)} \\
&= \vec{\mathbf{b}}^{(p,c)}_{new} && \text{(from Eq. 7)}
\end{aligned} \tag{9}$$

The 'SymMat property' corresponds to the property of symmetry rotation matrices, where the transpose of a rotation matrix equals its inverse. Eq. 9 demonstrates that the updated joint rotation $\boldsymbol{\theta}^{p\,\prime}_R$ correctly modifies $\vec{\mathbf{b}}^{(p,c)}$ to the desired absolute rotation $\vec{\mathbf{b}}^{(p,c)}_{new}$. □

D. More Ablation Studies

In this section, we present extended ablation studies for our framework design.

Tab. 1 details the results of eight different configurations of camera, shape, and pose refinement. The study reveals that when refining only one of these three factors fails to achieve effective results. This ineffectiveness can be attributed to the importance of 2D keypoints and 3D human mesh alignment in our approach. Without the proposed alignment mechanisms in camera and shape refinements, pose refinements alone are also insufficient.

Table 1. Detailed ablation study for three factors on 3DPW dataset. The first row represents the baseline HMR model, and the last row depicts our full model. The identical results in the first two rows are because updating only the camera does not change SMPL parameters, leading to unchanged outcomes.

Camera	Shape	Pose	PA-MPJPE ↓	MPJPE ↓	PVE ↓
✗	✗	✗	43.76	73.67	91.58
✓	✗	✗	43.76	73.67	91.58
✗	✓	✗	44.31	69.92	83.26
✗	✗	✓	45.92	87.33	100.73
✓	✓	✗	44.57	80.03	95.71
✓	✗	✓	35.00	69.02	84.25
✗	✓	✓	28.53	46.68	57.76
✓	✓	✓	**27.67**	**43.53**	**53.44**

In addition, we explored varying learning rates (Fig. 3) and refinement iteration numbers (Fig. 4) for the shape refinement model as discussed in Sec 5.2 of the main paper. The results depicted in these figures demonstrate that both the learning rate and iteration number do not significantly impact the performance of our method. This indicates a robustness in our approach to variations in these parameters.

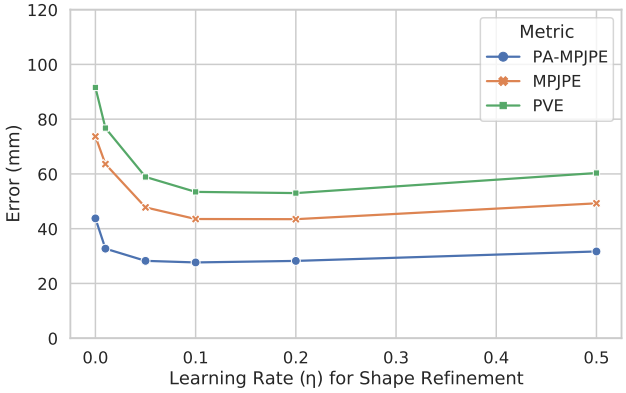


Figure 3. Error with different learning rates for shape refinement. These three line plots (PA-MPJPE, MPJPE, and PVE) illustrate our model's performance stability across a range of learning rates.

E. Improvement Distribution over Samples

In this section, we illustrate the extent of improvement in individual samples following our refinement process. Fig. 5

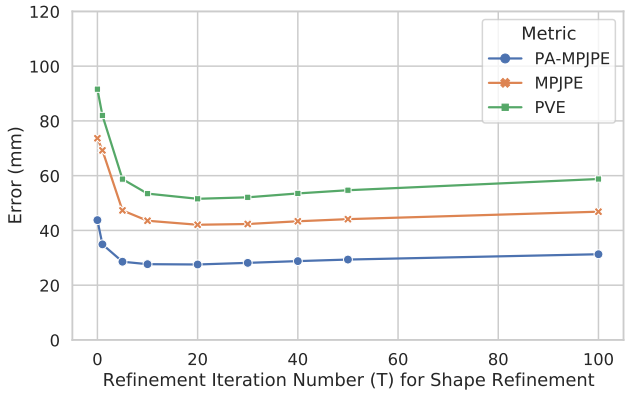


Figure 4. Error with the number of fine-tuning iterations. The results indicate consistent model performance over varying iterations for shape refinement.

and Fig. 6 displays the distribution of performance improvements on the 3DPW and Human3.6M dataset respectively. These visualizations demonstrate that a majority of the samples exhibit significant improvement, again demonstrating the comprehensive efficacy of our method.

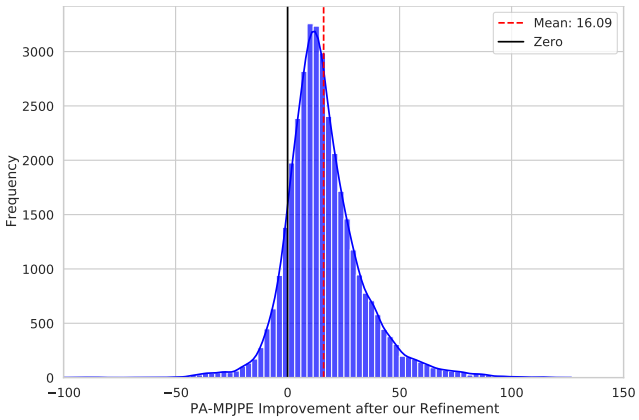


Figure 5. Distribution of performance improvement on 3DPW.

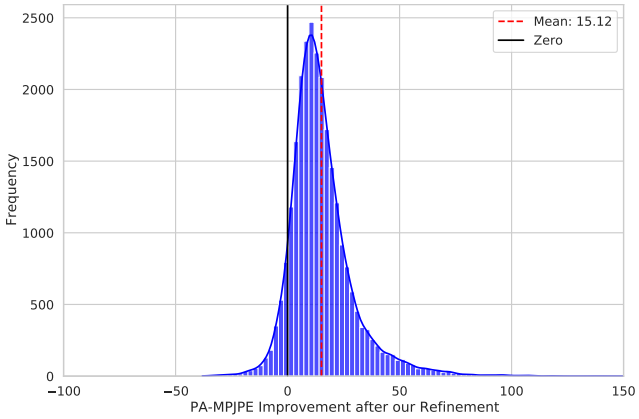
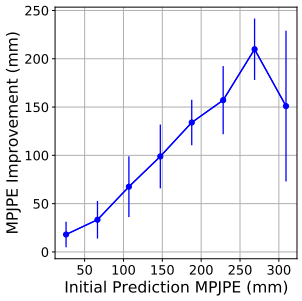
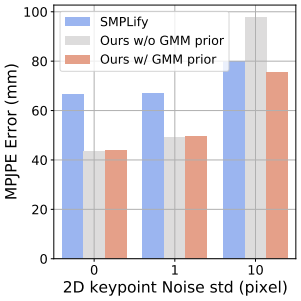


Figure 6. Distribution of performance improvement on the Human3.6M.



(a) Our robustness to high-errors.



(b) Impact of 2D keypoint quality.

F. Impact of Initial Predicted Mesh Quality

As mentioned in the limitation section, our method relies on the initial predicted mesh as a reference for hypothesis selection. Here we investigate how poor initial mesh predictions could impact the improvement. As shown in Fig. 7a, where we plot the MPJPE improvement w.r.t. the original MPJPE error, KITRO withstands high errors with consistently larger improvements. The reason is our global path selection with the decision tree tolerates a few initial bone-facing errors. Additionally, about half of the 13% bone-facing errors are from similar-valued solutions which are not serious mistakes. However, when the initial prediction is too wrong, e.g. 300mm MPJPE or too many bone-facing errors (see Tab. 2), then our improvements decrease.

Table 2. Impact of base models on bone-facing and PA-MPJPE.

Base Model	# of Correct Bone-facing (out of 23 bones)	PA-MPJPE	
		Ours	SMPLify
CLIFFb	20.1 ± 1.8	27.67	36.11
EFT	19.5 ± 2.1	32.34	44.69
SPIN	18.8 ± 2.4	42.46	47.99

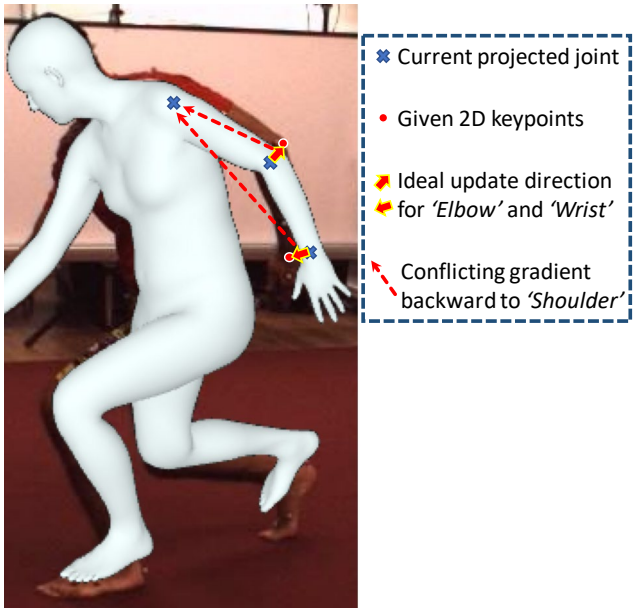


Figure 8. Illustration of gradient conflict: conflicting gradients at the Elbow and Wrist can complicate and negatively impact the optimization of the Shoulder joint, demonstrating a key limitation in previous gradient-based human mesh refinement methods.

Table 3. Refinement results using 2D keypoints mapping from Openpose detection result. '2DKP' denotes '2D Keypoints'; 'GT' denotes 'Ground Truth'.

Method	2DKP	PA-MPJPE ↓	MPJPE ↓	PVE ↓
SMPLify	GT	39.99	71.11	84.28
Ours	GT	27.45	48.42	59.65
SMPLify	Detected	51.35	90.68	107.41
Ours	Detected	**45.88**	**79.80**	**96.60**

G. Impact of Input 2D Keypoint Quality

We follow the protocol of previous human mesh refinement works [2, 4, 7, 8, 10], we and all these works refine 3D pose and shape estimates with ground truth 2D keypoints. Given the different conventions between 2D detectors and SMPL regarding the definition of joints [11], it's non-trivial to directly use detected 2D keypoints. For example, SMPL puts the hip joint where the bone rotation happens while Openpose [3] locates it at the surface landmark where the thigh begins, so one needs to define or learn a mapping from Openpose to SMPL format, or directly train a detector upon SMPL format. Here we performed a rudimentary experiment to train a basic neural network architecture—a simple 3-layer MLP—on 3DPW training data to get the mapping from Openpose to SMPL format. As shown in Tab. 3, the performance of both our method and SMPLify diminishes when subjected to noisy 2D keypoints detection and mapping. This degradation is expected as human mesh refinement inherently relies on the precision of 2D keypoints. Nevertheless, our method is still more robust than SMPLify under the detected keypoints scenario. It is important to note that the current approach, utilizing only an MLP for keypoint format mapping, leaves room for enhancement through more advanced mapping strategies and more training data. However, those explorations extend beyond the scope of this paper and are left for future investigation.

Furthermore, we add Gaussian noise under different standard deviations to simulate the poor 2D keypoint quality as shown in Fig. 7b. Our raw method inherently assumes that the 2D pose is correct and as such, can only withstand small errors (see Fig. 7b, 1-2 px std). This assumption does not hold for larger errors and some intervention is required. A simple strategy is to add a GMM prior, similar to SMPLify. SMPLify uses the GMM as a loss; we use it as a likelihood to reject super unnaturally refined outputs based on erroneous 2D poses (see Fig. 7b, 10 px std). More effective filtering strategies to improve overall robustness to 2D pose error are outside our current scope and are left for future work.

H. Joints and Bones Names on Kinematic-tree

According to SMPL [11], there are 24 joints and 23 bones defined in the human kinematic-tree. We list all joint names and bones in Fig. 9

I. More Visualization Results

In this section, we present additional visualization results. Fig. 10 illustrates the refinement process over iterations on more examples, and Fig. 11 provides further comparisons with SMPLify [2] and CLIFFr [10].

J. Extra Discussion

In this section, we provide more discussion about our method as follows

1) Why previous methods are suboptimal in proximal joints?
As discussed in the main paper, prior human refinement methods typically utilize parametric optimization to optimize all body joints collectively through gradient descent. However, this scheme has limitations with gradient descent: the gradient updates at different joints can be inconsistent or even conflicting. For instance, the optimal update directions for the Elbow and Wrist might significantly differ, as demonstrated in Fig. 8. Such conflicts in gradient directions, when backpropagated to proximal joints like the Shoulder, can lead to complications in their updates, ultimately resulting in suboptimal outcomes in proximal joints refinement (as illustrated in Fig. 1b of the main paper).

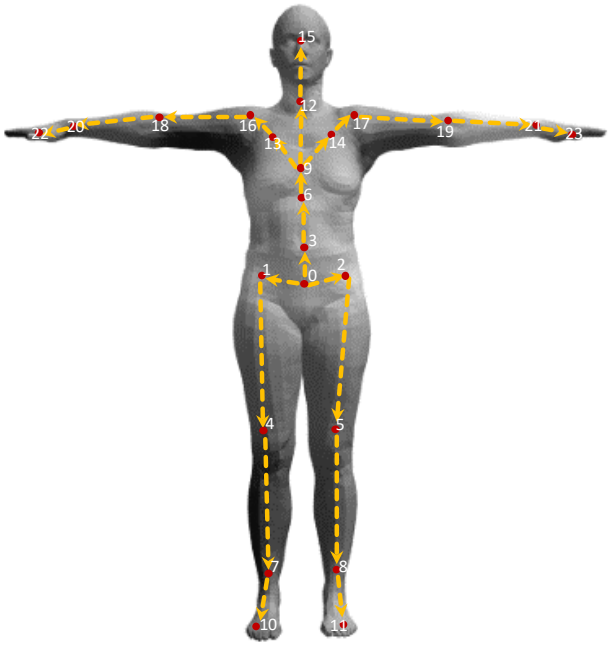
2) What is the computation complexity of KITRO?
As discussed in Sec. 4.3 of the main paper, the computation is efficient due to the decision tree's design, where the calculation depends solely on the depth, allowing nodes at each depth level to be processed in parallel. The most time-consuming part of our method is the Adam-based shape optimization and the total iteration number. However, Fig. 4 and Fig. 5 (in the main paper) show that these numbers are both relatively low for decent results. In practice, KITRO completes testing on all 35,515 samples in 3DPW test set in 15 minutes on a single NVIDIA GeForce RTX 2080 Ti GPU. For comparison, under identical conditions, SMPLify requires 20 minutes, and CLIFFr requires extra fine-tuning on the whole model taking more than 10 hours.

References

[1] Paolo Baerlocher and Ronan Boulic. Parametrization and range of motion of the ball-and-socket joint. In *Deformable Avatars: IFIP TC5/WG5. 10 DEFORM'2000 Workshop November 29–30, 2000 Geneva, Switzerland and AVATARS'2000 Workshop November 30–December 1, 2000 Lausanne, Switzerland*, pages 180–190. Springer, 2001. 1

[2] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, pages 561–578. Springer, 2016. 4

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017. 4



0 - pelvis
1 - left hip
2 - right hip
3 - spine1
4 - left knee
5 - right knee
6 - spine2
7 - left ankle
8 - right ankle
9 - spine3
10 - left foot
11 - right foot
12 - neck
13 - left collar
14 - right collar
15 - head
16 - left shoulder
17 - right shoulder
18 - left elbow
19 - right elbow
20 - left wrist
21 - right wrist
22 - left hand
23 - right hand

pelvis → left hip
pelvis → right hip
pelvis → spine1
left hip → left knee
right hip → right knee
spine1 → spine2
left knee → left ankle
right knee → right ankle
spine2 → spine3
left ankle → left foot
right ankle → right foot
spine3 → neck
spine3 → left collar
spine3 → right collar
neck → head
left collar → left shoulder
right collar → right shoulder
left shoulder → left elbow
right shoulder → right elbow
left elbow → left wrist
right elbow → right wrist
left wrist → left hand
right wrist → right hand

Figure 9. The definition of all 24 joints and 23 bones in the human kinematic-tree used by our approach.

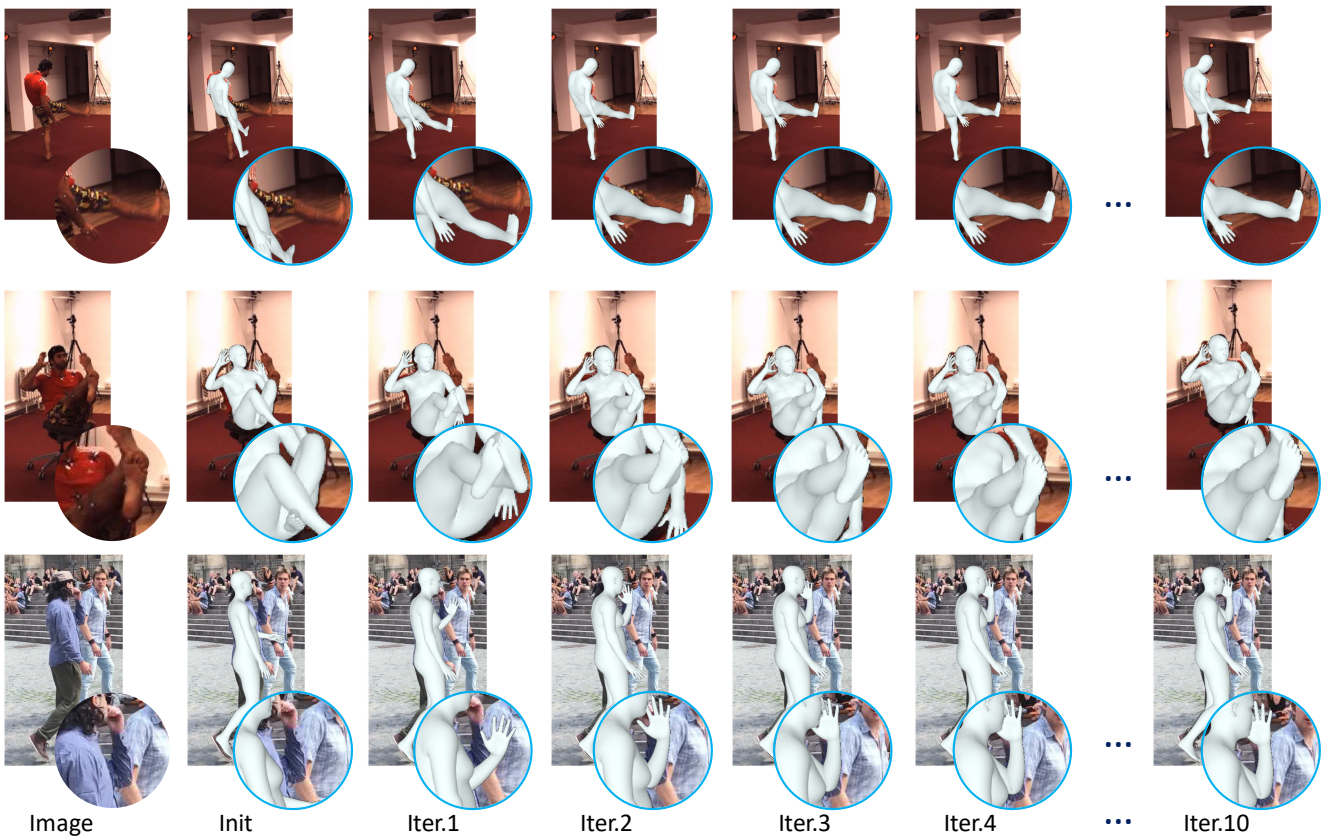


Figure 10. More iterative refinement visualizations.

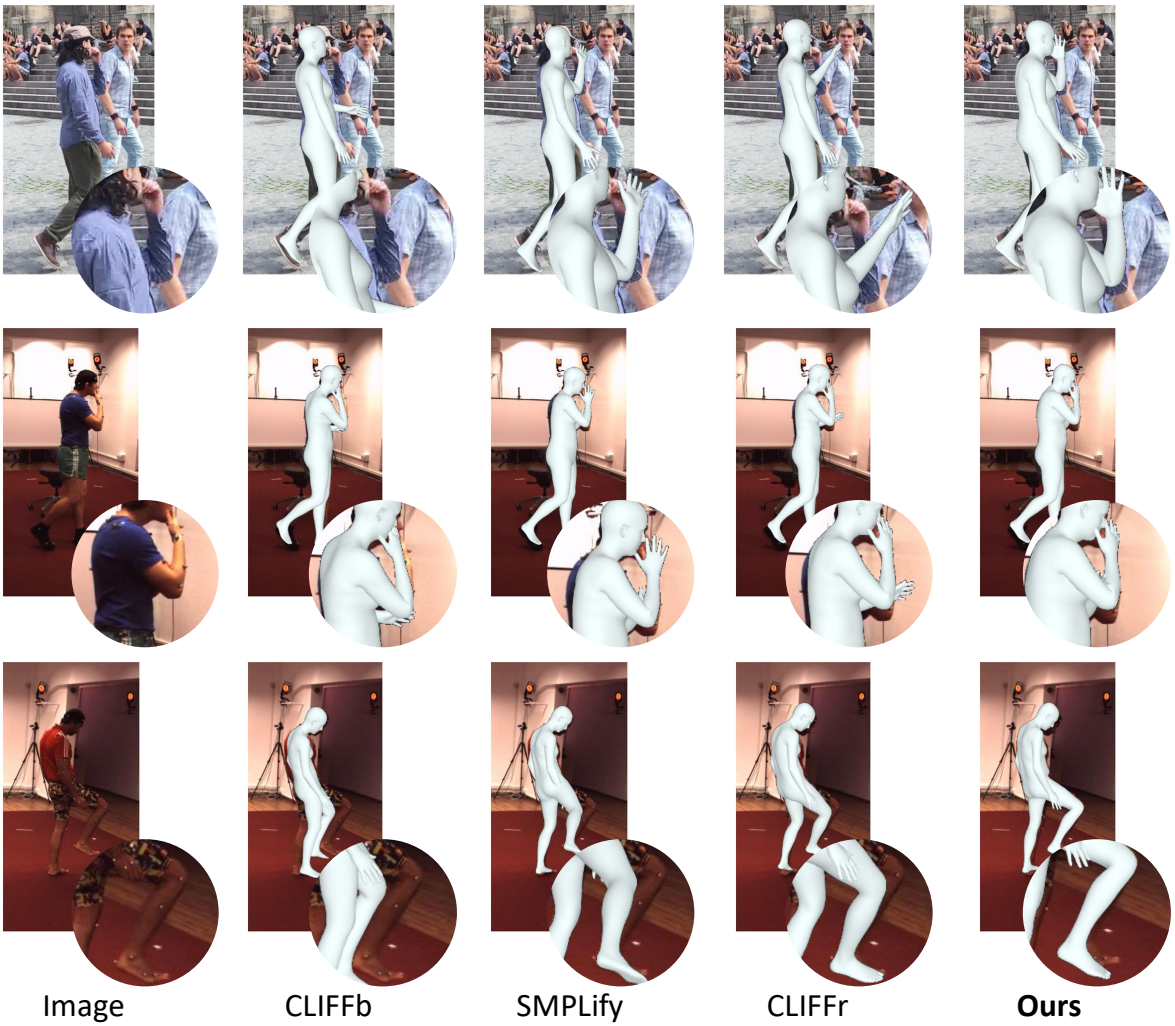


Figure 11. More comparison visualizations.

[4] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision (ECCV)*, pages 769–787. Springer, 2020. 4

[5] Przemysław Dobrowolski. Swing-twist decomposition in clifford algebra. *arXiv preprint arXiv:1506.05481*, 2015. 1

[6] F Sebastian Grassia. Practical parameterization of rotations using the exponential map. *Journal of graphics tools*, 3(3): 29–48, 1998. 1

[7] Shanyan Guan, Jingwei Xu, Michelle Zhang He, Yunbo Wang, Bingbing Ni, and Xiaokang Yang. Out-of-domain human mesh reconstruction via dynamic bilevel online adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(4):5070–5086, 2022. 4

[8] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2021. 4

[9] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3383–3393, 2021. 1

[10] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *European Conference on Computer Vision (ECCV)*, pages 590–606. Springer, 2022. 4

[11] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248:1–248:16, 2015. 4