Temporally Consistent 3D Human Body Animation from RGB Video

Alexandru Ulesan
alex.ulesan@tum.de
03811739

Ece Karasu
ece.karasu@tum.de
03797980

Lucas Pimentel
lucas.pimentel@tum.de
03817782

Ozden Gursoy
ozden.gursoy@tum.de
03778621

Technical University of Munich

1. Motivation and Idea

Monocular RGB video is one of the most accessible sources of human motion, but recovering a realistic 3D body animation from it remains challenging because depth pose are inherently ambiguous from a single view. A common strategy is to combine 2D joint detections with a parametric body model such as SMPL [8], and estimate pose and shape by fitting model parameters so that projected 3D joints match the detected 2D keypoints, as in SMPLify [3]. However, per-frame fitting ignores the strong temporal continuity of human motion.

Motivated by this gap, our project targets smoother and more coherent 3D reconstructions from monocular video by extending a standard SMPLify pipeline [3, 8] with explicit temporal consistency. The key idea is to retain a model-based and training-free formulation while improving stability and convergence by coupling consecutive frames with temporal priors and warm-start initialization.

2. Related Work

2.1. 2D Pose Estimation

Most monocular 3D human reconstruction methods rely on 2D keypoints as an intermediate representation, as they provide a compact and robust description of human pose. While top-down approaches such as HRNet [10] achieve high accuracy, they introduce an additional detection stage that can lead to frame-to-frame inconsistencies when applied to video. In this work, we use OpenPose [4], a bottom-up method that directly detects body joints and provides per-joint confidence scores.

2.2. Parametric human body models

Recovering a full 3D human mesh from monocular observations is severely underconstrained, motivating the use of parametric body models that restrict solutions to realistic human shapes and poses. Early approaches such as SCAPE [2] introduced low-dimensional shape spaces learned from 3D scans, but practical optimization pipelines require representations that are compact, differentiable, and compatible with kinematic constraints. SMPL [8] addresses these requirements by representing the human body as a triangulated mesh whose vertices are a differentiable function of low-dimensional shape and pose parameters, with pose defined as joint rotations in a kinematic tree. This formulation provides a well-behaved parameter space for non-linear least-squares optimization and produces consistent meshes and joint locations across frames, making it well-suited for reconstructing coherent human motion from video.

2.3. Optimization-based methods

Optimization-based approaches estimate 3D pose and shape by fitting SMPL parameters to 2D observations. SMPLify [3] formulates this problem as minimizing a multi-term energy consisting of 2D keypoint reprojection error and priors. This training-free paradigm is flexible and can be applied to arbitrary video sequences. Our baseline follows this per-frame SMPLify optimization, which serves as a reference for evaluating temporal extensions.

2.4. Learning-based methods

When applied independently per frame, optimization-based fitting often suffers from temporal jitter in video sequences. Learning-based methods such as HMR [5] and VIBE [6] address this issue by regressing SMPL [8] parameters using learned temporal models. In contrast, our work remains training-free, and improves temporal stability by introducing explicit temporal regularization.

3. Method

3.1. Overview

Our method takes a monocular RGB video as input and produces a temporally coherent sequence of 3D body meshes. An overview of the pipeline is shown in Figure 1. First, we extract per-frame 2D keypoints using OpenPose. Next, we load a preprocessed SMPL model. Given these inputs, we fit the SMPL pose, shape, and global translation to the

detected keypoints using the Ceres non-linear least-squares solver. The optimization follows a two-phase strategy: a torso-based initialization to estimate depth and root orientation, followed by a full solve with a coarse-to-fine schedule that gradually relaxes the prior weights. To ensure smooth motion across frames, we apply temporal regularization and start each frame from the previous frame's solution. Finally, we render the recovered meshes back onto the input video for qualitative inspection.

3.2. Data Preprocessing

We assume as input a monocular video showing a single person. Before optimization, we convert the original SMPL model files into a single unified model representation that can be loaded efficiently by our fitting pipeline. This preprocessing step combines the official SMPL model (male/female), the Gaussian Mixture Model (GMM) pose prior used in SMPLify, and an OpenPose joint regressor into one consistent format. The resulting model package contains all the required data for the pipeline, including template mesh, shape and pose blend shapes, linear blend skinning weights, the kinematic tree, joint regressor, and the GMM statistics. We select the appropriate model, male or female, according to the video.

3.3. 2D Pose Estimation

For each video frame, we compute 2D keypoints using OpenPose with the BODY_25 joint layout. Each joint is represented by image coordinates (u, v) and a confidence score $c \in [0, 1]$. These keypoints are the only observations used to reconstruct the 3D pose and shape.

3.4. Optimization

We denote the full set of parameters as $\mathbf{\Theta} = \{\mathbf{t}, \boldsymbol{\theta}, \boldsymbol{\beta}\}$, where:

- $\mathbf{t} \in \mathbb{R}^3$: global translation,
- $\boldsymbol{\theta} \in \mathbb{R}^{72}$: pose parameters (24 joints $\times$ 3 axis-angle),
- $\boldsymbol{\beta} \in \mathbb{R}^{10}$: shape coefficients.

These parameters are estimated by minimizing a non-linear least-squares objective with Ceres Solver [1]. Following SMPLify [3], the objective is:

$$E(\mathbf{\Theta}) = E_{\text{reproj}}(\mathbf{\Theta}) + \lambda_\theta E_\theta(\boldsymbol{\theta}) + \lambda_\beta E_\beta(\boldsymbol{\beta}) + \lambda_a E_a(\boldsymbol{\theta}) + E_{\text{temp}}(\mathbf{\Theta}), \tag{1}$$

where:

- E_{reproj} is the **reprojection error**, which measures the weighted distance between SMPL joints projected onto the image and the corresponding 2D OpenPose keypoints, with each term scaled by the keypoint confidence score;
- E_θ is a **pose prior** based on a Gaussian Mixture Model (GMM), penalizing the Mahalanobis distance of the body pose $\boldsymbol{\theta}_{4:72}$ to the closest Gaussian component;
- $E_\beta = \|\boldsymbol{\beta}\|^2$ is an L_2 **shape prior** that prevents extreme body deformations;
- E_a enforces **anatomical joint limits** by penalizing rotations that exceed plausible ranges at specific joints such as elbows and knees;
- E_{temp} is a **temporal regularization** term described in Section 3.5.

Following SMPLify, the problem is solved in an initialization step followed by four optimization stages with a coarse-to-fine weight schedule. In the initialization step, only the global orientation and translation are optimized using the more reliably detected torso keypoints. In the staged optimization step, all body joints are included and the regularization weights are progressively relaxed, in order to avoid implausible local minima.

3.5. Temporal Regularization

When fitting video sequences, we extend the per-frame objective (1) with temporal regularization terms that enforce smooth motion consistency. These terms operate in 3D joint space rather than in parameter space, directly penalizing physically implausible motion.

Let $\mathbf{J}_j^{(t)} \in \mathbb{R}^3$ denote the position of SMPL joint j at frame t, relative to the root. Given the two preceding frames $t-1$ and $t-2$, we define the discrete velocity and acceleration:

$$\mathbf{v}_j^{(t)} = \mathbf{J}_j^{(t)} - \mathbf{J}_j^{(t-1)}, \tag{2}$$

$$\mathbf{a}_j^{(t)} = \mathbf{v}_j^{(t)} - \mathbf{v}_j^{(t-1)}. \tag{3}$$

The temporal cost consists of three components:

Acceleration. Penalizing acceleration suppresses high-frequency jitter while still permitting smooth changes in velocity.

Velocity (friction). A velocity penalty acts as friction to prevent drift, with its strength adapted based on keypoint visibility. When a joint is visible in the 2D detections (i.e., its confidence score exceeds a threshold), sufficient reprojection evidence exists to guide it. When a joint is occluded, the velocity penalty is increased to prevent unobserved joints from drifting freely.

Shape consistency. Since body shape should remain constant across a sequence, we penalize frame-to-frame deviations in $\boldsymbol{\beta}$, as simply freezing the shape parameters after the first few frames led to shape–pose ambiguities in later frames.

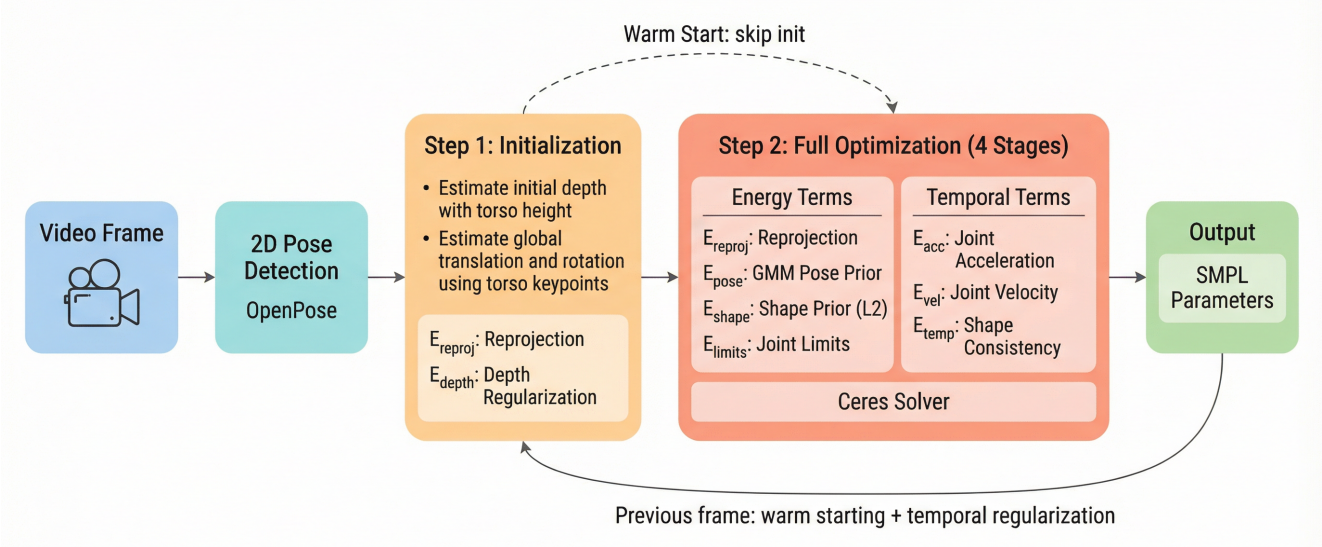

Figure 1. Pipeline Overview.

4. Results

We evaluate our pipeline on the MPI-INF-3DHP test dataset [9], which consists of six video sequences featuring diverse activities and camera viewpoints.

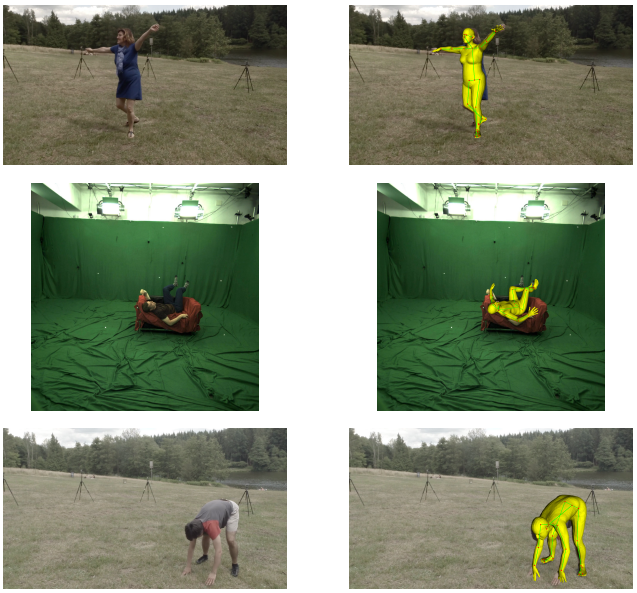

Figure 2. Original frames (left) and corresponding results (right).

For qualitative evaluation, we project the posed SMPL meshes back onto the input frames to verify alignment with the original footage, as can be seen in Figure 2. For quantitative evaluation, we report four standard metrics based on the estimated 3D joint coordinates:

- **MPJPE** [9]: Mean Per Joint Position Error, the average Euclidean distance between estimated and ground-truth joints;
- **PA-MPJPE** [9]: Procrustes-Aligned MPJPE, computed after rigid alignment to isolate structural accuracy from global positioning errors;
- **PCK** [7]: Percentage of Correct Keypoints, the fraction of joints within a 150 mm error threshold;
- **AUC** [7]: Area Under the Curve, aggregating PCK scores across thresholds from 0 to 150 mm to provide a robust overall accuracy measure.

Table 1. Metrics computed across all video sequences from the MPI-INF-3DHP test dataset. Best results per metric in **bold**.

	Baseline	Proposal
MPJPE (mm)	329.87	**246.95**
PA-MPJPE (mm)	192.52	**129.95**
PCK (%)	**62.48**	57.18
AUC	**33.35**	30.68

Table 1 compares the per-frame baseline against our full pipeline, which adds warm starting and temporal regularization. The proposed pipeline reduces MPJPE by 82.9 mm (25.1%) and PA-MPJPE by 62.6 mm (32.5%), indicating more accurate 3D pose estimates on average. However, PCK decreases by 5.3 percentage points and AUC by 2.7 points.

5. Analysis

The divergent behavior between mean error metrics (MPJPE, PA-MPJPE) and threshold-based metrics (PCK,

AUC) reveals a trade-off introduced by temporal regularization.

MPJPE and PA-MPJPE measure the average Euclidean error across all joints and frames, making them sensitive to outliers. The per-frame baseline optimizes each frame independently, while this can recover good poses on individual frames, it also produces catastrophic failures that heavily inflate the mean error. Warm starting eliminates these failures by initializing from the previous frame's solution, and temporal smoothing further prevents unexpected jumps between frames.

PCK and AUC, on the other hand, measure the fraction of joints falling below a certain threshold. These metrics are insensitive to the magnitude of large errors, so the baseline's catastrophic frames therefore have limited impact. The temporal pipeline, by contrast, introduces two effects that slightly degrade threshold-based accuracy. First, warm starting bypasses the initialization, relying instead on the previous frame's solution. Second, temporal regularizer prevents jitter but can also bias joints away from their per-frame optimum.

6. Conclusion

We presented a monocular 3D body fitting pipeline that extends per-frame SMPLify optimization with warm starting and temporal regularization. On the MPI-INF-3DHP test set, the proposed approach reduces MPJPE by 25% and PA-MPJPE by 33% compared to the per-frame baseline, demonstrating that enforcing temporal consistency yields more accurate pose estimates for video sequences. Future work could address the observed drop in PCK and AUC by incorporating more expressive motion priors.

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres solver. https://github.com/ceres-solver/ceres-solver, 2023. 2

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005. 1

[3] 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 *Computer Vision – ECCV 2016*. Springer International Publishing, 2016. 1, 2

[4] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1

[5] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[6] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[7] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 3

[8] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 1

[9] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 3

[10] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition, 2020. 1