A1. 3D Mesh Reconstruction Details

As described in the main paper (Sec. 3.1.1), we reconstruct 3D human motion in the form of SMPL-X meshes by optimizing over multi-view, third-person camera observations in an iterative loop. Our goal is to recover the optimal SMPL-X parameters at each time step t

$$\Theta_t = \{\theta_t,\ \beta,\ \tau_t\},$$

where $\theta_t \in \mathbb{R}^{66}$ are the 3D joint angles for the 22 body joints, $\beta \in \mathbb{R}^{10}$ is the shape vector, and $\tau_t \in \mathbb{R}^6$ denotes the global root orientation and translation. Following SMPLify-X [3], we do not optimize the joint angles θ_t directly. Instead, we optimize a low-dimensional latent pose vector z_t, and recover the full pose via a pretrained VAE decoder:

$$\theta_t = V(z_t)\,.$$

At each iteration, we synthesize the fitted mesh

$$\mathcal{M}_t = S\big(V(z_t),\, \beta,\, \tau_t\big),$$

where $S(\cdot)$ is the differentiable SMPL-X forward function that maps parameters to a detailed full-body mesh. We then obtain the 3D keypoints positions by applying a fixed regression matrix J to the mesh vertices:

$$\hat{x}_t^{3D} = J\,\mathcal{M}_t\,.$$

Finally, these 3D keypoints are projected into each camera's image plane via the standard pinhole model:

$$\hat{x}_t^{2D} = \Pi\big(\hat{x}_t^{3D} \mid P\big),$$

where $\Pi(\cdot)$ is the projection operation and P is the camera projection matrix. The optimization is formulated as

$$\hat{\mathbf{z}},\, \hat{\beta},\, \hat{\boldsymbol{\tau}} = \arg\max_{\mathbf{z},\,\beta,\,\boldsymbol{\tau}} \big(\lambda_{3D}\,\mathcal{L}_{3D} + \lambda_{2D}\,\mathcal{L}_{2D} + \lambda_z\,\mathcal{L}_z + \lambda_\beta\,\mathcal{L}_\beta + \lambda_{\text{smooth}}\,\mathcal{L}_{\text{smooth}}\big)\,.$$

$\mathcal{L}_{3D}$ is the 3D keypoint loss that computes the euclidean distance between the observed 3D keypoints $\mathbf{x}^{3D}$ and the fitted keypoints $\hat{\mathbf{x}}^{3D}$

$$\mathcal{L}_{3D} = \frac{1}{T\,K}\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\|x_{t,k}^{3D} - \hat{x}_{t,k}^{3D}\|,$$

where T and K are the number of frames and keypoints, respectively. 2D keypoint loss $\mathcal{L}_{2D}$ is robustified using the Geman–McClure function [1, 2] $\phi(r;\rho) = \frac{r^2}{r^2+\rho^2}$, with hyperparameter $\rho > 0$. Specifically:

$$\mathcal{L}_{2D} = \frac{1}{T\,K\,C}\sum_{t=0}^{T-1}\sum_{k=0}^{K-1}\sum_{c=0}^{C-1}\phi\big(\|x_{t,k,c}^{2D} - \hat{x}_{t,k,c}^{2D}\|;\rho\big),$$

where C is the number of cameras. Pose and shape priors use simple ℓ_2 regularization:

$$\mathcal{L}_z = \frac{1}{T}\sum_{t=0}^{T-1}||z_t||_2^2, \qquad \mathcal{L}_\beta = ||\beta||_2^2.$$

Finally, the temporal smoothing term $\mathcal{L}_{\text{smooth}}$ penalizes acceleration in both the 3D keypoint trajectories and the joint-angle sequence of $J = 22$ body joints. Concretely,

$$\mathcal{L}_{\text{smooth}} = \underbrace{\frac{1}{(T-2)\,K}\sum_{t=0}^{T-3}\sum_{k=0}^{K-1}\big\|\hat{x}_{t,k}^{3D} - 2\,\hat{x}_{t+1,k}^{3D} + \hat{x}_{t+2,k}^{3D}\big\|_2^2}_{\mathcal{L}_{\text{smooth}}^{3D}} + \underbrace{\frac{1}{(T-2)\,J}\sum_{t=0}^{T-3}\sum_{j=0}^{J-1}\big\|\theta_{t,j} - 2\,\theta_{t+1,j} + \theta_{t+2,j}\big\|_2^2}_{\mathcal{L}_{\text{smooth}}^{\theta}}.$$

Our implementation follows a three-stage pipeline. First, we initialize the root parameters τ_t by aligning unposed template keypoints (shoulders and hips) to the triangulated 3D targets. In Stage 1, we optimize parameters over the first 100 frames where the subjects are particularly doing a simple pose (such as A or T-pose) using $\lambda_{3D} = 5.0$, $\lambda_{2D} = 0.0$, $\lambda_{\text{smooth}} = 0.1$, $\lambda_z = 0.005$, and $\lambda_\beta = 0.04$. In the subsequent stages, we freeze β as the value optimized from Stage 1. In Stage 2, we optimize the latent pose vectors z_t and τ_t across all frames with $\lambda_{3D} = 5.0$, $\lambda_{2D} = 0.0$, $\lambda_{\text{smooth}} = 2.5$, and $\lambda_z = 0.1$. Finally, in Stage 3, we refine z_t and τ_t using smaller priors with $\lambda_{3D} = 1.6$, $\lambda_{2D} = 0.5$, $\lambda_{\text{smooth}} = 10.0$, and $\lambda_z = 0.01$. For Stage 2 and Stage 3, we crop each sequence into 500-frames chunks and optimize separately.

A2. Sensor Signal Processing

Video and contact sensor synchronization. To synchronize the contact-sensor recordings with the video, each trial begins and ends with the subject performing hand claps. These claps generate large, easily identifiable peaks in the sensor signal (see Figure 1), which we mark as the start and end synchronization events. For each marked peak, we then manually locate the corresponding video frame in which the subject's hands are in the clapping pose. By matching the timestamps of these sensor-signal peaks to the video-frame indices, we can synchronize and resample contact sensor signals corresponding to the video frames.

Analog-to-binary contact mask conversion. The raw contact signals were acquired using an Arduino microcontroller, which reads the voltage across each FSR sensor as a 10-bit integer in the range 0–1023. Although the resistor on each sensor defines a sensitivity, optimal thresholds must be tuned for each contact point according to its placement and the

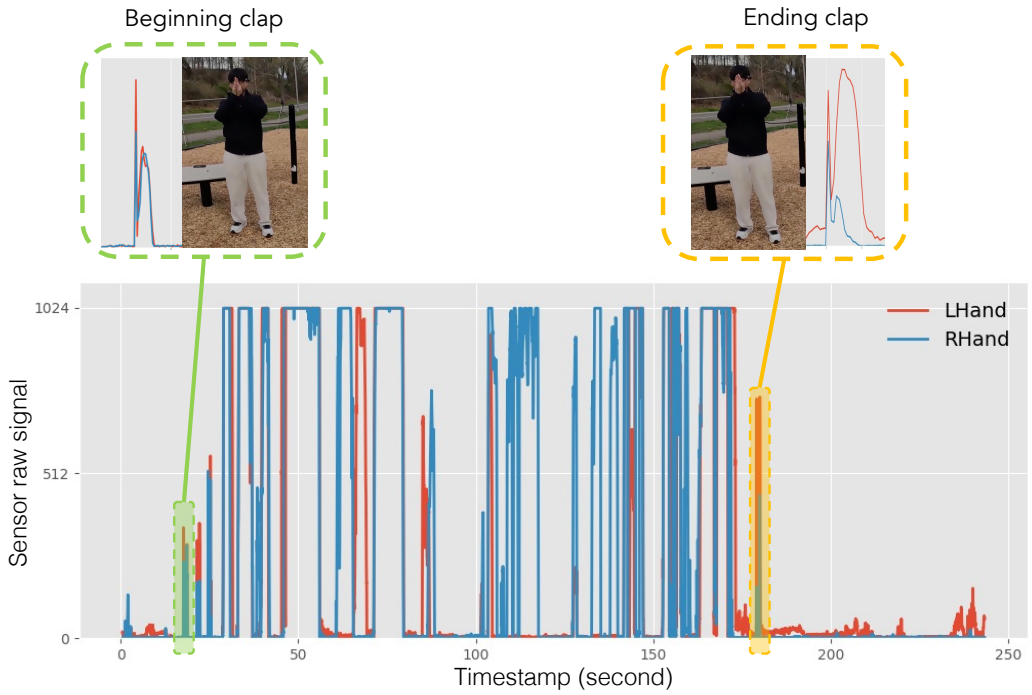

Figure 1. Synchronization of contact sensor and video via hand-clap events. The sensor trace shows pronounced peaks at the start and end of each sequence, corresponding to the subject's hand claps. These peaks allow manual alignment of the sensor data with the matching video frames.

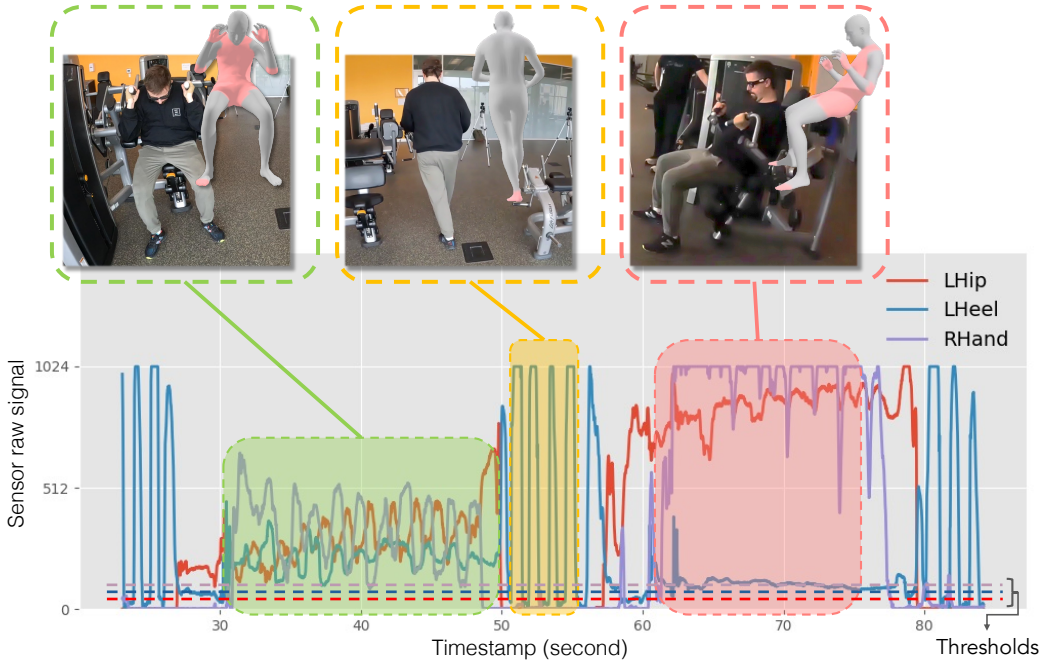

Figure 2. Raw analog contact-sensor signals from three FSR channels over a representative trial. Each trace shows the Arduino-measured voltage (0–1023) for one sensor. Shaded regions highlight distinct motion segments with characteristic signal patterns, and dashed horizontal lines indicate the per-sensor thresholds used to binarize contact events.

characteristics of the performed motion. To this end, we plot exemplar recordings from three sensors (see Fig. 2), identify segments corresponding to distinct motions, and visually select per-sensor thresholds that cleanly separate true contact events from baseline noise. Samples exceeding these thresholds are binarized into a frame-by-frame contact mask. Because point-based sensors can still miss contacts in adjacent uninstrumented areas or register spurious peaks due to clothing pressure, we manually inspect and correct any frames with obvious misclassifications. This procedure produces reliable binary contact labels for downstream analysis.

References

[1] 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*, pages 561–578. Springer International Publishing, 2016.

[2] Stuart Geman and Donald E. McClure. Statistical methods for tomographic image reconstruction. 52(4):5–21, 1987.

[3] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.