EMDB: The Electromagnetic Database of Global 3D Human Pose and Shape in the Wild - Supplementary Material

Manuel Kaufmann[1] Jie Song[1] Chen Guo[1] Kaiyue Shen[1] Tianjian Jiang[1]
Chengcheng Tang[2] Juan José Zárate[1] Otmar Hilliges[1]

[1]ETH Zürich, Department of Computer Science [2]Meta Reality Labs

Contents

A. Details on EMDB's Contents

We describe the activity of each sequence appearing in EMDB in more detail in Tab. 4 and Tab. 5. From these tables we note that 55.4% (44.6%) of all frames in EMDB are performed by female (male) participants. Furthermore, 15.5% of all data was recorded on our multi-view volumetric capture system (MVS, [7]), 12.7% was recorded indoors (but not on the MVS) and the remaining 71.8% of EMDB were captured outdoors.

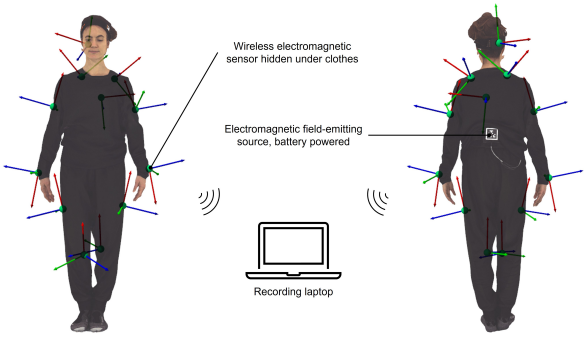

Figure 10: Sensor placement and recording setup. The Apriltag on the source is only required for the body calibration.

B. Sensor Placement

We place sensors under regular clothing as shown in Fig. 10. All sensors communicate wirelessly with two receivers plugged into a recording laptop via USB. The laptop usually stays within 3-4 meters of the participant to minimize packet loss. The sensors measure their position and orientation relative to the EM field emitting source mounted on the lower back. For more details please refer to [10]. As seen in Fig. 10 an Apriltag [12, 15, 22] is attached to the source, which we only require for the body calibration as explained in more detail in Sec. D. Sensors and source are battery-powered, which can all be neatly stowed away under regular clothing. For a depiction of how the sensors are strapped to the body, please refer to Fig. 13.

C. SMPL Registration to Multi-View Data

In this section we explain how we obtain SMPL [14] ground-truth registrations from data recorded with our MVS [7]. We use the same procedure to obtain the ground-truth SMPL shape β on minimally clothed scans as mentioned in Sec. 4.2 of the main paper and to register SMPL parameters

Subj. ID	Seq. ID	Loc.	Activity	# Frames
P0	0	MVS	arm rotation, leg raises, jump	381
P0	1	MVS	walk, crouch, bend, arm swing, arm raises	554
P0	2	MVS	punches, pirouette, leg curls	543
P0	3	MVS	upper body range of motion	661
P0	4	MVS	upper body range of motion	667
P0	5	MVS	kicks, punches, leg raises, arm swings	635
P0	6	I	jumping, swirl, boxing	858
P0	7	O	lunges, push-ups	1 277
P0	8	O	remove jacket	591
P0	9	O	long walk, straight line	2 009
P0	10	O	range of motion	1 183
P1	11	MVS	range of motion	951
P1	12	MVS	walk in circle, punch, kick, crouch, upper body twist	991
P1	13	O	very long walk	4 028
P1	14	O	climb on platform, sit, jog around platform	1 284
P1	15	O	range of motion	1 348
P1	16	O	warm-up, side-stepping, push-ups	1 365
P2	17	MVS	arm and leg motions, jump	574
P2	18	MVS	occluded arm motions, crouch	577
P2	19	I	walk off stage	1 299
P2	20	O	long walk	2 713
P2	21	O	sit, stand and balance	1 272
P2	22	O	play with basketball	1 438
P2	23	O	hug tree	1 086
P2	24	O	long walk, climbing	3 280
P3	25	MVS	walk while moving arms	528
P3	26	MVS	cross arms, arm motions with occlusions, squats	557
P3	27	I/O	walk off stage	1 448
P3	28	O	lunges while walking	1 836
P3	29	O	walk up stairs	1 205
P3	30	O	walk down stairs	1 128
P3	31	O	workout	1 216
P3	32	O	soccer warmup 1	1 084
P3	33	O	soccer warmup 2	1 433
P4	34	MVS	walk, upper body twist, arm motions	541
P4	35	I	walk along hallway	1 226
P4	36	O	long walk	2 160
P4	37	O	jog in circle	881
Total				**46 808**

Table 4: More detailed description of the sequences appearing in EMDB for subjects P0 to P4 (male participants). For subjects P5 to P9 please refer to table Tab. 5. *Loc.* refers to where the recording took place (*MVS*: In our multi-view volumetric capture studio, *I*: Indoor, *O*: Outdoor). The data is recorded at 30 fps (26 minutes).

Subj. ID	Seq. ID	Loc.	Activity	# Frames
P5	38	MVS	arm and leg motions	500
P5	39	MVS	walk and jog in circle	600
P5	40	I	walk in big circle	2 661
P5	41	I	jog in circle, workout	1 762
P5	42	I	freestyle dancing	1 291
P5	43	I	drink water	1 400
P5	44	I	range of motion	1 381
P6	45	MVS	range of motion	994
P6	46	MVS	jumping jacks, lunges, squats, torso twists	1 005
P6	47	O	slalom with occlusions	677
P6	48	O	walk down slope	1 959
P6	49	O	walk down and up big stairs	1 559
P6	50	O	workout	1 532
P6	51	O	dancing, lunges	1 427
P6	52	O	walk behind low wall	509
P7	53	MVS	range of motion, walk	967
P7	54	MVS	crouching, arm crossing, jump, balance on one leg	1 045
P7	55	O	long walk	2 179
P7	56	O	walk stairs up and down	1 120
P7	57	O	lie, rock on chair	1 558
P7	58	O	parcours!	1 332
P7	59	O	range of motion	1 839
P7	60	O	push-ups, dips, jumping jacks	1 693
P7	61	O	sit on bench, walk	1 914
P8	62	MVS	freestyle movement	1 035
P8	63	MVS	range of motion with occlusions	1 007
P8	64	O	skateboarding	1 704
P8	65	O	walk straight line	1 981
P8	66	O	range of motion	1 808
P8	67	O	sprint back and forth	801
P8	68	O	handstand	1 606
P8	69	O	cartwheel, jump	656
P9	70	MVS	range of motion	1 045
P9	71	MVS	jog in circle, head motions	970
P9	72	O	jump on bench	707
P9	73	O	body scanner motions	1 264
P9	74	O	range of motion	1 814
P9	75	O	slalom around tree	1 117
P9	76	O	sitting	1 768
P9	77	O	walk stairs up	728
P9	78	O	walk stairs up and down	1 083
P9	79	O	walk in rectangle	1 917
P9	80	O	walk in big circle	2 240
Total				**58 155**

Table 5: More detailed description of the sequences appearing in EMDB for subjects P5 to P9 (female participants). For subjects P0 to P4 please refer to table Tab. 5. *Loc.* refers to where the recording took place (*MVS*: In our multi-view volumetric capture studio, *I*: Indoor, *O*: Outdoor). The data is recorded at 30 fps (32.3 minutes).

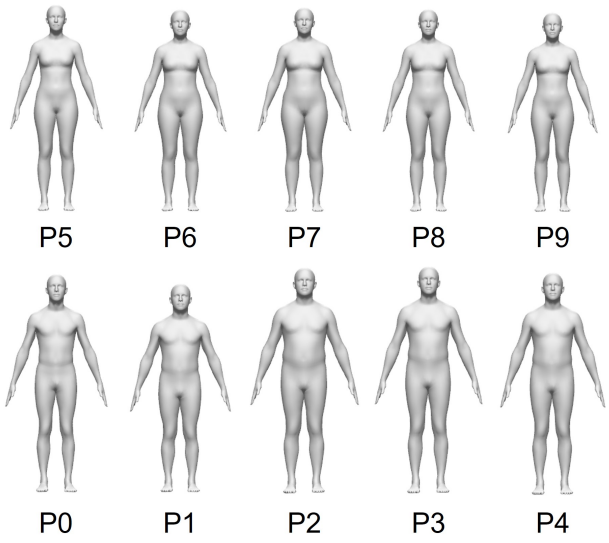

Figure 11: Shapes of all 10 participants appearing in EMDB. For reference, the height of P0 is 177 cm.

for the pose accuracy evaluations in Sec. 6.1 of the main paper.

Our MVS provides high-quality 3D scans (in the form of watertight meshes with 40k vertices) and high-resolution RGB images from 53 camera views. We fit an SMPL [14] model parameterized by $\mathbf{\Omega} = (\boldsymbol{\theta}_r, \boldsymbol{\theta}_b, \mathbf{t}, \boldsymbol{\beta})$ to this data. We use a gender-specific model with the gender that our participants have indicated on a respective questionnaire or a neutral body model if they chose not to answer that question. For a visualization of scans and registrations, please refer to Fig. 12.

C.1. 3D Keypoint Triangulation

We start the registration process by detecting Openpose 2D keypoints [5, 6, 18, 23] in the multi-view camera images. When we record sequences with the MVS and the iPhone together, some of the RGB images will show multiple people, *i.e.*, including the person holding the iPhone. This would require running a person-tracker to isolate the correct Openpose Keypoints. To avoid this complication, we instead back-project the high-quality scans obtained from the MVS into the camera views with a white background and then run Openpose on these images. Note that the quality of the scans is not affected by the presence of a second person on the capture stage, as the assistant holding the iPhone is outside the calibrated capture volume. Given the 2D keypoint detections in the various views, we triangulate 3D keypoints using ordinary least squares to solve the over-determined linear system. This produces 25 3D keypoints per time step in COCO format, denoted as $\mathbf{x}_i^{3D}$.

C.2. SMPL Fitting

After triangulation, we employ an optimization procedure to fit SMPL to the 3D keypoints and scans following [1, 17]. Our implementation follows code published by [3, 4] and uses PyTorch [16]. We explain the optimization terms and details of the fitting procedure in the following.

3D Keypoint Term To optimize the pose, a 3D keypoint term is used, where keypoints $\hat{\mathbf{x}}_j^{3D}$ are extracted from the SMPL joints $X(\mathbf{\Omega})$ via a pre-defined mapping, and $\mathbf{x}_j^{3D}$ are the triangulated points described in Sec. C.1. Joints are weighted via $w_j \in \mathbb{R}$.

$$E_{\text{J}} = \frac{1}{J} \sum_{j=1}^{J} w_j \cdot ||\mathbf{x}_j^{3D} - \hat{\mathbf{x}}_j^{3D}||_2^2 \tag{8}$$

Surface Term To incorporate dense surface information from our scans, we use the following term:

$$E_{\text{S}} = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{v} \in \mathcal{V}} \rho(d(\mathbf{v}, \mathcal{M}(\mathbf{\Omega}))) + \frac{1}{|\mathcal{M}(\mathbf{\Omega})|} \sum_{\mathbf{m} \in \mathcal{M}(\mathbf{\Omega})} \rho(d(\mathbf{m}, \mathcal{V})) \tag{9}$$

where $\mathcal{V}$ is the set of points sampled from the scan, $\mathcal{M}$ the SMPL mesh, $d(\mathbf{p}, \mathcal{R})$ measures the squared Euclidean distance of a point $\mathbf{p} \in \mathbb{R}^3$ to the closest vertex in the point cloud $\mathcal{R}$ and ρ is a generalized robustifier [2]. We sample $|\mathcal{V}| = 50\,000$ points on each scan. To encourage the SMPL mesh to lie within the scan, we follow [1] and enforce all points $\mathbf{v} \in \mathcal{V}$ that lie outside of the SMPL mesh $\mathcal{M}$ to move inside by increasing their weight in the surface term E_{S}.

Regularization It may happen that the SMPL spine is bent unnaturally leading to bulging belly artifacts. To counteract this, we leverage the tracked Apriltag pose on the EM source strapped to the lower back to add a regularizer E_{spine}. This prior enforces that a set of hand-picked SMPL vertices that are close to the Apriltag remain close to it. We further add regularizers E_{reg} to penalize impossible joint angles.

Optimization Details We use Adam [11] and optimize a given sequence frame-by-frame where we use the previous output as the initialization for the current time step. For every frame, we use two optimization stages. In the first stage we optimize for all parameters $\mathbf{\Omega}$ using terms $E_{\text{J}}, E_{\text{S}}, E_{\text{spine}}, E_{\text{reg}}$. In the second stage we use the same terms, but refine the pose parameters only $(\boldsymbol{\theta}_b, \boldsymbol{\theta}_r)$.

Shape To deal with shape ambiguity that is caused by loose clothing, AGORA [17] uses Graphonomy [8] to obtain a skin-cloth segmentation. To avoid this rather time-consuming procedure, we instead disentangle the shape from the pose optimization. To do so, we first scan participants in minimal, tight-fitting clothing while they perform an easy A-pose. We then run the registration pipeline on

Figure 12: Examples of raw scans (left) and our resulting SMPL registrations (middle). (Right) Scan and SMPL registration overlaid.

this sequence to obtain the shape $\boldsymbol{\beta} \in \mathbb{R}^{10}$. Henceforth, the shape is fixed and no longer treated as an optimization parameter.

D. Body Calibration Details

In this section we provide more details on how we calibrate skin-to-sensor offsets for each subject as mentioned in Sec. 4.2 of the main paper.

D.1. EM and MVS Alignment

To compute skin-to-sensor offsets we must first spatially and temporally align the EM space with our MVS. For temporal alignment we use the Atomos Ultrasync One Box [20] to generate a timecode that we feed via LTC to our MVS and the EM sensors. Both the cameras and the EM sensors can be triggered via LTC timecode allowing for a precise temporal alignment.

For the spatial alignment we track the EM source on the lower back with an Apriltag [12, 15, 22]. For a visualization please refer to Fig. 13. We track the EM source because the origin of the EM's coordinate system is the source. The Apriltag is a square with a side length of roughly 5 cm. With 53 RGB cameras with a resolution of $4\,088 \times 3\,000$ pixels we can triangulate the Apriltag's keypoints with millimeter accuracy. Only tracking the Apriltag is however not enough to align the coordinate frame of the MVS with the coordinate frame of the EM. This is because there is a constant rigid offset between the Apriltag and the center of the EM source where the origin of the EM coordinate frame is located. We thus need to determine this offset.

To do so we track an additional S EM sensors with S Apriltags and move the sensors around for roughly 15 seconds while recording both EM and image data. This allows us to formulate an optimization procedure that solves for the rigid constant offset between the Apriltag on the source and the unknown origin of the EM coordinate system.

For a calibration sequence of T frames, let $\{(\mathbf{p}^{\mathrm{E}}_{s,t}, \mathbf{R}^{\mathrm{E}}_{s,t})\}_{t=1}^{T}$ be the position and orientation measurements for each sensor $1 \leqslant s \leqslant S$ in the EM-local coordinate system, *i.e.*, relative to the source. Further, we assume time-synchronized 6-DoF Apriltag measurements $\{(\mathbf{q}^{\mathrm{W}}_{s,t}, \mathbf{U}^{\mathrm{W}}_{s,t})\}_{t=1}^{T}$ for each sensor s in the world coordinate system, *i.e.*, the MVS' coordinate frame. We denote the measurement of the Apriltag attached to the source with index $s = 0$. Here $\mathbf{q} \in \mathbb{R}^3$ and $\mathbf{U} \in SO(3)$.

Assuming an unknown rotational offset $\mathbf{R}^{\mathrm{o}}_0 \in SO(3)$ and an unknown translational offset $\mathbf{t}^{\mathrm{o}}_0 \in \mathbb{R}^3$ that describes the offset from the Apriltag on the source to the source center, we can compute the position of that origin in world coordinates as:

$$\begin{aligned} \dot{\mathbf{U}}^{\mathrm{W}}_{0,t} &= \mathbf{U}^{\mathrm{W}}_{0,t} \cdot \mathbf{R}^{\mathrm{o}}_0 \\ \dot{\mathbf{q}}^{\mathrm{W}}_{\mathbf{0,t}} &= \dot{\mathbf{U}}^{\mathrm{W}}_{0,t} \cdot \mathbf{t}^{\mathrm{o}}_0 + \mathbf{q}^{\mathrm{W}}_{0,t} \end{aligned} \tag{10}$$

We abbreviate Eq. (10) with a general function $\sigma(\mathbf{q}, \mathbf{U}, \mathbf{t}^{\mathrm{o}}, \mathbf{R}^{\mathrm{o}})$ that applies offsets $(\mathbf{t}^{\mathrm{o}}, \mathbf{R}^{\mathrm{o}})$ to positions and orientations $(\mathbf{q}, \mathbf{U})$. This is, we re-write Eq. (10)

$$\dot{\mathbf{q}}^{\mathrm{W}}_{\mathbf{0,t}}, \dot{\mathbf{U}}^{\mathrm{W}}_{0,t} = \sigma(\mathbf{q}^{\mathrm{W}}_{0,t}, \mathbf{U}^{\mathrm{W}}_{0,t}, \mathbf{t}^{\mathrm{o}}_0, \mathbf{R}^{\mathrm{o}}_0) \tag{11}$$

Having determined the origin of the EM source in world space, *i.e.*, $(\dot{\mathbf{q}}^{\mathrm{W}}_{0,t}, \dot{\mathbf{U}}^{\mathrm{W}}_{0,t})$ we can now map all EM sensor measurements into the world:

$$\begin{aligned} \mathbf{R}^{\mathrm{W}}_{s,t} &= \dot{\mathbf{U}}^{\mathrm{W}}_{0,t} \cdot \mathbf{R}^{\mathrm{E}}_{s,t} \\ \mathbf{p}^{\mathrm{W}}_{s,t} &= \dot{\mathbf{U}}^{\mathrm{W}}_{0,t} \cdot \mathbf{p}^{\mathrm{E}}_{s,t} + \dot{\mathbf{q}}^{\mathrm{W}}_{0,t} \end{aligned} \tag{12}$$

As there is another rigid offset between the sensors' measurements in world space and the Apriltags attached to each sensor, we thus model another set of rigid rotational and translational offsets for every sensor $(\mathbf{t}^{\mathrm{o}}_s, \mathbf{R}^{\mathrm{o}}_s)$ and compute

$$\hat{\mathbf{p}}^{\mathrm{W}}_{s,t}, \hat{\mathbf{R}}^{\mathrm{W}}_{s,t} = \sigma(\mathbf{p}^{\mathrm{W}}_{s,t}, \mathbf{R}^{\mathrm{W}}_{s,t}, \mathbf{t}^{\mathrm{o}}_s, \mathbf{R}^{\mathrm{o}}_s) \tag{13}$$

With this, we formulate the objective:

$$\arg\min_{\mathcal{O}_s} \sum_{t=1}^{T} \sum_{s=1}^{S} ||\hat{\mathbf{p}}^{\mathrm{W}}_{s,t} - \mathbf{q}^{\mathrm{W}}_{s,t}||_2^2 + ||\hat{\mathbf{R}}^{\mathrm{W}}_{s,t} - \mathbf{U}^{\mathrm{W}}_{s,t}||_2^2 \tag{14}$$

where $\mathcal{O}_s = \{(\mathbf{t}^{\mathrm{o}}_s, \mathbf{R}^{\mathrm{o}}_s)\}_{s=0}^{S}$. To sufficiently constrain this optimization we use $S = 5$ sensors and move them around randomly for 15 seconds, *i.e.*, $T = 450$. The output of this spatial alignment are the offsets of the source $\mathcal{O}_0$. Note that the Apriltag is rigidly glued to the EM source, *i.e.*, this procedure must only be done once.

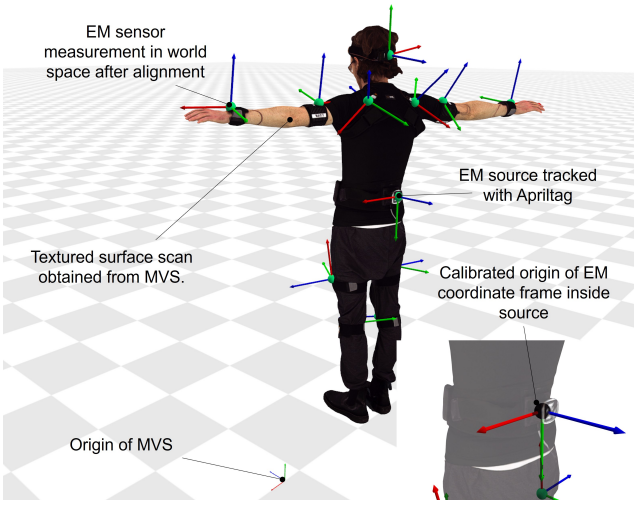

Figure 13: Visualization of coordinate frames and Apriltags involved for the body calibration procedure (see Sec. D.1).

D.2. Computing Skin-To-Sensor Offsets

With the source offsets $\mathcal{O}_0$ obtained in Sec. D.1 we can now move EM sensor measurements into the MVS' coordinate frame using Eq. (12). This and our SMPL registration pipeline described in Sec. C allows us to compute skin-to-sensor offsets $\mathbf{o}_s = (\mathbf{v}_s, \mathbf{Q}_s)$ which are required for EMP's stage 1.

To do so, we first define anchor points parameterized as a position $\tilde{\mathbf{p}}_s$ and orientation $\tilde{\mathbf{R}}_s$ on the SMPL mesh. The position $\tilde{\mathbf{p}}_s$ is simply the position of a hand-picked vertex and the orientation $\tilde{\mathbf{R}}_s$ can be constructed using any adjacent vertex and the corresponding vertex normal. Note that manually picking those anchor points on the SMPL mesh must only be done once.

Next, we take the registered SMPL meshes of a short 3-second calibration sequence and apply unknown offsets $(\mathbf{v}_s, \mathbf{Q}_s)$ to the anchor points to obtain virtual sensor orientations $\mathbf{R}^v_{s,t} = \tilde{\mathbf{R}}_{s,t}\mathbf{Q}_s$ and virtual sensor positions $\mathbf{p}^v_{s,t} = \tilde{\mathbf{R}}_{s,t}\mathbf{v}_s + \tilde{\mathbf{p}}_{s,t}$. We then equate the virtual measurements to the real measurements (which have been rotated to world space with $\mathcal{O}_0$) and optimize for the sensor offsets:

$$\underset{\mathbf{v}_s, \mathbf{Q}_s}{\arg\min} \sum_{t=1}^{T} ||\mathbf{p}^{\mathrm{W}}_{s,t} - \mathbf{p}^v_{s,t}||_2^2 + ||\mathbf{R}^{\mathrm{W}}_{s,t} - \mathbf{R}^v_{s,t}||_2^2 \qquad (15)$$

The output of this optimization are subject-specific skin-to-sensor offsets $\{\mathbf{o}_s\}_{s=1}^{S}$ which we compute for every participant and every capture session.

E. Stage 3 Smoothing

As mentioned in the main paper in Sec. 6.1, we perform a light smoothing pass on the outputs obtained by minimizing E_{S3}. Because this only requires a light adjustment, it does not break pose-to-image alignment. Note that the same could not be said if we were to simply omit stage 3 and smooth the outputs of stage 2. To achieve the same reduction in jitter by smoothing stage 2, a more aggressive smoothing pass is required, which will lead to misalignments, especially in fast motions. We show and discuss such a case in Fig. 14.

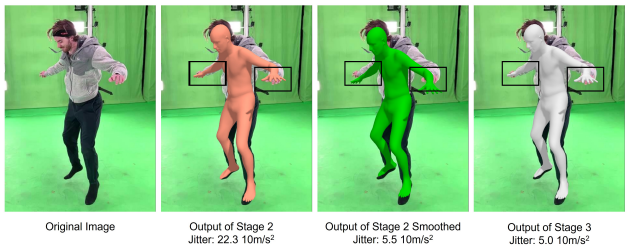

Figure 14: Effect of smoothing. The output of stage 2 (second to left) is jittery, but in this time instance well aligned with the image. Smoothing the output of stage 2 breaks this alignment (second to right). This is not the case for the outputs of stage 3 (right). Here, the pose is well aligned with the image observations while having a similar level of jitter as the naïvely smoothed outputs of stage 2.

In the smoothing pass, we smooth the SMPL parameters $\boldsymbol{\theta}_r, \boldsymbol{\theta}_b, \mathbf{t}$ using a Savitzky-Golay filter with a window length of 7 and a second order polynomial. For the translation $\mathbf{t}$ we can directly apply the filter. For the SMPL body and root orientations, we first convert them into quaternions and apply the filter to each coordinate of the quaternions separately. For this to work, it is important to ensure that the quaternions are continuous because the quaternion q represents the some rotation as the quaternion $-q$. Hence, we first make sure the sign of a quaternion does not flip within a given sequence before we apply the filter. After the filter is applied, we normalize the quaternions to ensure valid rotations and convert them back to the angle-axis format.

F. Visual Comparison to 3DPW

In the main paper we compare quantitatively to 3DPW [21] (see Sec. 6.1). Here we also show a few visual comparisons (see Fig. 15). To do so we recorded a similar motion sequence where the participant is walking around poles and is briefly occluded. In Fig. 15 we observe higher fidelity SMPL fits in our results.

G. Fine-tuning with EMDB

We fine-tune an existing human pose estimation method with EMDB and investigate how this influences the performance on 3DPW (see Tab. 6). For this example, we use ROMP [19] and their publicly available code. The first row

Figure 15: Visual comparison to 3DPW [21] on a similar sequence. We observe sometimes large image-to-pose misalignments (top left) as well as unrealistic poses (top right) in 3DPW. In contrast, we provide better alignment and more accurate poses for similar (bottom left) or even higher levels of occlusion (bottom right).

Method	MPJPE	PA-MPJPE
ROMP (HRNet, w/o 3DPW)	83.9	54.1
ROMP + EMDB fine-tuning	**80.8**	**52.6**

Table 6: Effect of fine-tuning ROMP [19] on EMDB, evaluated on the test set of 3DPW, using ROMP's official, pre-trained model.

in Tab. 6 reports the result on 3DPW of their pre-trained model that has never seen 3DPW before. We observe that the joint errors on 3DPW decrease after fine-tuning this model with EMDB, which further higlights the usefulness of EMDB.

H. Evaluation of Global Trajectories

H.1. Camera Trajectory

In Sec. 6.2 of the main paper we measure the accuracy of the iPhone's self-localized 6D poses. To do so, we attach an Apriltag rigidly to the iPhone and record both iPhone poses and Apriltag 6D poses on our MVS. This allows us to compare the iPhone's poses with the Apriltag tracking. However, the former are in the iPhone's own coordinate system, while the latter are relative to the MVS' tracking space (because we triangulate the Apriltag with the known calibration of the MVS). In addition, there is a constant rigid offset between the iPhone's sensor origin and the Apriltag. We thus solve an optimization problem to align the two spaces, which is explained in the following. Note that this problem is very similar to the optimization we run to align the EM space with the MVS as described in Sec. D.1.

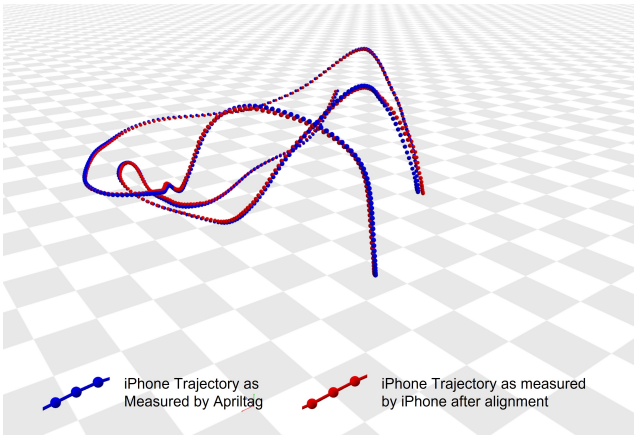

Figure 16: Visualization of optically tracked (blue) and self-localized (red) iPhone trajectory.

We are given triangulated Apriltag positions $\mathbf{q}_t^{\mathrm{W}} \in \mathbb{R}^3$ and orientations $\mathbf{U}_t^{\mathrm{W}} \in SO(3)$ in the MVS' coordinate system (here the world) and iPhone positions $\mathbf{p}_t^{\mathrm{i}} \in \mathbb{R}^3$ and orientations $\mathbf{R}_t^{\mathrm{i}} \in SO(3)$ in the iPhone's coordinate frame.

We first move the iPhone's 6D pose into the world with an unknown rigid transformation $\mathbf{T}^{\mathrm{i}\to\mathrm{W}} = \left[\mathbf{R}^{\mathrm{i}\to\mathrm{W}} \mid \mathbf{t}^{\mathrm{i}\to\mathrm{W}}\right]$ to obtain

$$\begin{aligned} \mathbf{p}_t^{\mathrm{W}} &= \mathbf{T}^{\mathrm{i}\to\mathrm{W}} \cdot \mathbf{p}_t^{\mathrm{i}} \\ \mathbf{R}_t^{\mathrm{W}} &= \mathbf{R}^{\mathrm{i}\to\mathrm{W}} \mathbf{R}_t^{\mathrm{i}} \end{aligned} \tag{16}$$

Next, we model an unknown translational and rotational offset $(\mathbf{t}^{\mathrm{o}}, \mathbf{R}^{\mathrm{o}})$ to account for the constant rigid offset between the Apriltag measurement and the iPhone's pose in the world space, *i.e.*, $\hat{\mathbf{p}}_t^{\mathrm{W}}, \hat{\mathbf{R}}_t^{\mathrm{W}} = \sigma(\mathbf{p}_t^{\mathrm{W}}, \mathbf{R}_t^{\mathrm{W}}, \mathbf{t}^{\mathrm{o}}, \mathbf{R}^{\mathrm{o}})$ where σ is the function defined in Sec. D.1. We then compare the estimated Apriltag pose with the actual Apriltag measurement to minimize:

$$\underset{\mathbf{T}^{\mathrm{i}\to\mathrm{W}}, \mathbf{t}^{\mathrm{o}}, \mathbf{R}^{\mathrm{o}}}{\arg\min} \sum_{t=1}^{T} ||\hat{\mathbf{p}}_t^{\mathrm{W}} - \mathbf{q}_t^{\mathrm{W}}||_2^2 + ||\hat{\mathbf{R}}_t^{\mathrm{W}} - \mathbf{U}_t^{\mathrm{W}}||_2^2 \tag{17}$$

The objective value after this optimization is the alignment error we report in Sec. 6.2 (*iPhone Pose Accuracy*) of the main paper. For a visualization of the aligned trajectories, please refer to Fig. 16.

H.2. SMPL Root Trajectory

We proceed similarly as described in Sec. H.1 to compute the error of the SMPL root trajectory estimated by EMP to ground-truth SMPL root trajectories obtained with the MVS. Note that we cannot simply re-use the transformation $\mathbf{T}^{\mathrm{i}\to\mathrm{W}}$ found in that section. This is because the

iPhone's coordinate system changes with every new recording. In addition, evaluation takes with our MVS have fewer iPhone movements so as to not obstruct the MVS' cameras. This means that Eq. (17) tends to be underconstrained and thus the optimization does not always converge to meaningful solutions.

To address this, we add another term to Eq. (17) in which we move the SMPL root joint position predicted by EMP, $\mathbf{r}_t^{\text{i}}$, into the world frame and then compare it with the SMPL root joint given by our ground-truth registration, $\mathbf{r}_t^{\text{W}}$. This is, we compute $\hat{\mathbf{r}}_t^{\text{W}} = \mathbf{T}^{\text{i}\to\text{W}} \cdot \mathbf{r}_t^{\text{i}}$ and add the term $||\hat{\mathbf{r}}_t^{\text{W}} - \mathbf{r}_t^{\text{W}}||_2^2$ to Eq. (17). This effectively removes a global rigid misalignment between estimated and ground-truth SMPL root trajectories. The remaining Euclidean distance between $\hat{\mathbf{r}}_t^{\text{W}}$ and $\mathbf{r}_t^{\text{W}}$, is the alignment error we report in the main paper in Sec. 6.2 (*Global SMPL Trajectories*).

I. EMP Implementation Details

We include implementation details of EMP's three stages here, as well as rough runtime estimates. We use PyTorch [16] for all computations.

In stage 1, we first only optimize for the SMPL root parameters $\boldsymbol{\theta}_r, \mathbf{t}$ to get a rough alignment of the SMPL body to the sensor cloud. In a second pass we then optimize for all SMPL parameters, *i.e.*, including $\boldsymbol{\theta}_b$. For both passes, we use an L-BFGS [13] optimizer with a learning rate of 1.0 and strong Wolfe line search. We iterate the line search 20 times and take 5 steps with the L-BFGS optimizer. The remaining hyperparameters are chosen as $\lambda_{\text{p}} = 1.0, \lambda_{\text{r}} = 1.0, \lambda_{\text{bp}} = 1.0e^{-5}, \lambda_{\text{rec}} = 1.0$. This stage typically finishes in 1-2 minutes as we can use large batch sizes (the entire sequence fits into a single batch on a 24 GB GPU).

Stage 2 is a sequential optimization, where we optimize for each frame given the previous as initialization. In this stage we use the Adam optimizer [11] with a learning rate of 0.01. We optimize for 100 iterations in each frame. The hyperparameters are set to $\lambda_{\text{2D}} = 0.01, \lambda_{\text{rec}} = 1.0, \lambda_{\text{prior}} = 1.0, \lambda_{\text{pcl}} = 10.0$. Each frame's optimization takes approx. 5 seconds, so optimization of a typical sequence of 45 seconds length finishes in roughly 2 hours.

In stage 3 we fit a neural implicit human model. We also use the Adam optimizer [11] with a learning rate of $5.0e^{-4}$. The learning rate decays to half after 200 and 500 epochs respectively. The hyperparameters are chosen as $\lambda_{\text{rgb}} = 1.0, \lambda_{\text{eik}} = 0.1, \lambda_{\text{reg}} = 10.0$. The hyperparameters for the scene decomposition loss follow the same setting as in V2A [9]. Stage 3 is computationally the heaviest. We train the model for 48 hours on a single 24 GB GPU. We split very long sequences into several subparts and train each part in parallel on several GPUs in order to increase convergence speed.

J. Socetial Impact

Extracting human body shape and pose from imagery or other sensory data is an important building block in the endeavor to understand human behavior with computational methods. Having an accurate system available promises valuable applications such as immersive remote telepresence (thus saving CO2-intensive travel), automated rehabilitation (*e.g.*, for the recovery of post-stroke patients), or computer-guided fitness and health coaches, to name just a few. All these applications directly benefit society, be it to provide more cost-effective treatments in medicine, smart tools to improve personal health, or ways to reduce our environmental footprint.

While this work does not directly improve human pose estimation methods, it fills an important gap: the availability of paired input sensor data and 3D human pose. This in turn will allow other researchers in the field to improve pose estimators by either using our dataset as training data or as a new evaluation benchmark. The evolution of Deep Learning in recent years has shown that the availability of data is a prime contributor to advancements in the field. Hence, we expect our dataset to lead to knowledge advancement in the field, which in turn will enable more sophisticated technical applications.

Human pose estimation methods, specifically so from images, might be abused for malicious surveillance or person identification via gait analysis or face recognition. Although EMDB does not publish any identifiable information or directly propose improved pose estimators, future advancements in pose estimation directly imply that adverse uses of such technology automatically benefit, too. This presents an ethical and societal concern, which must be considered in future developments of such technology. Yet, given the promising technical applications of accurate human body pose estimators, we feel that the benefits of this research clearly outweighs the risks.

References

[1] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 3

[2] Jonathan T. Barron. A general and adaptive robust loss function. *CVPR*, 2019. 3

[3] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer, aug 2020. 3

[4] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 3

[5] 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. 3

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016. 3

[7] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34(4), jul 2015. 1

[8] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *CVPR*, 2019. 3

[9] Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[10] Manuel Kaufmann, Yi Zhao, Chengcheng Tang, Lingling Tao, Christopher Twigg, Jie Song, Robert Wang, and Otmar Hilliges. Em-pose: 3d human pose estimation from sparse electromagnetic trackers. In *International Conference on Computer Vision (ICCV)*, 2021. 1

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 3, 7

[12] Maximilian Krogius, Acshi Haggenmiller, and Edwin Olson. Flexible layouts for fiducial tags. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2019. 1, 4

[13] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1–3):503–528, aug 1989. 7

[14] 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, Oct. 2015. 1, 3

[15] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011. 1, 4

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 3, 7

[17] Priyanka Patel, Chun-Hao P. Huang, Joachim Tesch, David T. Hoffmann, Shashank Tripathi, and Michael J. Black. AGORA: Avatars in geography optimized for regression analysis. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 3

[18] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 3

[19] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *ICCV*, 2021. 5, 6

[20] *Atomos Ultrasync One*, 2023. https://www.atomos.com/accessories/ultrasync-one. 4

[21] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018. 5, 6

[22] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016. 1, 4

[23] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 3