Joint Optimization for Multi-Person Shape Models from Markerless 3D-Scans: Supplementary Materials

Samuel Zeitvogel[1], Johannes Dornheim[1], and Astrid Laubenheimer[1]

Intelligent Systems Research Group (ISRG), Karlsruhe, Germany
{samuel.zeitvogel,johannes.dornheim,astrid.laubenheimer}@hs-karlsruhe.de

1 Symbol Tables

Symbol tables for the paper are available in Section 8.

2 Subdivision Surface Implementation Details

For each face in the mesh, we construct 16 control points for the limit surface and 12 control points for two tangent vector fields. We denote the extraction of the control points by the very sparse matrices $L_f \in \mathbb{R}^{N\times 16}$ and $L'_f, L''_f \in \mathbb{R}^{N\times 12}$ for all faces $1 \leq f \leq F$. These matrices only depend on the mesh connectivity and for a detailed review of their structure, we refer to [5]. $U \in \mathbb{R}^{N\times 3}$ denotes the matrix representation of $\boldsymbol{T}'$ where the columns contain the x,y and z coordinates of $\boldsymbol{T}'$. The respective limit control points are computed via

$$\eta^{(f)} = U^T L_f, \quad \upsilon^{(f)} = U^T L'_f, \quad \xi^{(f)} = U^T L''_f. \tag{1}$$

The surface point and the pair of tangent vectors are computed using

$$S_f(s,t;\boldsymbol{T}') = \sum_{j=0}^{3}\sum_{k=0}^{3} A_j^3(s)A_k^3(t)\eta_{4j+k}^{(f)}, \tag{2}$$

$$\frac{\partial}{\partial s}S_f(s,t;\boldsymbol{T}') = \sum_{j=0}^{2}\sum_{k=0}^{3} A_j^2(s)A_k^3(t)\upsilon_{4j+k}^{(f)}, \tag{3}$$

$$\frac{\partial}{\partial t}S_f(s,t;\boldsymbol{T}') = \sum_{j=0}^{3}\sum_{k=0}^{2} A_j^3(s)A_k^2(t)\xi_{4k+j}^{(f)}, \tag{4}$$

where $\eta_j^{(f)}$, $\upsilon_j^{(f)}$ and $\xi_j^{(f)}$ denote the j-th column of the matrices and the Bernstein basis polynomial is definded as

$$A_\mu^\nu(x) = \binom{\nu}{\mu} x^\mu (1-x)^{\nu-\mu}. \tag{5}$$

Fig. 1: Visualization of our prescribed mean shape $\bar{\boldsymbol{T}}^{\text{init}}$ with underlying subdivision surface from front (left) and back (right) view. The shape is manually split into $K+1$ body parts (denoted by the different coloring) which leads to K joints. The initial skinning weights $\mathcal{W}^{\text{init}}$ are initialized using the body part segmentation. A vertex in the interior of a body part is set to 1 for the corresponding joint. Skinning weights for vertices on the body part boundary are set to 0.5 for the corresponding joint pair. The remaining entries in the skinning weight matrix are set to 0. Vertex indices on the body part boundary are denoted by the sets Ring_k, where k denotes the corresponding joint index.

Following this approach we can compute the surface unit normal at a point $u = (s, t, f)$ with

$$S_f^{\perp} = \frac{\frac{\partial}{\partial s} S_f \times \frac{\partial}{\partial t} S_f}{||\frac{\partial}{\partial s} S_f \times \frac{\partial}{\partial t} S_f||_2}. \tag{6}$$

With the definitions for the surface and normal field of each surface patch the whole surface S and normal field $S^{\perp}$ is defined as

$$S(u; \boldsymbol{T}') = S((s, t, f); \boldsymbol{T}') = S_f(s, t; \boldsymbol{T}'), \tag{7}$$

$$S^{\perp}(u; \boldsymbol{T}') = S^{\perp}((s, t, f); \boldsymbol{T}') = S_f^{\perp}(s, t; \boldsymbol{T}'). \tag{8}$$

3 Model Initialization

The used mesh template with a segmentation into body parts is depicted in Fig. 1. The initial joints are automatically positioned in the center of all vertices that are on the boundary of a body part. Our OpenPose-to-surface correspondences are depicted in Fig. 2.

4 Ablation Study

The shape blend-shapes are highly correlated. This is not surprising, as using multiple shape blend-shapes for coarse variations like body height leads to a

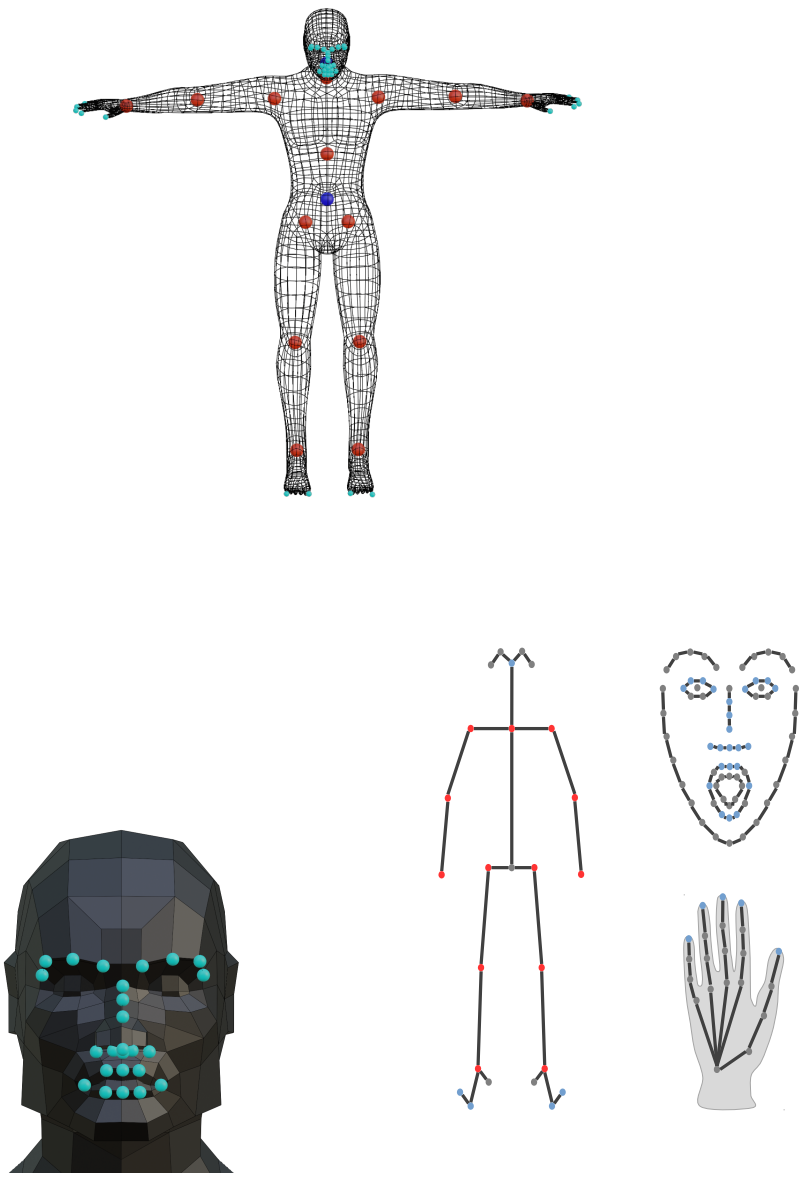

Fig. 2: Visualization of the model skeleton and OpenPose label correspondences. Top: Skeleton joints are depicted in blue and red. Only the red joints are used in $E_{\text{2D-joint}}$. Surface correspondences used in $E_{\text{2D-surf}}$ are depicted in cyan. Left: Surface correspondences on the face. Right: OpenPose keypoints. The red keypoints are used for $E_{\text{2D-joint}}$ and the cyan keypoints are used by $E_{\text{2D-surf}}$. The remaining keypoints are not used.

lower error in E_{bshape} and E_{shape}. We depict the covariance matrix of the trained shape blend-shapes $\mathcal{S}$ after training the unisex model in Fig. 3.

Side-by-side comparison of our model with SMPL: We fit our trained unisex model to two animated sequences[1] of SMPL. This results in two shape estimates $\vec{\beta}$ and two sequences of pose estimates. We provide a side-by-side comparison in Fig. 4 and Fig. 5.

We showcase the factorization of shape and pose by generating 5 different subjects and using the same pose parameters for each subject. In Fig. 6, 3 different poses are depicted. The effect of the pose blend-shapes are depicted in Fig. 7.

We track the triangle area expansion and contraction of our model and SMPL during an animated sequence. We fit our model to a SMPL sequence and compare the local surface deformation behavior. The local area change is depicted in Fig. 8.

We modify the weights of our sum of squares error function E drastically to highlight the importance of selected terms. The failure cases are depicted in Fig 9.

	0	1	2	3	4	5	6	7	8	9
0	0.87	-0.47	0.3	0.67	-0.36	-0.89	0.54	0.21	-0.13	-0.52
1	-0.47	0.4	-0.098	-0.25	0.21	0.37	-0.19	-0.11	0.077	0.16
2	0.3	-0.098	0.52	0.35	0.13	-0.26	0.11	-0.14	0.24	-0.14
3	0.67	-0.25	0.35	1.4	0.018	-0.14	0.56	-0.11	-0.35	-0.66
4	-0.36	0.21	0.13	0.018	0.7	0.97	-0.31	-0.61	0.041	0.2
5	-0.89	0.37	-0.26	-0.14	0.97	2.5	-0.92	-1.1	-0.16	0.82
6	0.54	-0.19	0.11	0.56	-0.31	-0.92	0.65	0.33	-0.23	-0.62
7	0.21	-0.11	-0.14	-0.11	-0.61	-1.1	0.33	0.74	0.061	-0.26
8	-0.13	0.077	0.24	-0.35	0.041	-0.16	-0.23	0.061	0.55	0.21
9	-0.52	0.16	-0.14	-0.66	0.2	0.82	-0.62	-0.26	0.21	0.71

Fig. 3: Covariance matrix of the shape blend-shapes of our unisex model without orthogonalization.

5 Optimization Implementation Details

Skinning Weights: We constrain the skinning weights to $[0, 1]$ and parameterize all skinning weights using the nonlinear squashing function

$$w_{k,i} = \frac{1}{2} + \frac{\tilde{w}_{k,i}}{2\sqrt{1 + \tilde{w}_{k,i}^2}} \tag{9}$$

which constrains the new variables $\tilde{w}_{k,i} \in \mathbb{R}$ to the range $w_{k,i} \in (0, 1)$ for all K joints and all N vertices.

[1] `smpl.is.tue.mpg.de`

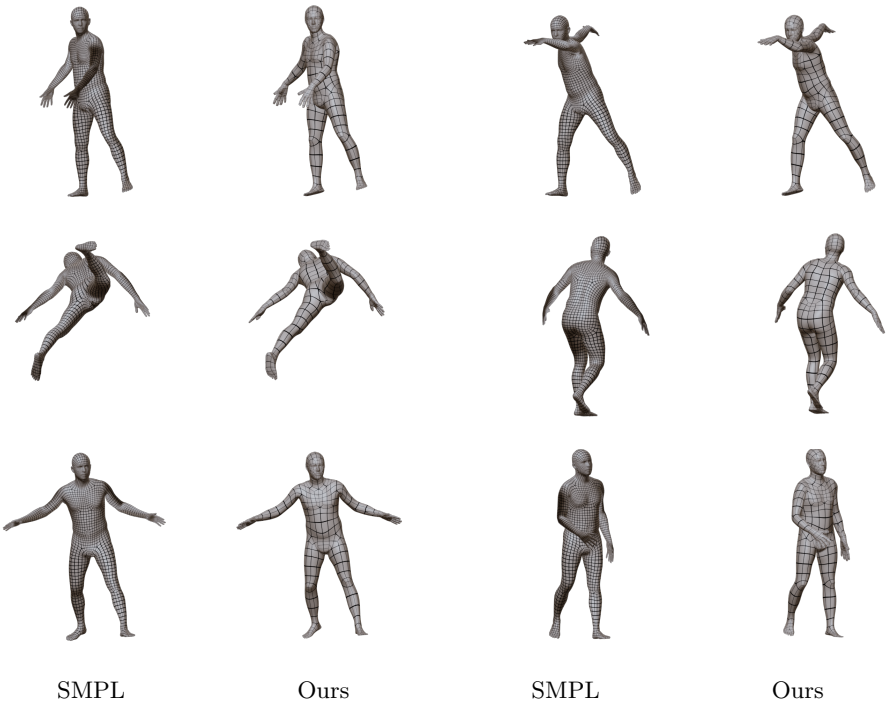

Fig. 4: Side-by-side comparison of the male SMPL model and our proposed unisex model. A video for the whole sequence is also available in the supplementary material.

Fig. 5: Side-by-side comparison of the female SMPL model and our proposed unisex model. A video for the whole sequence is also available in the supplementary material.

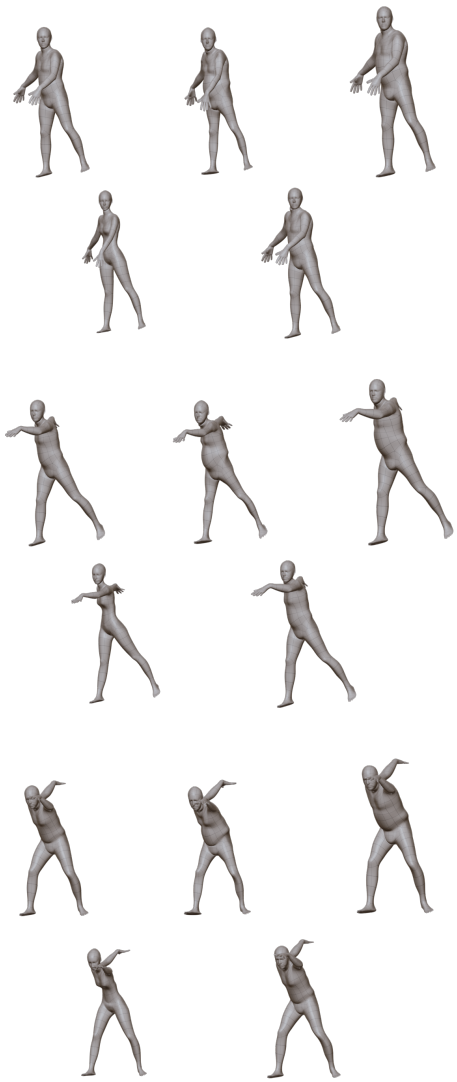

Fig. 6: Five different subjects are depicted in three different poses. A video for the whole sequence is also available in the supplementary material.

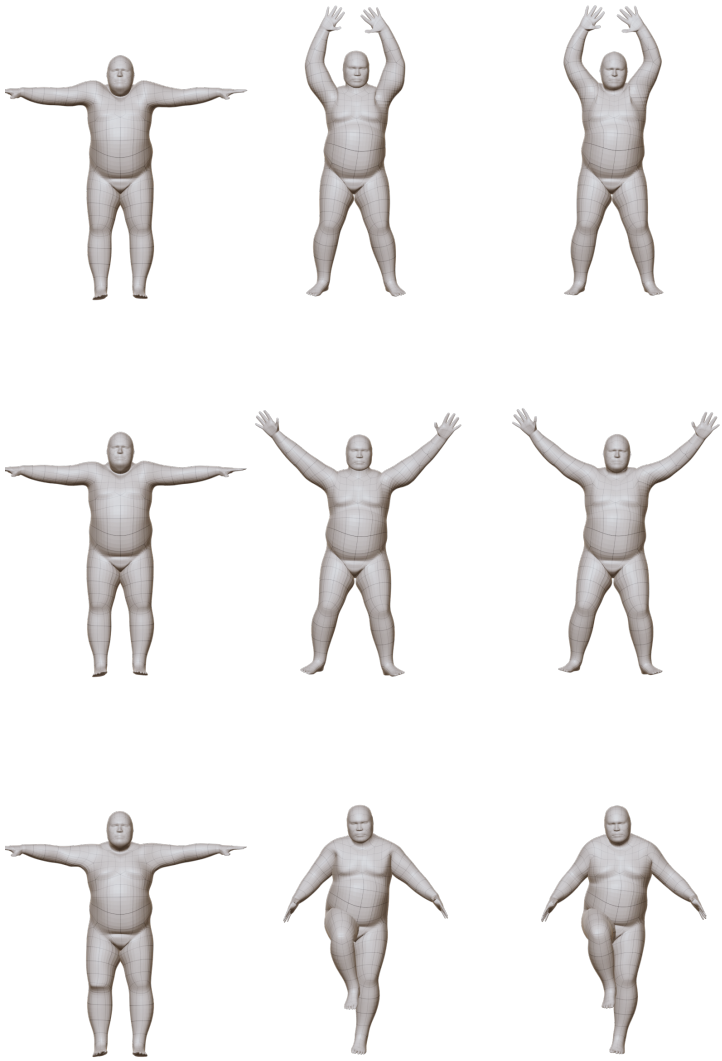

Fig. 7: Pose blend-shapes before linear blend skinning (left column), with linear blend skinning (right column). Linear blend skinning without any pose blend-shapes is depicted in the center column. The depicted poses are part of the Dynamic FAUST dataset [1] (used for our model training). A video for the whole sequence is also available in the supplementary material.

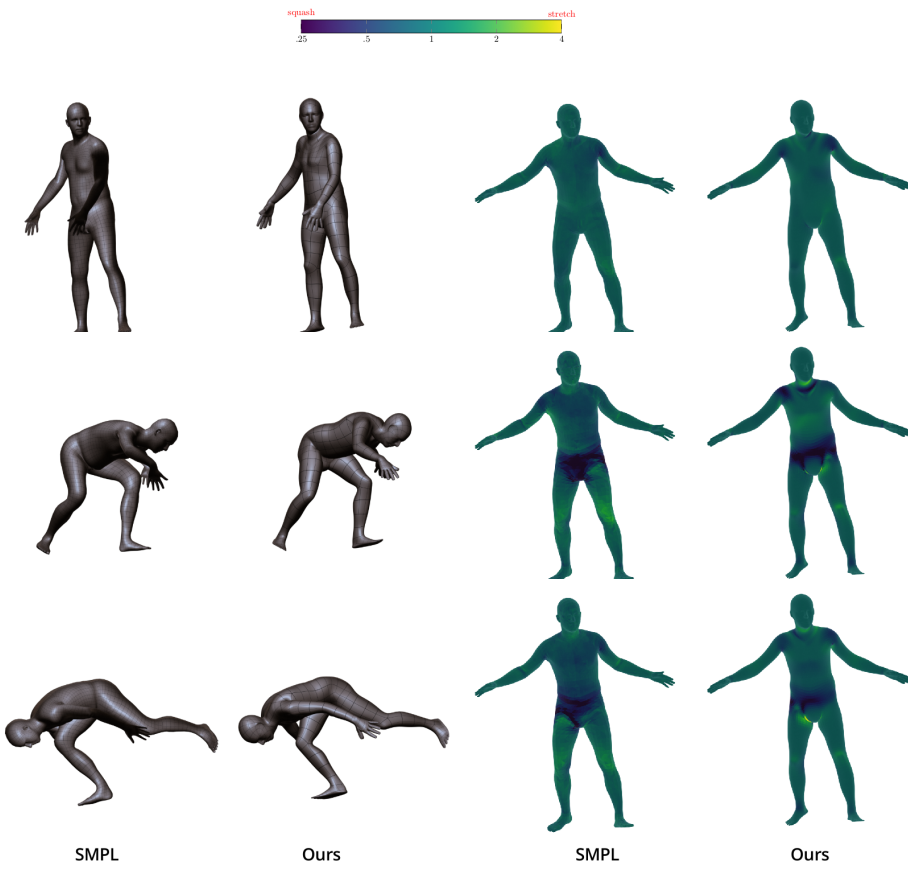

Fig. 8: Triangle mesh area visualization during changes in pose. Left: Animation of SMPL and our model fitted to SMPL. Right: Each triangle is colored by the area change of the triange with respect to the depicted reference pose. A video for the whole sequence is also available in the supplementary material.

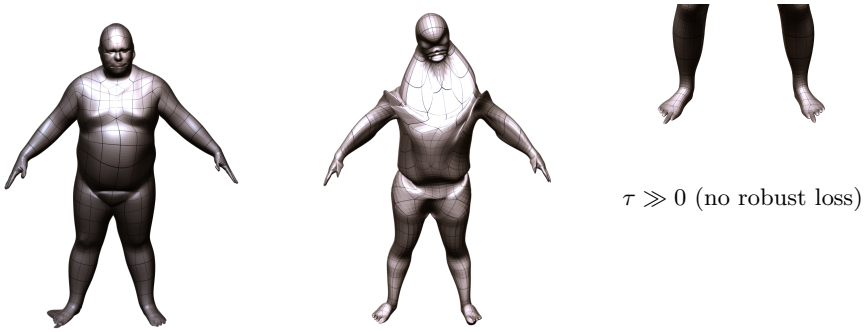

Fig. 9: Failure cases during training.

Rotation: We optimize on $SO(3)$ by linearizing at our current estimation and projecting the update step back onto $SO(3)$ using the exponential map that can be computed in closed form using the Rodrigues formula [2].

Surface Points: To optimize the surface correspondences u we follow the approach outlined in [6] where the surface is linearized at $S(u, \boldsymbol{T}')$ to get the update with respect to the tangent plane at $S(u, \boldsymbol{T}')$. The tangent plane exists everywhere due to the C^2-continuity property of Catmull-Clark subdivision surfaces that is valid almost everywhere. To project the update step back onto the surface we move the update in each patch and transform with respect to each new patch that is crossed. This update procedure is computed on the current surface estimate before it is updated.

Discrete Correspondence Updates: We interleave the joint optimization of surface correspondences with discrete updates to jump *e.g.* across fingers. We employ a combination of two different discrete update strategies:

I We sample points on the model surface S, store them in a fast temporary data structure, query the data point $\boldsymbol{p}$ and set the closest point to the closest surface point if it is closer than the current estimate. This mirrors the NICP correspondence update step.

II Given a measured point $\boldsymbol{p}$ and a corresponding normal $\boldsymbol{n}$ one fast way to approximate the nearest neighbor is using ray-casting where we seek the intersection point of the current surface S and the ray originating from $\boldsymbol{p}$ in normal direction $\boldsymbol{n}$. We reject points where the model surface normal $S^{\perp}$ and data $\boldsymbol{n}$ span an angle that is larger than 90 degrees. We use an off-the-shelf implementation from the ray-tracing kernel collection Embree [7] which implements fast ray to Catmull-Clark subdivision surface intersections.

We disable continuous surface optimization during the first few iterations and use discrete update strategy (II) at each iteration. After a few steps, we

enable continuous surface optimization and switch to discrete update strategy (I) every 5th iteration to enable hops of correspondences between distant body parts (*e.g.* fingers and legs).

Robust Least Squares: To optimize the robust cost function ϕ with off-the-shelf nonlinear least squares solvers, we follow the approach outlined in [8] by lifting the robust kernel.

An additional parameter ψ is introduced per data point as well as one term per data point. Following [9] for the robust Geman-McClure error function the new lifted cost is

$$E_{\text{robust}} = \sum_{(\psi,\boldsymbol{p},u,R,\boldsymbol{m},\vec{\beta})\in\mathcal{D}'} \left(\psi^2||S(u,M(R,\boldsymbol{m},\vec{\beta};\Theta))-\boldsymbol{p}||^2+\tau^2(\sqrt{\psi^2}-1)^2\right), \quad (10)$$

where $\tau \in \mathbb{R}^+$ relates to the radius of the kernel and $\mathcal{D}' \subset \mathbb{R}\times\mathcal{D}$. This formulation is again a nonlinear least squares objective. We anneal τ during our optimization, starting with $\tau \gg 0$ and lowering τ as the optimization progresses.

Function Splitting: Optlang cannot handle the long residual term expression E_{Data}. To simplify the expression we split the residuals into many subexpressions with the introduction of additional latent variables and regularization terms.

In general, let us assume we have the following optimization problem

$$\arg\min_X ||h(g(X))||_F^2, \quad (11)$$

then we introduce a new optimization problem

$$\arg\min_{X,Y} ||h(Y)||_F^2 + \lambda||Y-g(X)||_F^2, \quad (12)$$

where the function chain is replaced by a simpler term and a coupling term $\lambda \in \mathbb{R}^+$ through the introduction of additional latent variables Y. This approach is commonly used in alternating registration and model training to split the model-regularized nonrigid registration and model parameter updates [4]. The introduced latent variables are referred to as "registrations" in this context. In practice λ cannot be set to an arbitrarily high value due to numerical instabilities. In contrast, we use this method to simplify individual error terms and enable large scale joint optimization. The downside of this approach is the introduction of additional variables. A split of Eq. (1) of the paper would introduce $3Nn$ scalar variables denoted by ${}_1\boldsymbol{T}, {}_2\boldsymbol{T}, \ldots, {}_n\boldsymbol{T} \in \mathbb{R}^{3N}$. The new regularization term would be

$$\lambda_{\boldsymbol{T}} \sum_{({}_i\boldsymbol{T},\beta,R,\boldsymbol{m})} ||\bar{\boldsymbol{T}} + \mathcal{S}\vec{\beta} + \mathcal{P}\,\text{vec}(R_1 - I, R_2 - I, \ldots, R_K - I) - {}_i\boldsymbol{T}||^2 \quad (13)$$

and in the model formulation we replace $\boldsymbol{T}$ by ${}_i\boldsymbol{T}$ for all $1 \le i \le n$. In total, we split the data term at the following subexpressions of $S(u, \boldsymbol{T}')$:

1. Eq. (1) of the paper: $\boldsymbol{T}$ introduces $3Nn$ variables,
2. Eq. (2) of the paper: $\boldsymbol{J}$ introduces $3KQ$ variables,
3. Eq. (5) of the paper: $\boldsymbol{T}'$ introduces $3Nn$ variables and
4. SupMat Eq. 1: $\eta^{(f)}$, $\upsilon^{(f)}$ and $\xi^{(f)}$ introduce $nF(16 + 12 + 12)$ variables.

Optimization on the GPU: As mentioned, we solve the nonlinear least squares objective using the Optlang framework [3]. In particular, we use the modified version introduced in [10] that supports optimization on $SO(3)$ and on surface points that live on Catmull-Clark subdivision surfaces. We choose the Levenberg-Marquardt solver with an inexact, iterative preconditioned conjugate gradient implementation that is used to solve the damped normal equations that arise in Levenberg-Marquardt during each optimization step. We use diagonal Jacobi-preconditioning that is implemented in Optlang.

6 Hyperparameter Settings

The hyperparameter settings are listed in Table 1. The subject-specific shape regularization is different for each subject and denoted by $\lambda_{\text{shape},i}$. The number of scans for the i-th subject is denoted by n_i. At Levenberg-Marquardt iteration 0, 15, 50 and 75 we modify the weighting. The weights were refined during the development of our approach in an iterative, incremental fashion. We do not claim that these parameter settings are optimal or that the parameter dependencies work for arbitrary object classes and dataset modalities. Future work is required to find a more systematic approach for these parameter settings.

7 Detailed Experimental Results

An anonymized in-depth inspection of the FAUST correspondence challenge is possible on the dataset website. JOMS (**J**oint **O**ptimization for Multi-Person Shape Models from **M**arkerless 3D-**S**cans) results can be inspected for the intra-subject (JOMS-intra) and the inter-subject (JOMS-inter) challenge.

Parameter	Iter 0	Iter 15	Iter 50	Iter 75
λ_{Data}	0	0	30	30
$\lambda_{\text{2D-surf}}$	$\frac{P}{4L}$	$\frac{P}{4L}$	$\frac{P}{256L}$	$\frac{P}{256L}$
$\lambda_{\text{2D-joint}}$	$\frac{P}{4L}$	$\frac{P}{4L}$	$\frac{P}{256L}$	$\frac{P}{256L}$
λ_{mean}	$\frac{nP}{F(B+1)}$	$\frac{nP}{F(B+1)}$	$\frac{4nP}{9F(B+1)}$	$\frac{nP}{9F(B+1)}$
λ_{bshape}	$\frac{nP}{F(B+1)}$	$\frac{nP}{F(B+1)}$	$\frac{4nP}{9F(B+1)}$	$\frac{nP}{9F(B+1)}$
λ_{pshape}	$\gg 0$	$\gg 0$	$\frac{10nP}{9F(B+1)}$	$\frac{10nP}{36F(B+1)}$
λ_{joint}	$\frac{25nP}{(B+1)K}$	$\frac{25nP}{(B+1)K}$	$\frac{25nP}{(B+1)K}$	$\frac{1}{100}$
λ_{weights}	100	100	100	0
λ_{convex}	$\frac{100n^2P^2}{N^2}$	$\frac{100n^2P^2}{N^2}$	$\frac{100n^2P^2}{N^2}$	$\frac{100n^2P^2}{N^2}$
$\lambda_{\text{shape},i}$	$\frac{18^2Pnn_i}{QB}$	$\frac{9Pnn_i}{QB}$	$\frac{Pnn_i}{4QB}$	$\frac{Pnn_i}{16QB}$
λ_{pose}	$\frac{0.0151^2P}{K}$	$\frac{0.0151^2P}{K}$	$\frac{0.0151^2P}{K}$	0
λ_{ground}	0	0	$\frac{(100L)^2}{N^2}$	$\frac{(100L)^2}{N^2}$
λ_{symm}	$\frac{100nP}{(B+1)N}$	$\frac{100nP}{(B+1)N}$	$\frac{100nP}{(B+1)N}$	$\frac{100nP}{(B+1)N}$

Table 1: Hyperparameter setting changes at different Levenberg-Marquardt iterations.

8 Symbol Tables

Model

$\Theta = (\bar{\boldsymbol{T}}, \mathcal{S}, \bar{\boldsymbol{J}}, \mathcal{J}, \mathcal{P}, \mathcal{W})$	aggregated model parameters
$\vec{\beta} \in \mathbb{R}^B$	subject specific blend-shape coefficients
$(R, \boldsymbol{m}) \in SO(3)^{K+1} \times \mathbb{R}^3$	pose specific parameters
$\bar{\boldsymbol{T}} \in \mathbb{R}^{3N}$	mean shape
$\boldsymbol{T}$	shape after blend-shape deformations
$\boldsymbol{t}_i \in \mathbb{R}^3$	i-th vertex in $\boldsymbol{T}$
$\boldsymbol{T}'$	shape after linear blend skining
$\boldsymbol{t}'_i \in \mathbb{R}^3$	i-th vertex in $\boldsymbol{T}'$
$\mathcal{S} \in \mathbb{R}^{3N \times B}$	shape blend-shapes
$\mathcal{P} \in \mathbb{R}^{3N \times 9K}$	corrective pose blend-shapes
$\bar{J} \in \mathbb{R}^{3K}$	mean skeleton joint locations corresponding to $\bar{\boldsymbol{T}}$
$\mathcal{J} \in \mathbb{R}^{3K \times B}$	skeleton basis shapes corresponding to $\mathcal{S}$
$\boldsymbol{J} \in \mathbb{R}^{3K}$	skeleton joint locations after skeleton basis shapes application
$\mathcal{J}_k^{(b)} \in \mathbb{R}^3$	position of the k-th joint of the b-th skeleton basis shape
$\mathcal{W} \in [0,1]^{K \times N}$	linear blend skinning weights
$w_{k,i} \in [0,1]$	linear blend skinning weight for vertex t_i and the k-th joint
$M(R, \boldsymbol{m}, \vec{\beta}; \Theta)$	parametric model without applied subdivision surfaces
$S(u; \boldsymbol{T}') : \Omega \times \mathbb{R}^{3N} \to \mathbb{R}^3$	subdivision surface evaluation at arbitrary surface coordinates
$S^2 \subset \mathbb{R}^3$	points on the unit sphere
$S^\perp(u; \boldsymbol{T}') : \Omega \times \mathbb{R}^{3N} \to S^2$	subdivision surface normal evaluation at arbitrary surface coordinates
$u \in \Omega = [0,1]^2 \times \{1, 2, \cdots, F\}$	local parameterization on the surface S
$L_f \in \mathbb{R}^{N \times 16}; L'_f, L''_f \in \mathbb{R}^{N \times 12}$	sparse subdivision matrices for limit control points of the patch indexed by f
$U \in \mathbb{R}^{N \times 3}$	matrix representation of $\boldsymbol{T}'$
$\eta^{(f)}, \upsilon^{(f)}, \xi^{(f)}$	limit control points for the patch indexed by f

Optimization

$\phi(x)$	robust Geman-McClure kernel
ψ	latent lifting variable
$\tau \in \mathbb{R}^+$	kernel radius parameter
$\Delta \in \mathbb{R}^{N \times N}$	discrete Laplace-Beltrami operator (matrix representation)

Cardinalities

N model vertex count
F model quad face count
K skeleton joint count
B number of shape blend-shapes
n number of scans in the training set
n_i number of scans for the i-th subject
Q number of subjects in the training set
P number of sampled measurements per scan

Functions

$\text{vec}(\cdot)$ vectorization function

Objective

$\hat{\Theta}$	MAP model parameter estimate
$\Gamma = (\mathcal{B}, \mathcal{R}, \mathcal{U})$	aggregated latent parameters
$\mathcal{B}$	set of Q latent shape vectors β
$\mathcal{R}$	set of n latent pose configurations (R, m)
$\mathcal{U}$	set of nP latent surface correspondences u
$\boldsymbol{p} \in \mathbb{R}^3$	measurement
$E_\bullet$	error terms
$\lambda_\bullet$	error term weights
Π	set of virtual cameras
$\pi : \mathbb{R}^3 \to \mathbb{R}^2$	camera specific projection
$\boldsymbol{q_k} \in \mathbb{R}^2$	2D OpenPose joint keypoints for the k-th joint
$\boldsymbol{q} \in \mathbb{R}^2$	2D OpenPose surface keypoints
$\bar{R} \in SO(3)^K$	mean body pose

References

1. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic faust: Registering human bodies in motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6233–6242 (2017)
2. Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. Journal of Machine Learning Research **15**, 1455–1459 (2014)
3. DeVito, Z., Mara, M., Zollöfer, M., Bernstein, G., Theobalt, C., Hanrahan, P., Fisher, M., Nießner, M.: Opt: A domain specific language for non-linear least squares optimization in graphics and imaging. ACM Transactions on Graphics 2017 (TOG) (2017)
4. Hirshberg, D.A., Loper, M., Rachlin, E., Black, M.J.: Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 242–255. Springer (2012)
5. Loop, C., Schaefer, S.: Approximating catmull-clark subdivision surfaces with bicubic patches. ACM Transactions on Graphics (TOG) **27**(1), 8 (2008)
6. Taylor, J., Stebbing, R., Ramakrishna, V., Keskin, C., Shotton, J., Izadi, S., Hertzmann, A., Fitzgibbon, A.: User-specific hand modeling from monocular depth sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 644–651 (2014)
7. Wald, I., Woop, S., Benthin, C., Johnson, G.S., Ernst, M.: Embree: a kernel framework for efficient cpu ray tracing. ACM Transactions on Graphics (TOG) **33**(4), 143 (2014)
8. Zach, C.: Robust bundle adjustment revisited. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 772–787. Springer (2014)
9. Zach, C., Bourmaud, G.: Iterated lifting for robust cost optimization. In: Proceedings of the British Machine Vision Conference (BMVC) (2017)
10. Zeitvogel, S., Laubenheimer, A.: Towards end-to-end 3d human avatar shape reconstruction from 4d data. In: International Symposium on Electronics and Telecommunications (ISETC). pp. 1–4. IEEE (2018)