BLiSS: Bootstrapped Linear Shape Space

Sanjeev Muralikrishnan[1]   Chun-Hao Paul Huang[2]   Duygu Ceylan[2]   Niloy J. Mitra[1,2]

[1] University College London   [2] Adobe Research
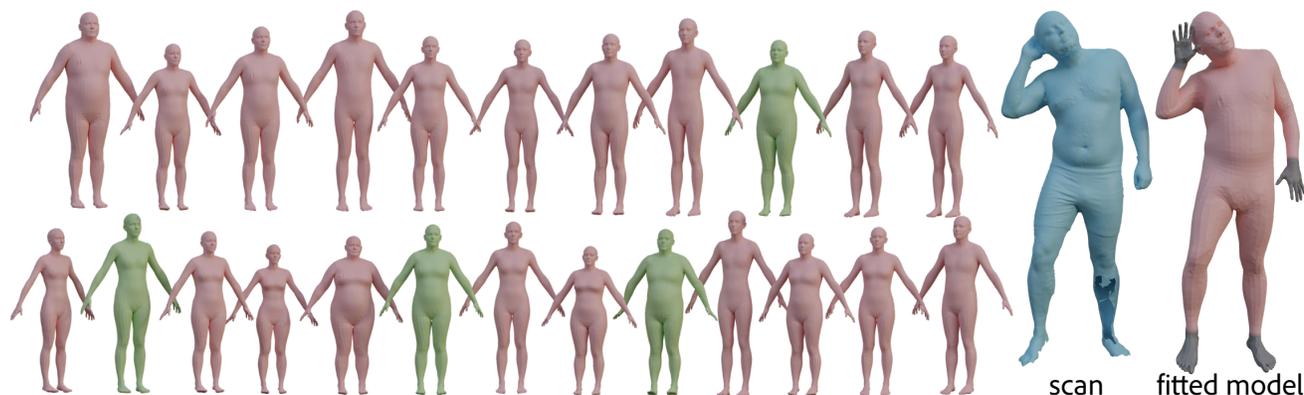
Project Webpage

Figure 1. We present BLiSS, which progressively builds a human body shape space and brings *unregistered* scans into correspondence to a given template mesh. Starting from as few as 200 manually registered scans (green samples), BLiSS creates an expressive shape space (pink samples), performing on par with state-of-the-art models such as SMPL, STAR, and GHUM, while requiring only 5% of annotations compared to the others. (Right) Our space can then recover the body-shape parameters of raw scans by projecting them directly to ours.

Abstract

Morphable models are fundamental to numerous human-centered processes as they offer a simple yet expressive shape space. Creating such morphable models, however, is both tedious and expensive. The main challenge is establishing dense correspondences across raw scans that capture sufficient shape variation. This is often addressed using a mix of significant manual intervention and non-rigid registration. We observe that creating a shape space and solving for dense correspondence are tightly coupled – while dense correspondence is needed to build shape spaces, an expressive shape space provides a reduced dimensional space to regularize the search. We introduce BLiSS, a method to solve both progressively. Starting from a small set of manually registered scans to bootstrap the process, we enrich the shape space and then use that to get new unregistered scans into correspondence automatically. The critical component of BLiSS is a non-linear deformation model that captures details missed by the low-dimensional shape space, thus allowing progressive enrichment of the space.

1. Introduction

Morphable models [8, 31, 46] continue to strongly influence research towards human-centric workflows. This success is explained by the simple and versatile encoding of the underlying shape space, while providing interpretable handles for both shape and pose variations. The compact shape space has been extensively used for a variety of applications, including retexturing [20], shape editing [14], pose and illumination manipulation [55], animation [51], avatar creation [24], to name only a few.

While morphable models are widely considered useful, creating them is surprisingly tedious. Theoretically, given a set of 3D shapes (e.g., scans of human bodies) with vertex-level correspondence, morphable models can 'simply' be built using linear (e.g., principal component analysis (PCA)) or nonlinear (e.g., autoencoder [36, 57]) dimensionality reduction methods. The hurdles lie first in getting scans of many subjects, with a wide coverage of body shape and pose variations, and second in establishing vertex-level dense correspondence across the scans. Given these challenges, not surprisingly, only very few high-quality morphable models (e.g., SMPL [31], STAR [37], GHUM [57]) are publicly available.

The first hurdle has been significantly lowered with rapid

advances [9, 17, 18] in affordable, portable, fast, and robust (hardware) 3D scanning solutions (e.g., RGBD sensors, range scanners etc.). The second hurdle is algorithmic. The dominant approach to establish dense correspondence across the raw scans is to use non-rigid registration [3] to align scans with a template (body) mesh. This works well when the input shapes have limited variations and are clean. Unfortunately, when shape variability is large (as among scans capturing representative variations across a population) or contains holes and noise, successful registration must rely on manual intervention or strong shape priors. Thus, either users have to annotate landmark correspondence across the scans, or provide shape priors to regularize the registration step. Manual annotation is expensive and does not scale easily. Providing a shape prior is also tricky as generating one requires shapes in correspondence – this leads to a chicken-and-egg problem.

We provide a solution that, starting from a small set of registered scans, alternates between building an underlying linear shape space and utilizing the current shape space model to automatically bring new (raw) scans into correspondence. At the core of our approach is a nonlinear deformation setup, expressed in the form of a neural network, i.e., Neural Jacobian Fields (NJF) [2], that helps to predict dense correspondence for scans *close* to the currently modeled shape space. NJF is trained to add information beyond the current PCA space which is critical for registering new target scans, especially in the early stages where our linear shape space may yet not be sufficiently expressive. Once such correspondences are established, they enrich the shape space with additional scans. We repeat this process iteratively until all scans are brought into correspondence and a final shape space is achieved. We term this bootstrapping scheme Bootstrapped Linear Shape Space (BLiSS). In its current form, BLiSS does not handle pose correctives.

We evaluate BLiSS on the commonly used CEASAR dataset [45] and show that starting from only as few as 200 manually registered scans, we can jointly learn a shape space and automatically bring additional scans into correspondence. We evaluate the expressive power of the learned shape space on held out test scans and show that our model performs on par with models that require all scans to be registered manually. We also compare with standard non-registration methods and demonstrate that our method is more robust against noise and holes in the scans. Finally, we apply the methodology of BLiSS in the context of face shape space construction to emphasize its generalizability.

In summary, we make the following contributions: (i) an on-the-fly PCA shape space and correspondence learning framework: starting from only 200 registered shapes, we progressively enrich the model with new shapes and eventually reach a shape space that is on par with the one trained with 3800 registrations directly; and (ii) a novel

combination of linear PCA and non-linear Neural Jacobian Field (NJF) deformation model that brings the target scans into better correspondence.

2. Related Work

2.1. Non-rigid registration

Registering two sets of raw scans (i.e., point clouds) is a long-standing problem [6, 16], typically consisting of two steps: (i) estimating correspondence between the source and the target scans; and (ii) minimizing the distances between each correspondence pair to bring the source closer to the target. Since this work is concerned with human bodies that often deform non-rigidly, we review how correspondences are estimated in non-rigid registration of 3D human data.

Optimization-based (ICP). When the source and the target points are roughly aligned in the ambient 3D space, correspondences can be approximated by seeking nearest points. Following this intuition, existing methods [22, 23, 25, 29, 40] alternate between searching the closest point and deforming the source points, which can be seen as non-rigid variants of the classical Iterative-Closest-Point (ICP) algorithm [7, 12]. For fast convergence, such methods assume the two sets of points to be close enough, or require an "oracle guess" to initialize the correspondences.

Furthermore, these methods often require additional regularization terms to avoid local minima, *e.g.*, Laplacian [49] and ARAP [48]. They impose extrinsic heuristics to constrain the deformation, which do not always apply to the target tasks. In contrast, we employ the recently introduced Neural Jacobian Fields (NJF) [2] that implicitly learns an appropriate regularization in a data-driven manner. We also use NJF in our method as it has been shown to better distribute error by having a global Poisson solve to integrate local gradient (*i.e.*, Jacobian) information.

Learning-based shape matching. Global registration methods exist that match two human shapes without assuming they are close in 3D space. Instead of matching points in 3D space, they measure the similarity in a pre-defined feature space [4, 42, 47, 50] and leverage machine-learning techniques to estimate correspondences [10, 26, 35, 54], optionally refined with a global optimization [5, 11, 44]. The quality of these methods degrades significantly when the shapes are outside the distribution of the training data. More importantly, such methods do not yet handle noise in raw scans, and hence cannot be easily used in our setting.

2.2. 3D Morphable Models for Humans

A standard human body model has to account for pose and shape deformation. In this work, we are particularly interested in the latter – the anthropometric variability across identities, and we focus the discussion on this aspect.

Parametric mesh. Representing human body parts as statistical shape models dates back to Cootes *et al.* [13] in 2D and Blanz & Vetter [8] in 3D. The latter has become the de facto standard for modeling 3D human shapes [19, 52], often called 3D morphable models (3DMM). The goal of a 3DMM is to adapt the template to each person by controlling the shape variations in a low-dimensional space. In the context of whole body, a myriad of work [3, 28, 31, 37–39, 57, 58] has been proposed for this purpose and has led to rapid progress in monocular and multi-view human body reconstruction [52].

Learning such a parametric shape space, however, requires firstly, a large database of body scans, and more difficultly, bringing them into correspondence by registering a common template mesh to them. Most models above are trained with thousands or ten thousands of registrations to body scans in CAESAR [45] and/or SizeUSA [1], curated with manual intervention for quality control (see Table 1). Another frequently overlooked challenge is that databases have each subject scanned in similar but not exactly the same pose (*e.g.*, A pose in CAESAR) while the template is desired to be in one canonical pose (typically T pose). To factor out the pose variation in the data, an un-posing process is performed to bring registrations to the canonical pose [31], which we refer to as "canonical shapes" in the rest of the text. Any artifact introduced in this step will be kept in the learned shape space. Our formulation can take A-posed scans as input and output the canonical shapes in T-pose, requiring no un-posing before including them to training.

The most relevant work of Hirshberg *et al.* [23] explores a "semi-supervised" setting of co-registration of multiple scans similar to ours. However, their approach differs from us in several aspects: (i) they aim to "simultaneously" learn a morphable model and bring scans into correspondence in one shot, without iterations. Consequently, it amounts to a big optimization problem where one has to provide good initialization (e.g., via manual landmarks) and carefully anneal the weights of each term, which may be easy to break. Due to the complexity of the optimization, this method can handle hundreds of scans whereas we can scale our iterative pipeline to thousands of scans; (ii) they rely on model-free registrations with a nearly isometric regularization term to capture information beyond the model space. While there is no publicly available code for us to perform an exact

comparison, we compare to baselines where we employ a similar edge-preserving non-rigid registration approach and demonstrate superior performance.

Implicit surfaces. A well-known limitation of meshes is that it is limited in handling deformations that require changes in the topology. Recent work [15, 33, 34, 53] explores representing human bodies with neural fields [56], which does not assume a consistent mesh topology. They take the translations and rotations of body joints as input and estimate whether a query 3D point is inside the body or not. However, so far the effort has been devoted primarily in generalization of articulated poses, where large-scale motion capture datasets [21, 32] are used for training.

To help generalize to multiple subjects, it is encouraged to condition the networks on body shapes. However, in these methods, shape information is carried only in the locations of input joints, which is a very coarse anthropometric feature as two bodies can share the same joints but different surface shapes. In this work, we consider explicit surface meshes in order to better capture details in human bodies.

3. Method

3.1. Overview

Given a large set of raw scans $\mathcal{S}$ of varied human body shapes in roughly a similar pose (e.g., A-pose), our goal is to learn a shape space in a canonical but different pose (e.g., T-pose) that captures the variation of plausible body shapes. We learn a shape space with respect to a predefined shape template topology; in our setup, we use the SMPL [31] template, denoted as T_{SMPL} with N vertices. We also assume having access to a small set of registered shapes $\mathcal{R}$ where a
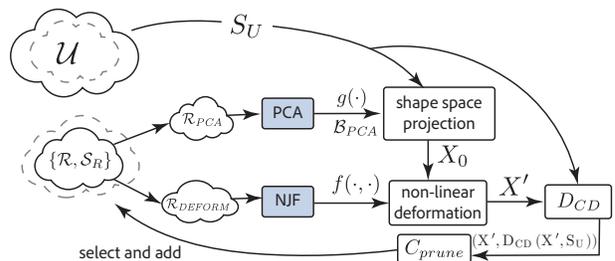

Figure 2. Given a sparse set of scans $\mathcal{S}_R$, and their registrations $\mathcal{R}$ to a common template, we learn a linear shape space $\mathcal{B}_{PCA}$ using $\mathcal{R}_{PCA}$ and train a non-linear NJF-based deformation model using $\mathcal{R}_{DEFORM}$. Then, given a scan S_U from a set of unregistered scans $\mathcal{U}$, we project it to the PCA basis to obtain X_o and utilize NJF-based deformation to recover its registration to the template X' in the canonical pose. To enhance our shape space, we calculate the Chamfer Distance (D_{CD}) of registrations to target scans. We add all registrations where the distance falls within one standard deviation of the minimum distance to $\mathcal{R}_{PCA}$. We repeat this process to jointly register raw scans and enrich our shape space.

Table 1. Comparison with SMPL [31], GHUM [57], DenseRac [58], and STAR [37] w.r.t. the number of registrations used in training respective morphable models.

Method	SMPL	GHUM	DenseRac	STAR	BLiSS
# shape space	3800 PCA	64000 PCA/VAE	2500 PCA	15000 PCA	200 PCA

Algorithm 1

```
1: procedure BLiSS(R, S_R, U, f, n)
     ▷ R = Registered Set, S_R = Corresponding set of scans
     ▷ U = Unregistered set of scans
     ▷ f = Deformation Model, n = Number of rounds
2:     for Round ∈ [1,n]: do
3:         R_PCA, R_DEFORM ← R
4:         B_PCA, g(·) ← PCA(R_PCA)              ▷ build shape sp.
5:         for X, S_X ∈ (R, S_R): do             ▷ build deform mod.
6:             {α*_i}, θ* = g(S_X)                ▷ Fit using basis
7:             X_o = S̄ + Σ^k_{i=1} α*_i v_{s_i}   ▷ init. canon. shape
8:             X' = f(X_o, S_X)                   ▷ Register with NJF
9:             L_vertex = ‖X' − X‖²              ▷ Vertex Loss
10:            L_Jacobian = ‖J' − J‖²           ▷ Jacobian Loss
11:            f(...; γ_i) = f(...; γ_{i−1})     ▷ Backprop
12:        end for
13:        C = ∅                                 ▷ New Candidate Registrations
14:        for S_U ∈ U: do
15:            {α*_i}, θ* = g(S_U)
16:            X_o = S̄ + Σ^k_{i=1} α*_i v_{s_i}
17:            X' = f(X_o, S_U)
18:            C ← C ∪ X'
19:        end for
20:        D ← ChamferDist(C, U)
21:        th ← min(D) + σ(D)
22:        C_prune ← {c|c ∈ C, ChamferDist(c, U) < th}
23:        R ← R ∪ C_prune                       ▷ Update Basis
24:    end for
25:    return R
26: end procedure
```

small set of scans S_R have been brought to the same topology as T_{SMPL} in the canonical pose via a manual non-rigid registration process to avoid any registration artifacts. Starting with R and T_{SMPL}, we iteratively expand R with new shapes from the unregistered scans U that are automatically brought into correspondence with T_{SMPL} and learn an enhanced shape space. Note that we always have $S = S_R \cup U$.

Our method works by deforming the template T_{SMPL} to closely match a new raw scan $S_U \in U$. This deformation model consists of two parts: (i) a PCA-based shape space B_{PCA} that provides a search space for shapes; and (ii) a neural deformation model f that maps shapes obtained by searching B_{PCA} to targets that better capture the shape details of the raw scan. The two deformation models work in tandem to jointly register scans and yield correspondence with T_{SMPL}, resulting in registered scans based on the current shape space. We then 'close the loop' by selecting a few new registrations based on their distance to the scan and adding them to recompute a PCA basis B_{PCA}, thus enriching the shape space further. We repeat this process for multiple iterations, with each pass progressively learning a richer shape space and using it to register raw scans. Figure 2 illustrates the pipeline of BLiSS.

3.2. PCA-based Shape Space

We use a subset of the shapes in R, R_{PCA} to compute a PCA basis in a similar fashion to classical works like

SMPL [31] and STAR [37]. Note that we use only a subset of the R and save the rest for the data-driven deformation model (see Section 3.3). Our shape space B_{PCA}, similar to others, is composed of a pose-corrective deformation basis allowing for pose-conditioned deformations and a shape basis that enables body-shape deformations. In our work, since we are primarily interested in learning a space of body shapes, we borrow the pose corrective directly from SMPL, which is denoted as $B_P(\theta) : \mathbb{R}^{\|pose\|} \to \mathbb{R}^{3N}$ as well as a rigged skeleton to pose T_{SMPL}, where $\|pose\| = 24 \times 3$, corresponding to 3 axis-angles for each of the 24 joints. We represent the shape basis B_{PCA} with k shape eigenvectors $B_{PCA} := \{v_{s_i}\}$, where k is selected such that the shape variation in the dataset is explained using the k basis vectors. In this computed space, we define a new shape S_c in any particular pose θ as,

$$S_c(\{\alpha_i\}, \theta) := \bar{S} + \sum_{i=1}^{k} \alpha_i v_{s_i} + B_P(\theta)$$

$$S_p(\{\alpha_i\}, \theta) := \mathcal{W}\left(S_c(\{\alpha_i\}, \theta), \mathcal{J}, \theta, W_s\right),$$

where $\bar{S}$ is the mean shape, $\mathcal{J}$ is the joint regressor that provides the joint locations given the vertex positions in the shape, W_s is a fixed set of skinning weights, and finally, $\mathcal{W}$ is the skinning function as defined in [31]. Now, given a target scan S and a current set of shape basis vectors v_{s_i}, we optimize for the pose parameters and shape coefficients:

$$
\begin{aligned}
g(S_U) &:= (\{\alpha_i^\star\}, \theta^\star) \\
&= \arg\min_{\{\alpha_i\}, \theta} D_{CD}(\mathcal{W}(S_c, \mathcal{J}, \theta, W_s), S_U), \quad (1)
\end{aligned}
$$

where D_{CD} is the Chamfer Distance and S_U is an unregistered raw scan.

We optimize Eq. 1 to find the shape in B_{PCA} that best matches the scan S_U while also optimizing for the pose parameters θ. In other words, the function g "projects" the raw scan S_U onto the shape space B_{PCA}. After optimization, we obtain the canonical shape that corresponds to the scan as $X_o := \bar{S} + \sum_{i=1}^{k} \alpha_i^\star v_{s_i}$. Note that due to the limited expressivity of the linear basis, X_o may not accurately represent S_U. We now seek a deformation model that can further enrich X_o with the details from S_U. This shape space optimization also provides a good initial point to seed our subsequent nonlinear deformation model, as explained next.

3.3. Neural Deformation with NJF

In our work, a nonlinear deformation is simply an assignment of new 3D positions to the vertices of the given (template) mesh. We adopt Neural Jacobian Fields (NJF) [2] as our nonlinear deformation model f. NJF trains an MLP to map triangles on a source mesh to a corresponding deformed triangle on a target mesh using only local information. The key step is to have this training receive gradients

through a differentiable global Poisson Solve layer to then directly predict the positions of the vertices.

We consider another subset of shapes in $\mathcal{R}$, $\mathcal{R}_{DEFORM}$ where $\mathcal{R}_{DEFORM} \subset \mathcal{R} \setminus \mathcal{R}_{PCA}$, and their corresponding raw scans to train NJF. For each shape $X \in \mathcal{R}_{DEFORM}$, we first optimize for parameters $g(S_X)$ where S_X is the raw scan corresponding to X, giving us a shape space projection X_o. We then train NJF to map the canonical X_o to the canonical X, *conditioned* on the scan S_X that can be in any pose. Essentially, we ask our deformation model f to deform the result of our (current) shape space projection X_o to the target registration X that contains richer details. The deformation function f is conditioned on the raw scan representing the target, and can fix any residues not covered by the optimization step. Specifically, we train f with per-vertex L_2 loss,

$$L_{vertex} := \|f(X_o, S_X; \gamma) - X\|^2, \qquad (2)$$

where γ represents learnable parameters, and a per-triangle Jacobian loss $L_{Jacobian}$ which supervises the ground-truth Jacobians J (see [2]), with our total training signal being,

$$L_{total} = 10 \cdot L_{vertex} + L_{Jacobian}. \qquad (3)$$

We slightly abuse notation in Equation 2 – in practice, X_o and S_X are represented as features and not by vertex locations themselves. Specifically, we use Pointnet encodings [41] of both shapes. Details of the network architectures and the features used are in the supplemental.

With our initial shape space defined by $\mathcal{R}_{PCA}$ and a non-linear deformation model trained with $\mathcal{R}_{DEFORM}$, we can now use these in tandem to register new scans.

3.4. Closing the Loop

For each unregistered scan $S_U \in \mathcal{U}$, we first fit the template T_{SMPL} to it by optimizing parameters $g(S_U)$ in Equation 1, to obtain the canonical pose X_o. We then use the trained NJF to predict the final registration as $f(X_o, S_U; \gamma) \to X'$. The model X' is then "posed" to match the pose of S_U by using the optimized pose parameters $\theta^\star$. We compute the Chamfer Distance to their corresponding scans, and if the error falls within one standard deviation from the minimum error, we augment $\mathcal{R}_{PCA}$ with the new shapes X' in T-pose. In the next iteration, the shape space will be updated by computing PCA with the augmented set $\mathcal{R}_{PCA}$. The updated basis also provides new initial states for training our deformation model f. Note that we do not add the PCA projections X_o to $\mathcal{R}_{PCA}$ as it does not carry "fresh" information like X'.

We repeat the steps of constructing a PCA basis, learning an NJF based deformation model, and registering new scans for several rounds, with each round expanding the shape space (see Algorithm 1).

4. Experiments

4.1. Dataset and Protocols

Dataset. We picked 429 scans from the CAESAR dataset and had them registered by a professional artist, *i.e.*, $|\mathcal{R}| = 429$. Note that the artist took 40-60 min per scan using a combination of landmark point specification, running non-rigid ICP, and then manually fine-tuning dense correspondence correction/specification (*e.g.*, around fingers, armpit, etc.), costing around \$25 for each scan. We consider these artist-registered meshes as Ground Truth for evaluation, training, and, in the case of some baselines, as targets. Specifically, we first sample two mutually exclusive sets $\mathcal{R}_{PCA}$ and $\mathcal{R}_{DEFORM}$ from $\mathcal{R}$ to train the initial PCA space and NJF respectively, where $|\mathcal{R}_{PCA}| = 100$ and $|\mathcal{R}_{DEFORM}| = 100$. We use all the remaining 229 registrations as $\mathcal{R}_{EVAL}$ for evaluation purposes unless noted otherwise.

Since the original CAESAR dataset consists of around 4000 scans, 429 of which we have registrations from the artist, we consider the rest 3.5k scans as unregistered scans $\mathcal{U}$. In Algorithm 1 Ln. 14-18, scans in $\mathcal{U}$ are brought to correspondence, added to $\mathcal{R}_{PCA}$, and contribute to the new shape space $\mathcal{B}_{PCA}$, whereas $\mathcal{R}_{DEFORM}$ is kept fixed to the initial 100 artist registrations. Throughout the experiment, we always use k=11 basis for any PCA-based shape space, denoted as PCA (see supplemental for tests with other k values). Note that despite $\mathcal{R}_{DEFORM}$ being fixed, in each round, since the basis of the shape space $\mathcal{B}_{PCA}$ changes, X_o changes, and consequently, the amount of details that NJF needs to compensate also changes. Hence, our NJF is re-built in each iteration (Algorithm 1 Ln. 5-11). At test time, since BLiSS consists of a PCA shape space and an NJF network, whenever we evaluate only the learned PCA space, we denote it as "PCA only" to distinguish from running the full pipeline.

Metrics. For each registration in $\mathcal{R}_{EVAL}$, we take the corresponding raw scan and obtain a registered shape using either our method or the baselines below. We measure the vertex-to-vertex (v2v) distance between the ground truth and the estimated canonical shapes, using the artist-annotated scan-to-template correspondences. When comparing shape spaces with different topology, such as GHUM, we start by non-linearly registering the target and our mean body shape models in the same pose. Then, we use barycentric coordinates on the corresponding face to map each point on our template to the target body model. Additionally, we calculate the vertex-to-plane error (v2p), which does not require meshes to have the same topology.

(i) **FULLANNOTATION**: We spare 29 registered shapes for evaluation and use all the remaining 400 annotated meshes to train a PCA model. We further train an NJF with the same 400 scans to add the missing details not cov-

Table 2. Evaluating ours against alternatives. (i) Learning a one-time static shape space from 400 available registrations provides an upper bound; (ii) and (iii) provide baselines replacing our non-linear deformation model with classical non-rigid registration. Errors are in cm.

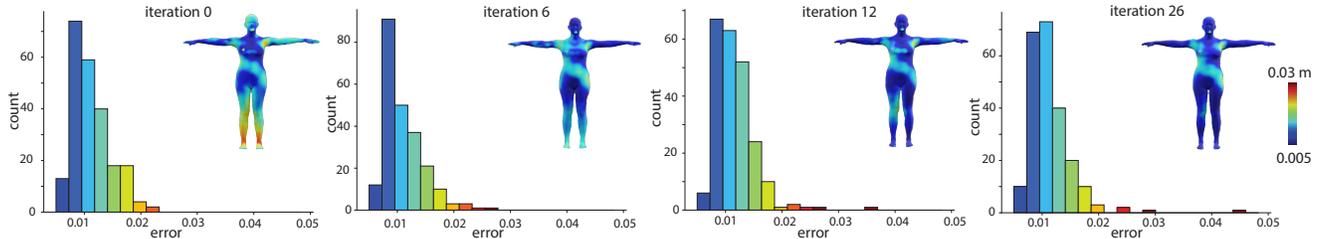| Method	$|\mathcal{R}_{EVAL}|$	initial $|\mathcal{R}_{PCA}|$	$|\mathcal{R}_{DEFORM}|$	regularizations	# shapes $\in \mathcal{U}$	v2v (↓)
(i) FULLANNOTATION- PCA+NJF	29	400	400	✗	✗	*0.87*
(ii) BASELINE1- PCA + non-rigid	229	100	✗	small $\|\Delta v\|$	800	3.11
(iii) BASELINE2- PCA + non-rigid	229	100	✗	edge-preserving	800	3.26
BLiSS (PCA only)	229	100	100	✗	800	1.31
BLiSS (PCA+NJF)	229	100	100	✗	800	**0.90**

Figure 3. We show the histogram of the v2v error of the scans in our test set at different iterations of our method. We also color code the per-vertex error for an example scan. As our method progresses, the error decreases, and we observe a slight left shift in the histogram as the shape space improves. Insets show residue error on one scan over iterations.

ered by the PCA model, denoted as "FULLANNOTATION-PCA+NJF". This baseline represents scenarios where one trains the model *in one go* with all available registrations, without any bootstrapping schemes that leverage the unregistered scans. Hence, this can be seen as an *upper bound.*

Baselines. We consider several baselines, where NJF is replaced in our pipeline with classical non-rigid registration methods. Given an unregistered scan S_U, we first obtain the projection in the PCA space X_o and then optimize the location of each vertex on X_o, such that when posed with $\theta^\star$, the shape yields low Chamfer Distance to the scan S_U. This "free-form deformation" scheme can fall into local minimum easily even if we provide X_o as close initialization. Therefore we define new baselines where we constrain it with standard regularization terms: (ii) **BASELINE1**: vertices should not be deviating too far from the canonical shapes X_o, *i.e.*, $\|\Delta v\|$ should be small favoring smooth surfaces; (iii) **BASELINE2**: deformation should preserve edge lengths, *i.e.*, favor near-isometric deformations.

Finally, we also compare to existing shape spaces including SMPL [31], STAR [37], and GHUM [57].

4.2. Results and Discussions

Progressive improvement of the shape space. First, we illustrate how our shape space is progressively improved, *i.e.*, becomes more expressive. In Figure 3, we show the histograms of the v2v residue error at different iterations of our method as it consumes new scans from $\mathcal{U}$ and visualize the error as heat maps in each inset image. One can observe

that the error gets reduced as we have more rounds, *i.e.*, the correspondence quality improves progressively.

Further, we compare with a method that simultaneously burns all available 400 scans into model training – FULLANNOTATION. As shown in Table 2 upper part, it attains the lowest v2v error of 0.87cm on a smaller evaluation set of 29. In the bottom part, we can observe that, despite being trained initially with only 100 registrations, BLiSS yields v2v error that is on par with the upper bound by automatically extending the set of registered scans to 800.

Effect of NJF as a non-linear registration module. In BASELINE1-2, we run Algo. 1 but estimate detailed shapes X' at Ln. 16 by non-rigid registration methods instead of NJF f. We then evaluate how well the resulting PCA shape space explains the test set $\mathcal{R}_{EVAL}$ and compare with our PCA shape space. In Table 2 bottom part, we observe that after consuming 800 unregistered scans, our shape space explains scans in $\mathcal{R}_{EVAL}$ with 1.31cm v2v error (PCA only), while NJF further reduces it to 0.90cm. Consuming the same amount of scans, shape spaces enriched with X' from non-rigid registration yield errors of 3.11cm and 3.26cm, respectively. This suggests using a data-driven NJF in the loop recovers better correspondence than optimization-based registration methods, and when included in $\mathcal{R}_{PCA}$, it leads to a new PCA space $\mathcal{B}_{PCA}$ with richer information.

Figure 3 shows how the model improves over iterations/rounds (from left to right). While NJF helps add lost details, it is still limited by the sparse set of training pairs (100 in our case) and may not recover all the details.

Comparison to other shape spaces. We compare BLiSS with the following existing shape spaces: (i) the classical SMPL [31] shape space trained with the registrations of 3800 CAESAR scans, (ii) its follow-up STAR [37] which uses additional 11000 registrations of the SizeUSA dataset [1], (iii) GHUM which uses registrations for an additional proprietary dataset of 64000 scans (where a majority consists of body, hand, and, facial pose variations) along with CAESAR. GHUM presents a linear shape space as well as a VAE-based non-linear shape space, both of which we include in our comparisons. For each corresponding scan in $\mathcal{R}_{EVAL}$, we optimize for the pose and shape parameters of each body model and report both the v2v and v2p errors in Table 3. We also show qualitative comparison in Figure 4 where we color code each optimized body model using the v2p error with respect to the ground truth artist provided registration.

Table 3. Ours, after absorbing 800 shapes from CEASER, outperformed SMPL, STAR, and GHUM even though we only used 200 registered scans, compared to their much larger number of scans.

Method	# shapes $\in \mathcal{R}$	v2v ($\downarrow$) (cm)	v2p ($\downarrow$) (cm)
SMPL (PCA)	3800	1.72	0.62
STAR (PCA)	15000	2.15	**0.58**
GHUM-PCA	64000	6.74	3.01
GHUM-VAE	64000	5.89	2.63
BLiSS	200	**0.90**	0.65

We observe that BLiSS yields consistently lower v2v error than other shape spaces. We use 11 PCA components for SMPL, STAR, and BLiSS while all the PCA components for GHUM. Our Ground Truth are artist-annotated registrations, which can still potentially contain errors which might have an effect on this gap. Nevertheless, our method per-
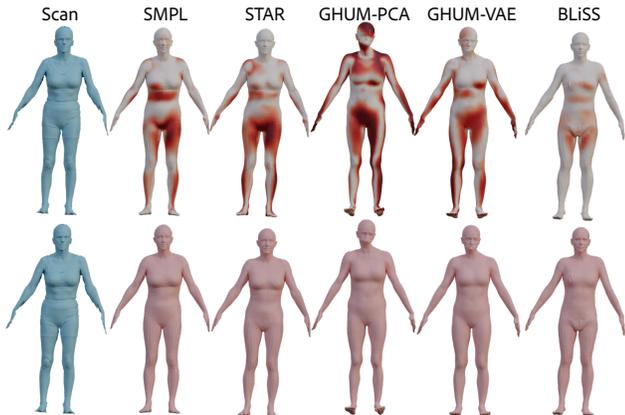

Figure 4. For a given raw scan, we register each body model by predicting pose and body shape parameters. (Top) Each result is color coded based on the v2p error in meters w.r.t. the ground truth registration provided by the artist.

forms on par with SMPL and STAR based on the v2p error and better than GHUM. Hence, the primary observation in Table 3 is that, despite starting from only a small amount of registrations (100+100), BLiSS yields on-par expressivity compared to a model trained with an order of magnitude more registrations. We attribute this to the novel combination of linear PCA and non-linear NJF deformation model, as well as the progressive scheme leveraging such a hybrid deformation model for better correspondence.

Diversity of body shape spaces. We qualitatively show the diversity of body shapes represented by our shape space by randomly sampling our final PCA space using farthest sampling in terms of vertex differences. Sampled shapes are shown in Figure 1. In Figure 5, we show shape variations captured by our shape space's top three PCA modes.
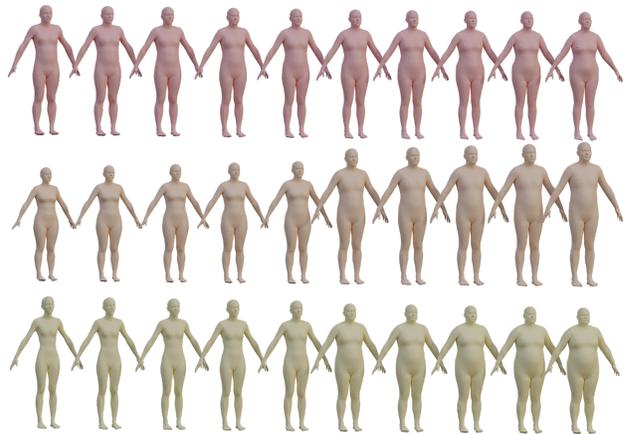

Figure 5. We show shapes along the top three principal directions in different rows, and observe variations in gender, height, and weight along the respective PCA modes.

In order to compare the diversity of our and existing shape spaces, we sample 500 body shapes in each shape space by furthest point sampling. For each sampled body shape, we compute is nearest sample within the same shape

Table 4. We use Farthest Sampling to gather 500 shapes from each space. To determine the similarity between the different spaces, we calculate the distance between each shape in one space and its closest counterpart in all other spaces, including off-diagonal entries. We then report the average distance (in centimeters) for each possible pairing of spaces in both directions. Low values for (A, B) and (B, A) suggest the two spaces are similar. We compute the diversity of samples inside each space for diagonal entries, with higher values indicating more diversity.

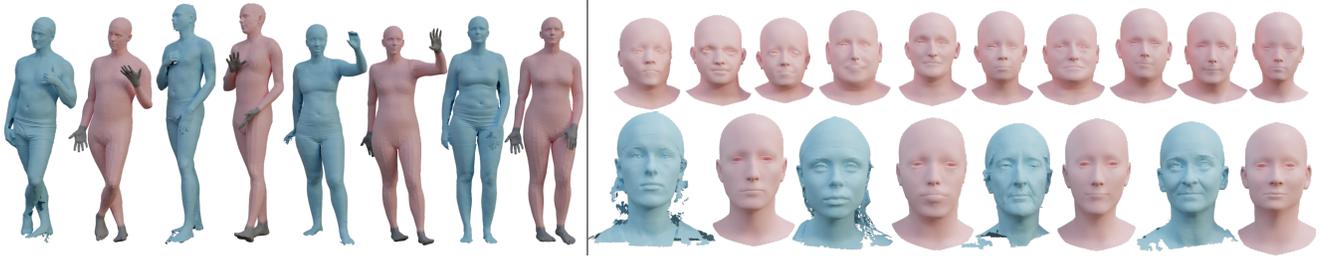| Space	Ours	GHUM	STAR	SMPL
Ours	(4.10)	3.57	1.38	1.46
GHUM	4.03	(4.48)	3.65	3.71
STAR	1.79	3.74	(3.96)	1.37
SMPL	1.90	3.75	1.36	(4.14)

Figure 6. *Left:* Registration (pink) of noisy scans (blue) with our final shape space. Since our model does not capture finger-level details, after optimization, the joints corresponding to the greyed-out regions are reset to default poses. *Right*: We show sampled faces from our final face-shape space after growing it from 20 → 800 shapes. We observe a variety of face changes in the cheek and nose regions. (Bottom) We take the test scans from the COMA dataset (in blue) and register them in our final face-shape space, which is shown in pink.

space by measuring the v2v error. We report the average of such pairwise sample error along the diagonal of Table 4 where higher pairwise distance means a diverse shape space. As shown quantitatively, the diversity of our shape space is on par with existing shape spaces. We also compute each sample in one body shape space's nearest sample in all other shape spaces. For each shape space in each row, we compute the pairwise sample distance for each shape space in each column. Smaller numbers indicate the similarity of shape spaces. We observe that our shape space is closer to the SMPL and STAR shape spaces.

Number of PCA components. While we use k=11 basis consistently for all PCA-bsaed methods, we also analyze the effect of using more bases with k=30, 50, 100. Using higher number of basis increases the expressivity of PCA, but empirically we observe very little change in our metrics – to the order of 10^{-5} – by considering more PCA components, and thus stick to using just k=11 in all experiments.

Application. A typical application of a body shape space is to predict a given raw scan's shape parameters. We demonstrate the use of BLiSS shape space in such an application in Figures 6 and 1 (right) . Since our work focuses on capturing body shape variation, we optimize for pose in SMPL's pose space. For each raw scan we use 9 manually annotated landmarks to estimate the initial pose, then we estimate the body pose (with SMPL) and shape parameters in our shape space. We observe that our space accurately estimates the body shape despite the scans being noisy.

Face registration. To demonstrate the generalization of our approach, we sample 20 faces from FLAME [30] to create an initial face shape space. We then iteratively register faces from the COMA [43] dataset. Note that we use the NJF module trained on full-body human scans. Registrations shown in Fig. 6. We do *not* assume paired data for this task and instead exploit NJF's invariance to topology and use the pre-trained NJF to deform the faces. NJF was trained on centroids and Wave Kernel Signature as input features; these features are computed for the linearly registered face scans and fed to NJF to add further details. We

iteratively train BLiSS until we add 800 face-shapes to ours.

5. Conclusions

We have presented a method that takes in a small set of artist-annotated scans along with a much larger corpus of unregistered scans, and jointly learns a (linear) shape space while progressively bringing the unregistered scans into correspondence. At the core of our approach is a novel formulation that continuously refines the underlying shape space and a learned nonlinear module that automatically registers models 'close' to the current shape space. We demonstrate that our approach trained only with 200 registered scans, can produce competitive performance compared to established shape space models, trained using thousands of registered scans. We also presented similar results for face datasets.

One limitation of our method is that it does not capture pose corrective shape space. To address this, we plan to use the non-linear deformation module in an iterative fashion to learn a pose corrective shape space from unregistered scans. We may also explore the use of nonlinear shape spaces, such as AE or VAE, but it poses a challenge of growing robustly on limited data, especially in the initial rounds of the approach. It is worth noting that our method cannot handle hands in complex poses due to the lack of finger articulation in SMPL's pose space.

It is advisable to minimize the use of shape regularizers like ARAPReg [27], data-efficient shape VAE [36] as they may limit the flexibility of the learned shape spaces. Instead, we can adopt an alternate approach by directly selecting models and asking artists to annotate them, maximizing fresh information beyond fully self-supervised shape space building.

References

[1] SizeUSA dataset. https://www.tc2.com/size-usa.html, 2017. 3, 7

[2] Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *SIGGRAPH*, 2022. 2, 4, 5

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

[4] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011. 2

[5] Omri Azencot, Anastasia Dubrovina, and Leonidas Guibas. Consistent shape matching via coupled optimization. In *Comput. Graph. Forum*, pages 13–25. Wiley Online Library, 2019. 2

[6] Ben Bellekens, Vincent Spruyt, Rafael Berkvens, and Maarten Weyn. A survey of rigid 3D pointcloud registration algorithms. In *AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, pages 8–13, 2014. 2

[7] Paul J Besl and Neil D McKay. A method for registration of 3-d shapes. *IEEE TPAMI*, 14(2):239–256, 1992. 2

[8] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 1, 3

[9] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *ICCV*, pages 2300–2308, 2015. 2

[10] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *NeurIPS*. Curran Associates, Inc., 2016. 2

[11] Qifeng Chen and Vladlen Koltun. Robust nonrigid registration by convex optimization. In *ICCV*, pages 2039–2047, 2015. 2

[12] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 2

[13] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 3

[14] Kevin Dale, Kalyan Sunkavalli, Micah K. Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. *ACM TOG*, 30(6):1–10, 2011. 1

[15] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA: Neural articulated shape approximation. In *ECCV*, 2020. 3

[16] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3D registration. In *Comput. Graph. Forum*, pages 559–589. Wiley Online Library, 2022. 2

[17] Mingsong Dou, Jonathan Taylor, Henry Fuchs, Andrew Fitzgibbon, and Shahram Izadi. 3D scanning deformable objects with a single rgbd sensor. In *CVPR*, pages 493–501, 2015. 2

[18] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4D: Real-time performance capture of challenging scenes. *ACM TOG*, 35(4):1–13, 2016. 2

[19] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3D morphable face models—past, present, and future. *ACM TOG*, 39(5):1–38, 2020. 3

[20] Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormahlen, Patrick Perez, and Christian Theobalt. Automatic face reenactment. In *CVPR*, pages 4217–4224, 2014. 1

[21] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. Movi: A large multi-purpose human motion and video dataset. *Plos one*, 16(6):e0253157, 2021. 3

[22] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3D-CODED : 3D correspondences by deep deformation. In *ECCV*, 2018. 2

[23] D. Hirshberg, M. Loper, E. Rachlin, and M.J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *ECCV*, pages 242–255, 2012. 2, 3

[24] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM TOG*, 36(6):1–14, 2017. 1

[25] Chun-Hao Huang, Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. A bayesian approach to multi-view 4d modeling. *IJCV*, 116(2):115–135, 2016. 2

[26] Chun-Hao Paul Huang, Benjamin Allain, Edmond Boyer, Jean-Sébastien Franco, Federico Tombari, Nassir Navab, and Slobodan Ilic. Tracking-by-detection of 3D human shapes: from surfaces to volumes. *IEEE TPAMI*, 40(8):1994–2008, 2017. 2

[27] Qixing Huang, Xiangru Huang, Bo Sun, Zaiwei Zhang, Junfeng Jiang, and Chandrajit Bajaj. Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators. *CoRR*, abs/2108.09432, 2021. 8

[28] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *CVPR*, pages 8320–8329, 2018. 3

[29] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Comput. Graph. Forum*, pages 1421–1430. Wiley Online Library, 2008. 2

[30] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 8

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 34(6):248:1–248:16, 2015. 1, 3, 4, 6, 7

[32] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, pages 5442–5451, 2019. 3

[33] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. LEAP: Learning articulated occupancy of people. In *CVPR*, 2021. 3

[34] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. COAP: Compositional articulated occupancy of people. In *CVPR*, pages 13191–13200, 2022. 3

[35] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5115–5124, 2017. 2

[36] Sanjeev Muralikrishnan, Siddhartha Chaudhuri, Noam Aigerman, Vladimir Kim, Matthew Fisher, and Niloy Mitra. Glass: Geometric latent augmentation for shape spaces. In *CVPR*, pages 18552–18561, 2022. 1, 8

[37] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. STAR: A sparse trained articulated human body regressor. In *ECCV*, pages 598–613, 2020. 1, 3, 4, 6, 7

[38] Ahmed A A Osman, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. SUPR: A sparse unified part-based human body model. In *ECCV*, 2022.

[39] 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 *CVPR*, pages 10975–10985, 2019. 3, 1

[40] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: A model of dynamic human shape in motion. *ACM TOG*, 34(4):120:1–120:14, 2015. 2

[41] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. 5

[42] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, pages 652–660, 2017. 2

[43] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 725–741, 2018. 8

[44] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete optimization for shape matching. In *Computer Graphics Forum*, pages 81–96. Wiley Online Library, 2021. 2

[45] Kathleen M. Robinette, Sherri Blackwell, Hein Daanen, Mark Boehmer, Scott Fleming, Tina Brill, David Hoeferlin, and Dennis Burnsides. Civilian American and European Surface Anthropometry Resource (CAESAR) final report. Technical Report AFRL-HE-WP-TR-2002-0169, US Air Force Research Laboratory, 2002. 2, 3

[46] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG*, 36(6):245:1–245:17, 2017. 1

[47] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 2

[48] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, pages 109–116, 2007. 2

[49] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004. 2

[50] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Comput. Graph. Forum*, pages 1383–1392. Wiley Online Library, 2009. 2

[51] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. *ACM TOG*, 34(6): 183–1, 2015. 1

[52] Yating Tian, Hongwen Zhang, Yebin Liu, and Limin Wang. Recovering 3D human mesh from monocular images: A survey. *arXiv preprint arXiv:2203.01923*, 2022. 3

[53] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-GIF: Neural generalized implicit functions for animating people in clothing. In *ICCV*, 2021. 3

[54] Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *CVPR*, pages 1544–1553, 2016. 2

[55] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: Face analysis in the wild using synthetic data alone. In *ICCV*, pages 3681–3691, 2021. 1

[56] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Comput. Graph. Forum*, 2022. 3

[57] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D human shape and articulated pose models. In *CVPR*, pages 6183–6192, 2020. 1, 3, 6, 2

[58] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. DenseRaC: Joint 3D pose and shape estimation by dense render-and-compare. In *ICCV*, 2019. 3

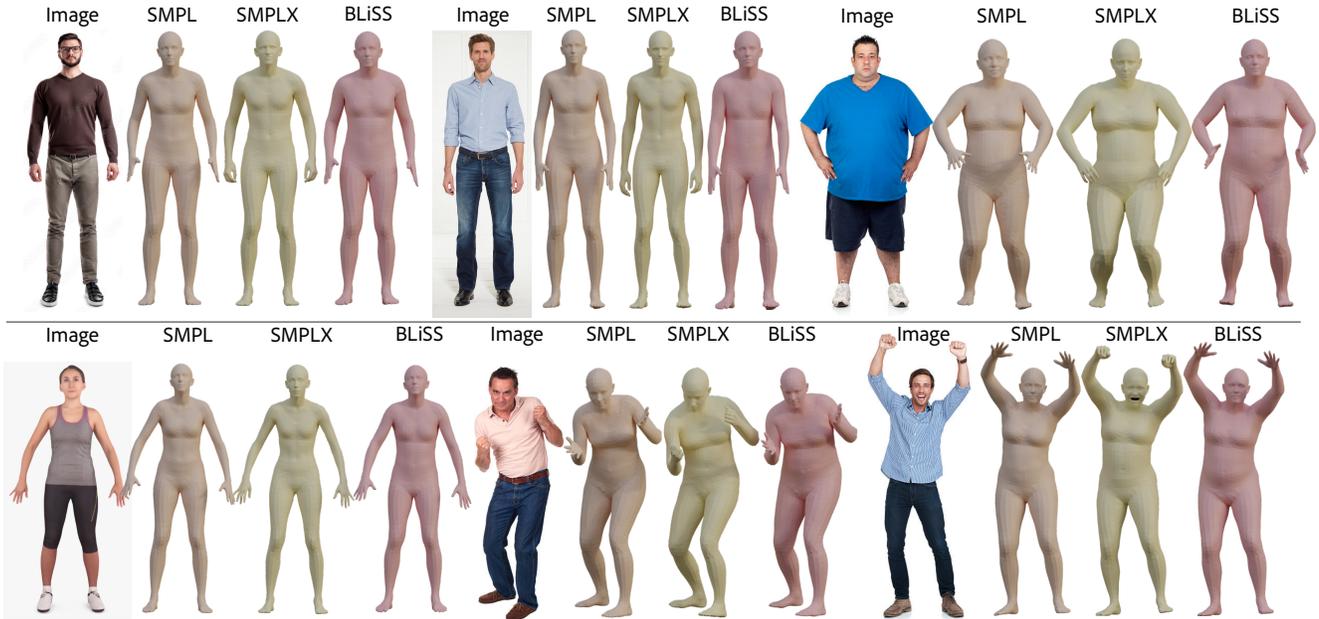# BLiSS: Bootstrapped Linear Shape Space

Figure 7. Here we use the trained SMPLify-X [39] model to estimate the shape from a single image. For BLiSS, we plugin our shape space as a drop-in replacement for SMPL's space, while using SMPL's pose space.

6. Shape Estimation from Single Image

We additionally compare our shape space with SMPL's by plugging them into an optimization-based body reconstruction framework SMPLify-X [39]. Since BLiSS doesn't handle pose, we use SMPL's pose space. SMPLify-X then searches the given space to fit the shape to the given image. We observe in Figure 7, that BLiSS's estimated shape is comparable with that of SMPL and SMPL-X, despite having started from only 200 given registrations. Note that unlike SMPL-X, SMPL (and hence BLiSS) does not articulate hand poses.

7. Iterative improvements

Registrations in the initial rounds of our shape-space are often coarse, lacking details, as shown in the two pale colored faces in the middle in Figure 2 8. As the space is more densely populated, later registration more closely capture finer details of the given scan.

8. Hand Registration - MANO

We sample 20 hands from MANO to create an initial shape space. We then iteratively register scans from the MANO dataset using the pre-trained NJF *without* assuming paired data. Since NJF is a topologically invariant mapping module, we are able to train the network for full body registrations, and use the trained MLP to register out-of-domain scans. Registrations are shown in Figure 9.

Figure 8. **Iterative Shape Corrections:** Lightly colored faces in the middle are our registrations in earlier iterations of the space, and the pink-colored face on the right is our registration after five rounds of BLiSS. As the rounds progress, registrations in later rounds more accurately capture the scan (left, in Blue), as observed by the broadening of the nose and jawline.

Figure 9. **Registering Hand scans:** We use our final hand-shape space to register hand scans (blue); registrations in canonical pose (i.e., default) shown in pink.

9. Implementation Details

9.1. Features for Neural Jacobian Field

At the core of NJF is a Multi-layer Perceptron (MLP) that processes the input features on each triangle of the given mesh to produce a per-triangle Jacobian, which is used in a differentiable Poisson solve to compute the deformed vertex positions. We use NJF to deform our PCA projection

X_o conditioned on the scan S_X, similarly to the shape morphing experiments proposed in the original paper [2]. In our tests, we used the following features:

- Since S_X is a raw scan, the PointNet encoding of its coordinates represents it. We use both the global encoding of the scan and its per-point features from PointNet. Since the scan and X_o are *not* in correspondence, we choose features of those points that are closest to a point on X_o. We note that despite X_o and S_X having different poses, the nearest neighbor feature look-up provides an indication to the MLP of the kind of shape transformation that is required.

- We associate with each triangle of X_o, the PointNet encoding of S_X and Ss_X's points as obtained above, and of X_o itself. Specifically for X_o as in the original work, we encode each triangle's centroids, normals and top 50 Wave Kernel Signatures [4]. In addition, we also pass the Jacobians of X_o itself as we observe that it significantly improves the mapping to target. Note that S_X and X_o are processed via different PointNets as their input features are different

We pass the above features to a 4-layer MLP, with each hidden layer being 128 wide and activated by ReLU. The final Linear layer produces a 9-dimensional vector for each triangle (as a Jacobian is a 3×3 matrix). Both PointNets - one for X_o and one for S_X - and the MLP are trained jointly to produce the mapping from X_o to the desired shape.

9.2. Summary of the Use of CAESAR data

We have registration information for 429 of all 4000 CAESAR scans. We use 100 of them for training the initial PCA shape space and another 100 for NJF, withholding the rest 229 for evaluation purposes, *i.e.*, $|\mathcal{R}_{PCA}| = |\mathcal{R}_{DEFORM}| = 100$, $|\mathcal{R}_{EVAL}^{\text{FULL}}| = 229$ or $|\mathcal{R}_{EVAL}^{\text{SMALL}}| = 29$. In each round of Algorithm 1, we sample random 100 among the rest of ~ 3.5k (=4000-429) unregistered shapes $\mathcal{U}$, bring it into correspondence and move it to $\mathcal{R}_{PCA}$ when selected by our pruning.

10. Nearest Lookup from Others to GHUM

In Table 3, we report a v2p error of 2.63 cms on 229 scans, while GHUM [57] report 1.91 cms (Chamfer distance) on the entire CAESAR [45] dataset. This difference may arise from the difference in the evaluation set and the different training sets. Further, we observe in Table 4 of the main paper, that the numbers reported with respect to GHUM are higher compared to others. In Figure 10, we show samples from each of SMPL, STAR and BLiSS and their corresponding nearest sample by vertex-to-vertex L2 in GHUM's shape space. Each space was randomly sampled to generate 10000 shapes, and we perform pairwise L2 distance across shapes in different spaces. We observe significant differences in body-shape indicating that there are non-overlapping regions between GHUM and other shape spaces.
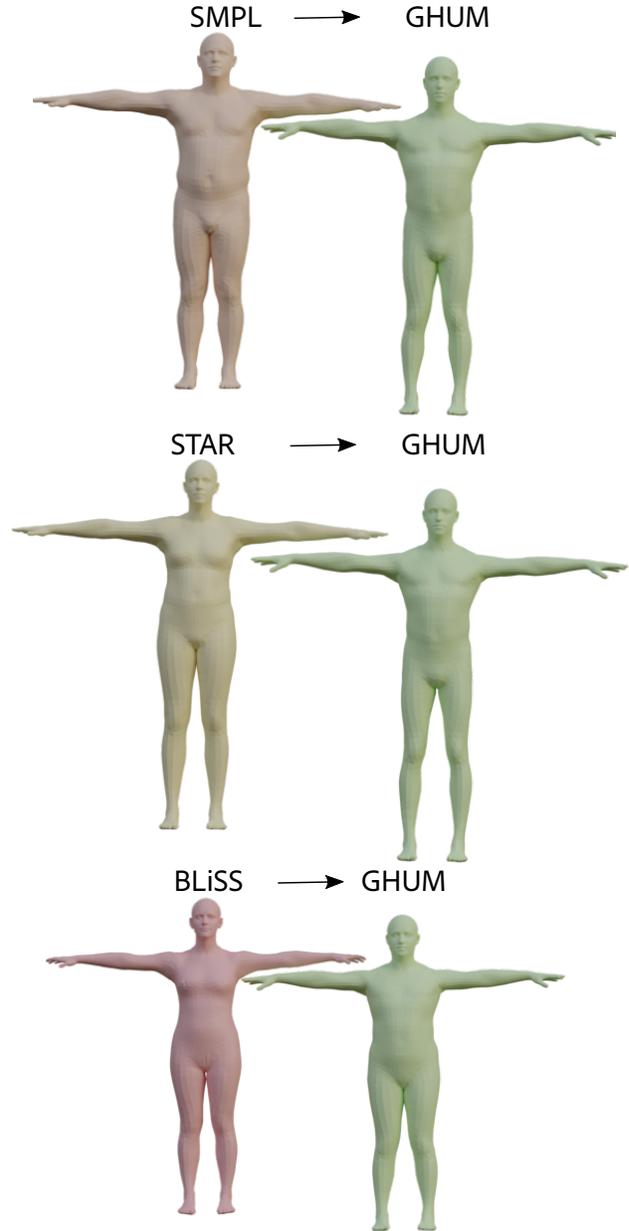

Figure 10. For a sample in each of SMPL, STAR and BLiSS, we lookup the nearest shape in GHUM's space (green). We randomly sampled 10000 shapes in each space.