Learning an Efficient Model of Hand Shape Variation from Depth Images

Sameh Khamis[1,2] Jonathan Taylor[2] Jamie Shotton[2]
Cem Keskin[2] Shahram Izadi[2] Andrew Fitzgibbon[2]
[1]University of Maryland [2]Microsoft Research

Abstract

We describe how to learn a compact and efficient model of the surface deformation of human hands. The model is built from a set of noisy depth images of a diverse set of subjects performing different poses with their hands. We represent the observed surface using Loop subdivision of a control mesh that is deformed by our learned parametric shape and pose model. The model simultaneously accounts for variation in subject-specific shape and subject-agnostic pose. Specifically, hand shape is parameterized as a linear combination of a mean mesh in a neutral pose with a small number of offset vectors. This mesh is then articulated using standard linear blend skinning (LBS) to generate the control mesh of a subdivision surface. We define an energy that encourages each depth pixel to be explained by our model, and the use of a smooth subdivision surface allows us to optimize for all parameters jointly from a rough initialization. The efficacy of our method is demonstrated using both synthetic and real data, where it is shown that hand shape variation can be represented using only a small number of basis components. We compare with other approaches including PCA and show a substantial improvement in the representational power of our model, while maintaining the efficiency of a linear shape basis.

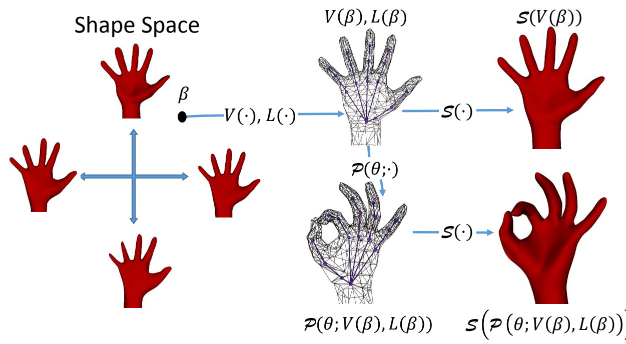

Figure 1. Our deformable surface model takes into account both pose (via an animation-ready kinematic model) and shape (in a shape space learned to best compose with the kinematic model). A set of shape parameters $\beta \in \mathbb{R}^K$ in shape space (left) specifies (upper center) a neutral mesh $V(\beta) \in \mathbb{R}^{3\times M}$ and skeleton parameters $L(\beta) \in \mathbb{R}^{3\times B}$. A set of joint angles θ deforms the mesh to obtain a specific posed mesh $\mathcal{P}(\theta; V(\beta), L(\beta)) \in \mathbb{R}^{3\times M}$ (bottom left) using the linear blend skinning function $\mathcal{P}(\cdot)$. A subdivision surface function $\mathcal{S}(\cdot)$ maps these meshes to smooth 3D surfaces (right column). Simultaneously optimizing the parameters on the full pipeline from joint angles to 3D shape gives the parameters that best relate the end-to-end model to sparse and noisy real data.

1. Introduction

Morphable models of the human body have been a great success story of computer vision and graphics. Starting from the face models of Blanz and Vetter [7], and proceeding to combined shape and pose models of the full body [5, 13, 15, 12], such models are now starting to see commercial deployment for applications including virtual shopping (*e.g*. Metail), performance capture (*e.g*. faceshift), and video gaming (*e.g*. Kinect Sports Rivals).

However, to our knowledge, no morphable model of the human hand has yet been constructed. The hand is in some senses ideal for such modeling: it is normally unclothed, and has huge potential for natural 3D user interfaces. Ballan *et al*. [6] demonstrate that extremely robust hand tracking is possible given a user-specialized hand model, but acquiring the model requires manual rigging and a multi-camera capture setup. Taylor *et al*. [24] demonstrate acquisition of a user-specialized model from a single depth camera, but require long calibration sequences in which all degrees of freedom of the hand have to be exercised.

Our hypothesis is that the absence of a morphable model for hands is because existing techniques for model construction depend on large datasets of high quality scans. Even though the number of degrees of freedom of the hand is similar to that of the body, hands tend to exhibit considerably more self-occlusion, and so such scans have more holes. Further, hands are smaller, so images of hands typically contain fewer foreground pixels and camera noise will have a larger effect. Finally, the space of hand poses may be larger, despite the number of joints being about the same, so that more captures would be required to allow one to accurately learn the pose space.

In this paper, we overcome these challenges and build a morphable model of hands from a set of short 'sequences' (we use a diversely-posed set of 15 frames from each subject) obtained from fifty different people using a single Kinect V2 sensor. The keys to our approach are twofold.

First, we learn only those aspects of pose and shape that are not explained by a standard rigged model. This reduces the data requirements, but also has the advantage that the output of our system is a standard subdivision surface model driven by a linear blend skinning. This ensures our approach can be evaluated extremely efficiently. In contrast, models such as SCAPE [5] and TenBo [10] involve an additional linear solve at test time, which, while readily implementable on GPUs, does represent significant additional computational cost.

Second, we fit the model jointly to all partial scans, rather than attempting to separately build complete scans per subject and performing principal component analysis (PCA). As we show in experiments on synthetic and real data, this yields a better model even for unoccluded synthetic data, and a much better model with real scans that contain missing and noisy data.

Our main contribution is thus a new technique for learning efficient skeleton-driven morphable models from sparse and noisy depth data. The learned models include a parameterized set of basis meshes as well as a parametrization of skeleton parameters such as bone lengths and skinning weights. While previous work has learned some of these parameters, this paper is the first that learns all parameters jointly, and the first to learn by direct explanation of the captured data.

1.1. Related Work

Learning a lower dimensional parametrization of shape from examples of range scans or other 3D data has proved effective in creating generic morphable models for human bodies and faces [7, 4, 5, 13, 10, 25, 17, 23, 26, 16, 27, 19]. Having built such morphable models, impressive applications *e.g.* for fitting body shape to monocular depth sequences or more precise body or face tracking have been demonstrated [7, 11, 14].

However, despite a long history of successes for faces and bodies, we are not aware of any existing statistical shape models for hands. This suggests that this is a challenging problem, where existing techniques for whole bodies [4, 5, 13, 10] do not directly transfer.

Similar to our work, Allen *et al.* [3] represent the model as an adaptation of a standard subdivision surface model with linear blend skinning. Crucially, however, their adaptations are displacement maps on top of a base surface. The displacements must be limited in magnitude to avoid self-intersections, and their shape basis is forced to coincide with the input scans. Also, their optimization steps are sequential (block coordinate descent) rather than simultaneous, which may result in a poor local optimum being obtained. Cashman and Fitzgibbon [9] demonstrate that morphable models using subdivision surfaces can be learned from extremely limited data (30 silhouette images). However their approach does not separate shape and pose, and neither learns a parametric shape basis.

Specific to hands, Rhee *et al.* [20] extract creases visible from a single frontal image of a hand under controlled illumination, localize joints, and fit a 3D model with user-specific skinning. This model is fit on a single image, resulting in very simplistic hand models with limited degrees of freedom. Albrecht *et al.* [2] go to the other extreme creating very detailed, physically-realistic hand models. However, the process is laborious requiring plaster casting of human hands, performing laser scans, and manually creating a physics-enabled hand model.

A more automatic technique is presented by Taylor *et al.* [24], which generates *personalized* hand models given noisy input depth sequences where the user's hand rotates 180° whilst articulating fingers. A continuous optimization that jointly solves for correspondences and model parameters across a smooth subdivision surface with as rigid as possible (ARAP) regularization leads to high-quality user-specific rigged hand models, though not a shape basis. Whilst the process is automatic, the hands are required to cover the full range of articulations, and longer sequences are required, leading to more complex capture requirements and more costly optimization.

While not explicitly explored in this paper, we hypothesize that hand shape will prove an important prior for robust hand pose estimation, much in the way that it has been shown for whole body tracking [14]. Studies of the anatomical structure of adult hands has shown considerable variation [8], which is clearly apparent across gender and age. Recent work on high-quality, offline, performance capture of hands, using multi-camera rigs, reaffirms our intuition regarding the importance of user-specific hand shape for pose estimation. Ballan *et al.* [6] construct a personalized hand mesh using a multiview camera rig and Poisson surface reconstruction, which is then manually skinned. They demonstrate high-quality results with complex two-handed and hand-object interactions, closely fitting the detailed mesh model to the data. However, this system focuses on pose estimation as opposed to the shape construction, which is performed in an time consuming manual manner.

2. Model

We now describe our deformable hand surface model. We will use triangular meshes extensively as a fundamental primitive. All meshes discussed in this paper represent a human right hand (although there is nothing otherwise hand-specific about our model), and will contain exactly M vertices and use a fixed triangulation. We thus represent such a mesh as a matrix $V \in \mathbb{R}^{3\times M}$ where the m^{th} column contains the location of the m^{th} vertex.

As explained in more detail below, we also employ a hand skeletal structure comprising B bones and use a matrix $L \in \mathbb{R}^{3\times B}$ to represent the locations of these bones. Once again, the b^{th} column represents the location of bone b. These bones are arranged in a fixed hierarchy with bone $b = 1$ being the root bone and the π_b denoting the index of the parent for any other bone $b \in \{2, ..., B\}$

As illustrated in Figure 1, our shape model linearly parameterizes the shape of a hand mesh $V(\beta)$ and skeleton $L(\beta)$ in a neutral pose using a set of shape parameters $\beta \in \mathbb{R}^K$. Our pose model accounts for articulation out of the neutral pose by mapping a neutral hand mesh and skeleton to a posed hand mesh $\mathcal{P}(\theta; V(\beta), L(\beta)) \in \mathbb{R}^{3\times M}$. Finally, our surface model uses loop subdivision to map this posed 'control mesh' to a smooth surface $\mathcal{S}(\mathcal{P}(\theta; V(\beta), L(\beta)) \subset \mathbb{R}^3$. In the remainder of this section, we detail the exact form of these functions.

2.1. Shape Model

Our shape model follows our intuition that the variation in the shape of a human hand (and skeleton) in a *single pose* is relatively compact and can be described by a low dimensional linear subspace. We therefore parameterize this space using a set of basis mesh matrices $\mathcal{V} = \{V_k\}_{k=1}^K \subset \mathbb{R}^{3\times M}$ and basis bone location matrices $\mathcal{L} = \{L_k\}_{k=1}^K \subset \mathbb{R}^{3\times B}$. It is not coincidence that we use the same number of dimensions K for both bases, as we enforce that skeletal and skin shape vary together. In particular, given a vector of shape parameters $\beta \in \mathbb{R}^K$, a neutral mesh

$$V(\beta; \mathcal{V}) = \sum_{k=1}^{K} \beta_k V_k \quad (1)$$

and a neutral skeleton

$$L(\beta; \mathcal{L}) = \sum_{k=1}^{K} \beta_k L_k \quad (2)$$

is recovered as a linear combination of these basis matrices.

Further, we believe that it is intuitive to have the first basis component V_1 and L_1 represent a 'mean' mesh and skeleton with the other basis components representing small offsets. Ideally then, β_1 should (approximately) encode scale, and the other coordinates in β should encode how much of the offset vectors to use. We do not explicitly enforce these desires, but instead employ regularizers (see Sections 3.1.2 and 3.1.3) to encourage this.

Our linear model has the same representational power as PCA, though differs substantially in how it is learned (see below). Compared to models such as [5], our approach is potentially considerably more efficient in both memory and compute.

2.2. Pose Model

A key feature of our model, is that we don't require the aforementioned linear *shape* model to account for any hand surface deformation resulting from non-neutral *poses*. Instead, we explicitly parameterize pose using a vector θ concatenating a set of joint angles (see Figure 2), global orientation and translation. Our pose model, defined in this section, specifies the articulated deformation that θ invokes on a mesh $V(\beta)$ in a neutral pose using the corresponding skeleton $L(\beta)$. Although many formulations are possible, we use linear blend skinning (LBS) as it is both common place and extremely efficient.

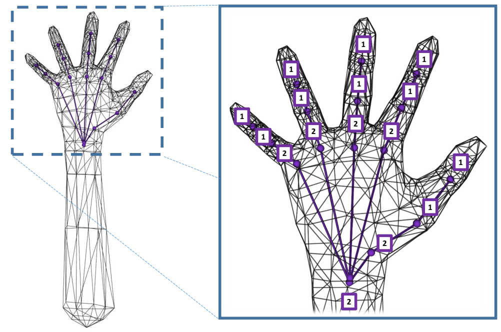

Figure 2. Our template hand model contains a mesh and kinematic skeleton. This defines the mesh topology and skeletal structure of our learned model, and provides a weak regularization on the shape model during optimization. As such the template does not need to be anatomically accurate: the learned model will adjust both the vertices and the skeleton to ensure a precise reconstruction of the observed hand surface. On the right, we have labeled the number of degrees of freedom each joint has.

For clarity, we momentarily drop the dependence on shape parameters β and demonstrate how LBS deforms a fixed mesh $V = \begin{bmatrix}\mathbf{v}_1 & \dots & \mathbf{v}_m\end{bmatrix}$ and skeleton $L = \begin{bmatrix}\mathbf{l}_1 & \dots & \mathbf{l}_b\end{bmatrix}$. This model requires that each bone b is further endowed with a fixed rotation matrix Q_b which indicates the principal axes of rotation of the bone's joint. Together with the bone location $\mathbf{l}_b$, this defines a coordinate system

$$H_b = \begin{bmatrix} Q_b & \mathbf{l}_b \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \quad (3)$$

that maps from bone space to world space via a rigid transform. Note also that the set $\mathcal{H} = \{H_b\}_{b=1}^B$ implicitly defines an equivalent set of transformations $\{T_b\}_{b=1}^B$, where T_b maps from bone b's coordinate system to its parent π_b's, and where $H_b = T_1 ... T_{\pi_b} T_b$.

Given a set of pose parameters θ, linear blend skinning articulates a joint by applying a 3D rotation $\tilde{R}_b$, in the form of a homogeneous rotation matrix

$$R_b(\theta) = \begin{bmatrix} \tilde{R}_b(\theta) & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \quad (4)$$

to each bone b's local coordinate system. In addition, global orientation and position is accounted for by a global rigid transformation $R(\theta) \in \mathbb{R}^{4\times 4}$ that is applied to the world. For each bone b, we thus obtain a rigid transformation

$G_b(\theta)$ relating bone b's local coordinate system to the world under pose θ via the following recurrence:

$$G_1(\theta) = R(\theta) T_1 R_1(\theta) \quad (5)$$
$$G_b(\theta) = G_{\pi_b}(\theta) T_b R_b(\theta) \,. \quad (6)$$

The mesh itself is 'skinned' to the bones by a set of skinning weights $\Gamma = \{\alpha_{bm} : b \in \{1, ..., B\}, m \in \{1, ..., M\}$, where $\sum_{b=1}^{B} \alpha_{bm} = 1$ for all m. Intuitively, α_{bm} indicates the strength of attachment of vertex m to bone b when the latter moves. More precisely, the articulated position under pose θ of the m^{th} vertex in V is

$$I_{3\times4} R(\theta) \sum_{b=1}^{B} \alpha_{bm} G_b(\theta) H_b^{-1} \hat{\mathbf{v}}_m \quad (7)$$

where $I_{3\times4}$ projects to Euclidean coordinates and $\hat{\mathbf{v}}_m = \begin{bmatrix} \mathbf{v}_m^\top & 1 \end{bmatrix}^\top$. Note that H_b^{-1} first maps the vertex $\mathbf{v}_m$ from world coordinates in the neutral pose to the local coordinate system of bone b. The transformation $G_b(\theta)$ then maps the vertex back to world coordinates under pose θ.

For notational convenience, we now create the triple $\Upsilon = (\mathcal{V}, \mathcal{L}, \Gamma)$ of pose-invariant shape parameters that we have so far introduced.[1] Reintroducing the shape parameters β, the parameters required by the LBS model are $\Phi(\beta; \Upsilon) = (V(\beta), L(\beta), \Gamma)$ and we thus write $\mathcal{P}(\theta; \Phi(\beta; \Upsilon)) \in \mathbb{R}^{3\times M}$ to indicate the resultant mesh by applying pose θ to the neutral hand mesh of shape β.

2.3. Surface Model

Following [24], we represent the actual surface of our model using Loop subdivision of a control mesh [18]. Given a mesh V, the Loop subdivision procedure works by iteratively subdividing each triangle face (from the fixed triangulation) and smoothing the vertex positions with their neighbors. The 'limit surface' $\mathcal{S}(V) \subset \mathbb{R}^3$ would be obtained by performing this subdivision procedure an infinite number of times.

In order to avoid this complicated construction, we follow [24] by instead parameterizing our surface using

$$\mathcal{S}(\mathbf{u}; V) : \Omega \times \mathbb{R}^{3\times M} \mapsto \mathbb{R}^3 \,, \quad (8)$$

which maps from a location $\mathbf{u}$, in an essentially 2D space Ω of surface coordinates, to a point on the 3D subdivision surface. With this definition, the full surface can be written $\mathcal{S}(V) = \{\mathcal{S}(\mathbf{u}; V) : \mathbf{u} \in \Omega\}$. Due to space limitations, we refer the reader [24] for the precise details of function $\mathcal{S}(\cdot)$ and the parameterization of $\mathbf{u}$. However, it suffices for our purposes that $\mathcal{S}(\cdot)$ and its derivatives with respect to $\mathbf{u}$ and V can be efficiently computed. We compose $\mathcal{S}(\cdot)$ with the rest of our model allowing us to produce $\mathcal{S}(\mathbf{u}; \mathcal{P}(\theta; \Phi(\beta; \Upsilon))$, the 3D location of the coordinate $\mathbf{u}$ on the hand with shape β and pose θ.

[1] We exclude the parameters $\{Q_b\}_{b=1}^{B}$, which are held fixed.

2.4. Full Model

To summarize, when a particular set of shape parameters β is chosen, we obtain the subject specific parameters $\Phi(\beta; \Upsilon)$ of a LBS hand model. We can then obtain a mesh with shape β in pose θ as $\mathcal{P}(\theta; \Phi(\beta; \Upsilon))$. Finally we can obtain the position of a coordinate $\mathbf{u} \in \Omega$ on the surface of the subdivision surface in pose θ with shape β as $\mathcal{S}(\mathbf{u}; \mathcal{P}(\theta; \Phi(\beta; \Upsilon)))$. Our desire then, is to learn a setting of Υ so that β and θ alone can be used to describe the majority of feasible human hand shapes and poses.

3. Fitting the Model

A major contribution of this work is showing how to learn the parameters Υ from a set of noisy depth images of users' hands. To this end, we assume that we have a diverse set (*i.e.* men, women, and children, with varying hand shapes) of S different subjects. For each subject s, we have F_s depth frames of the user performing various hand articulations. In each frame f, a set of N_{sf} data points $\{\mathbf{x}_{sfn}\}_{n=1}^{N_{sf}} \subset \mathbb{R}^3$ with corresponding estimated normals $\{\mathbf{n}_{sfn}\}_{n=1}^{N_{sf}} \subset \mathbb{R}^3$ is extracted.

3.1. Energy

We want to use this data to learn Υ such that our model can both explain the data and satisfy some straightforward priors. We cast this as the problem of minimizing the energy

$$E(\Upsilon) = \sum_{s=1}^{S} E^s(\Upsilon) + \lambda_{\text{arap}} E_{\text{arap}}(\Upsilon) + \lambda_{\text{skin}} E_{\text{skin}}(\Upsilon) \quad (9)$$

defined over the variables in Υ. The latter two weighted prior terms regularize the basis representation and skinning weights, and are described below. Each subject specific term

$$E^s(\Upsilon) = \min_{\beta} \sum_{f=1}^{F_s} E^{sf}(\beta; \Upsilon) + \lambda_{\text{shape}} E_{\text{shape}}(\beta) \quad (10)$$

provides constraints on Υ based on the data from subject s. The second term in (10) encodes a shape prior penalty, while each term

$$E^{sf}(\beta; \Upsilon) = \min_{\theta} \sum_{n=1}^{N_{sf}} E_{\text{data}}^{sfn}(\theta, \beta; \Upsilon) + \lambda_{\text{pose}} E_{\text{pose}}(\theta) \,,$$

measures how well the posed surface is at explaining the data in frame f.

3.1.1 Data term

The data term that we use is

$$\begin{aligned} E_{\text{data}}^{sfn}(\theta, \beta; \Upsilon) = & \quad (11) \\ & \min_{\mathbf{u}\in\Omega} \rho(\|W Q_{sfn}(\mathbf{x}_{sfn} - \mathcal{S}(\mathbf{u}; \mathcal{P}(\theta; \Phi(\beta, \Upsilon))))\|) \\ & + \lambda_{\text{normal}}\, \rho^{\perp}(\|1 - (\mathbf{n}_{sfn})^\top \mathcal{S}^{\perp}(\mathbf{u}; \mathcal{P}(\theta; \Phi(\beta, \Upsilon)))\|) \end{aligned}$$

where $\rho(e)$ and $\rho^{\perp}(e)$ correspond to robust kernels applied to the point position error and the squared normal error respectively. We set the scaling matrix $W = \text{diag}(1, 1, \zeta)$ which, combined with the rotation Q_{sfn} that rotates the 3D residual so that the line of sight to $\mathbf{x}_{sfn}$ aligns with the z-axis, models the depth sensor's relatively high uncertainty in the viewing direction.

3.1.2 As-Rigid-As-Possible Regularization

The term $E_{\text{arap}}(\Upsilon)$ invokes the 'As-Rigid-As-Possible' (ARAP) [22] deformation assumption to regularize $\mathcal{V}$ and $\mathcal{L}$. We use ARAP to define the regularization energy as

$$E_{\text{arap}}(\Upsilon) = D(V_1, V_{\text{template}}) + D^{\dagger}(L_1, V_1, L_{\text{template}}, V_{\text{template}}) + \sum_{k=2}^{K} \left(D(V_k, \varnothing) + D^{\dagger}(L_k, V_k, \varnothing, \varnothing) \right) \quad (12)$$

where V_{template} and L_{template} are the mesh and bone locations from our coarse template hand model (Figure 2), and $\varnothing$ is a matrix of an appropriate size filled with zeros. The terms in (12) taking only two arguments are the standard ARAP measure of deformation $D(V, V')$ between two meshes V and V' with vertex positions $\{\mathbf{v}_m\}_{m=1}^M$ and $\{\mathbf{v}'_m\}_{m=1}^M$ in $\mathbb{R}^{3\times M}$ is defined as

$$\sum_{m=1}^{M} \min_{R \in \text{SO}(3)} \sum_{n \in \mathcal{N}(m)} \|(\mathbf{v}_n - \mathbf{v}_m) - R(\mathbf{v}'_n - \mathbf{v}'_m)\|^2 , \quad (13)$$

where $\mathcal{N}(m)$ is the set of vertices neighboring vertex m.

Under ARAP, rigid transformations are not penalized, and smaller (localized) non-rigid transformations are penalized less than larger non-rigid transformations. Note that

$$D(V, \varnothing) = \sum_{m=1}^{M} \sum_{n \in \mathcal{N}(m)} \|(\mathbf{v}_n - \mathbf{v}_m)\|^2 , \quad (14)$$

which simply encourages neighboring vertices to coincide. In our case, for $k \geq 2$, V_k is meant to represent offsets from the 'mean' mesh V_1, and thus this translates into our desire that the vertex offset field be smooth.

The terms in (12) that take four arguments employ a modified version of ARAP that encourages the bone locations in the core meshes to remain consistent relative to a set of nearby vertices (typically a vertex ring). We denote the set of vertex indices as $C_b \subset \{1, \ldots, M\}$ for each bone b. For a pair of bone location matrices $L, L' \in \mathbb{R}^{3\times B}$ with columns $\{\mathbf{l}_b\}_{b=1}^B$, $\{\mathbf{l}'_b\}_{b=1}^B$ and mesh vertex matrices $V, V' \in \mathbb{R}^{3\times M}$ with columns $\{\mathbf{v}_m\}_{m=1}^M$ and $\{\mathbf{v}'_m\}_{m=1}^M$, $D(L, V, L', V')$ is defined as

$$\sum_{b=1}^{B} \min_{R \in \text{SO}(3)} \sum_{m \in C_b} \|(\mathbf{v}_m - \mathbf{l}_b) - R(\mathbf{v}'_m - \mathbf{l}'_b)\|^2 . \quad (15)$$

Similarly, we note that

$$D^{\dagger}(L_k, V_k, \varnothing, \varnothing) = \sum_{b=1}^{B} \sum_{m \in C_b} \|(\mathbf{v}_m - \mathbf{l}_b)\|^2 , \quad (16)$$

which for basis components $k \geq 2$ equivalently encourages the offsets of a bone to be similar to the offsets of the vertices this bone is anchored to.

3.1.3 Shape Prior

We regularize the shape parameters β using the term

$$E_{\text{shape}}(\beta) = (1 - \beta_1)^2 + \sum_{k=2}^{K} \beta_k^2 \quad (17)$$

which encourages the user-specific hand model to stay relatively similar to the template model with minor vertex and bone location offsets applied.

3.1.4 Pose Prior

We highly penalize any pose deformations that violate human physical constraints by adding barrier constraints on the pose θ using the term

$$E_{\text{pose}}(\theta) = \sum_i \begin{cases} (\theta_i - \theta_i^{\text{min}})^4 & \text{if } \theta_i < \theta_i^{\text{min}} \\ (\theta_i^{\text{max}} - \theta_i)^4 & \text{if } \theta_i > \theta_i^{\text{max}} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where θ^{min} and θ^{max} are approximations of the minimum and maximum rotation angles for the joints of the hand.

3.1.5 Skinning Weight Prior

To ensure that the skinning weights for each vertex m sum to 1, we find that it is sufficient to add another energy $E_{\text{skin}}(\Upsilon) = \sum_{m=1}^{M} \| \sum_{b=1}^{B} \alpha_{bm} - 1\|^2$ penalizing deviations by the large weight λ_{skin} in (9). In order to ensure that the skinning weights remain non-negative, we simply parameterize the weight of vertex m with bone b in the log domain as $\tilde{\alpha}_{bm} = \log(\alpha_{bm})$.

4. Optimization

In order to optimize the energy function of your model, we 'lift' it to a simpler energy, defined by introducing a set of latent variables, that can be optimized using a standard non-linear optimizer.

4.1. Lifted Energy

As defined above, our energy $E(\Upsilon)$ is of a complicated form that contains many summations over minimizations. Following [24], we note that the following is true of two real valued functions $f(x)$ and $g(x)$

$$\min_x f(x) + \min_x g(x) = (\min_{x_1} f(x_1) + \min_{x_2} g(x_2)) = \min_{x_1, x_2} (f(x_1) + f(x_2)) \leq f(x_1) + f(x_2) \quad (19)$$

for any x_1 and x_2. That is, the variables being minimized over in a sum can be labeled and passed through the sum. Our energy can also be 'lifted' in this manner by introducing a set of shape parameters $\mathcal{B} = \{\beta^s\}_{s=1}^S$, poses $\Theta = \{\theta_{sf} : s \in \{1, ..., S\}, f \in \{1, ..., F_s\}\}$, correspondences $\mathcal{U} = \{\mathbf{u}_{sfn} : s \in \{1, ..., s\}, f \in \{1, ..., F_s\}, n \in \{1, ..., N_{sf}\}\}$ and ARAP rotations[2] $\mathcal{R} = \{R_m\}_{m=1}^M \cup \{R_b^\dagger\}_{b=1}^B$. This introduces a new energy $E'(\Upsilon, \mathcal{B}, \Theta, \mathcal{U}, \mathcal{R})$ such that

$$E(\Upsilon) = \min_{\mathcal{B},\Theta,\mathcal{U},\mathcal{R}} E'(\Upsilon, \mathcal{B}, \Theta, \mathcal{U}, \mathcal{R}) \leq E'(\Upsilon, \mathcal{B}, \Theta, \mathcal{U}, \mathcal{R}) \tag{20}$$

for any setting of $\mathcal{B}, \Theta, \mathcal{U}, \mathcal{R}$. We include the full form of this lifted energy in the supplementary material, but imagine here for simplicity a case in which $\lambda_{\text{arap}} = \lambda_{\text{skin}} = \lambda_{\text{shape}} = \lambda_{\text{pose}} = 0$ and that our data term is simply

$$E_{\text{data}}^{sfn}(\theta, \beta, \Upsilon) = \min_{\mathbf{u}\in\Omega} \|\mathbf{x}_{sfn} - \mathcal{S}(\mathbf{u}; \mathcal{P}(\theta; \Phi(\beta, \Upsilon)))\|^2 . \tag{21}$$

Then the lifted energy would be

$$E'(\Upsilon, \mathcal{B}, \Theta, \mathcal{U}, \mathcal{R}) = \sum_{s=1}^{S} \sum_{f=1}^{F_s} \sum_{n=1}^{N_{sf}} E_{\text{data}}^{\prime sfn}(\mathbf{u}_{sfn}, \theta_{sf}, \beta_s; \Upsilon) \tag{22}$$

with a lifted data term

$$E_{\text{data}}^{\prime sfn}(\mathbf{u}, \theta, \beta, \Upsilon) = \|\mathbf{x}_{sfn} - \mathcal{S}(\mathbf{u}; \mathcal{P}(\theta; \Phi(\beta, \Upsilon)))\|^2 . \tag{23}$$

This lifted data term removes the 'inner minimization' over $\mathbf{u} \in \Omega$.

4.2. Non-Linear Optimization

We use Levenberg-Marquardt to optimize this energy, and exploit the Ceres solver [1] to automatically deal with the large, but dynamic, sparsity in the problem. Our optimization schedule (see below) will make use of a subroutine NONLINEAR(N, $\mathcal{F}$) that attempts to take N Levenberg-Marquardt steps optimizing all parameters except for those in $\mathcal{F} \subset \{\Upsilon, \mathcal{B}, \Theta, \mathcal{U}, \mathcal{R}\}$, an operation supported by Ceres.

4.3. Discrete Update

To help jump out of local minima, we also make use of a subroutine DISCRETEUPDATE() that attempts to improve the correspondences $\mathcal{U}$ by searching over a discrete set of candidates. Specifically, we consider a proposed set of samples $\mathcal{U}_{\text{prop}} = \mathcal{U} \cup \mathcal{U}_{\text{samp}} \subseteq \Omega$ where $\mathcal{U}_{\text{samp}}$ is a fixed set of surface coordinates, sampled roughly uniformly over the domain Ω. We then consider performing a loop over subject s, frame f and data point n to find a new surface coordinate

$$\mathbf{u}'_{sfn} = \arg\min_{\mathbf{u}\in\mathcal{U}_{\text{prop}}} E_{\text{data}}^{\prime sfn}(\mathbf{u}, \theta_{sf}, \beta^s, \Upsilon) . \tag{24}$$

The resulting set $\mathcal{U}' = \{\mathbf{u}'_{sfn}\}$ is guaranteed to not increase the energy (*i.e.* $E'(\Upsilon, \mathcal{B}, \Theta, \mathcal{U}', \mathcal{R}) \leq E'(\Upsilon, \mathcal{B}, \Theta, \mathcal{U}, \mathcal{R})$).

[2]The rotations in the ARAP regularizers fall out for any terms involving basis components $k \geq 2$, and thus need not be parameterized.

4.4. Initialization

We manually initialize the poses Θ so that the template roughly aligns with the point clouds. Although one could consider automated methods, the task was not overly onerous and needs only to be performed once. Similarly, each $\beta^s \in \mathcal{B}$ is initialized so that β_1^s corresponds to the rough scale of subject s and $\beta_k^s = 0$ for all $k \geq 2$. We initialize V_1 and L_1 using our rough hand template (see Section 5), and initialize the other basis components using zero mean noise. All ARAP rotations in $\mathcal{R}$ are initialized to the identity and $\mathcal{U}$ with a call to DISCRETEUPDATE().

4.5. Optimization Schedule

After initialization, we then perform a scheduled optimization (see Algorithm 1) that interleaves discrete updates with continuous optimization, while gradually unfreezing parameters. We found that ordering the various 'stages' in this way made the algorithm quite robust in finding a good minimum, and as such, the exact timing of the switches from stage to stage mattered little.

Algorithm 1 Optimization Schedule

```
SUBOPTIMIZE(4, B ∪ Γ)
SWITCHTOGM()              ▷ Switch to Geman McClure
                            robust error function
SUBOPTIMIZE(4, B ∪ Γ)
SUBOPTIMIZE(4, Γ)
SUBOPTIMIZE(4, ∅)
function SUBOPTIMIZE(N, F)
    for i=1:N do
        NonLinear(25, F)
        DiscreteUpdate()
```

5. Evaluation

We now describe the setting and the various experiments performed to evaluate our approach.

Hand template. We rigged the template hand model by hand, using the 3D modeling software Blender. The template mesh comprises 452 vertices, and the skeleton contains 21 bones. See Figure 2.

Parameter settings. The robust kernels for the data terms $\rho(e)$ and $\rho^\perp(e)$ are initially set to the Cauchy kernel $\rho(e; \sigma) = \sigma^2 \log(1 + e^2/\sigma^2)$ which is moderately robust to outliers. We then switch to the extremely robust Geman-McClure kernel $\rho(e, \sigma) = e^2/(e^2 + \sigma^2)$ to avoid fitting to most outliers in the data once the parameters are reasonably close to a good solution. See Algorithm 1.

5.1. Datasets

To evaluate and compare our method, we use three datasets: (i) SYNTHETIC3D, a synthetic dataset containing 3D data point clouds covering the hand surface; (ii) SYNTHETIC2.5D, a synthetic dataset of depth images; and (iii)

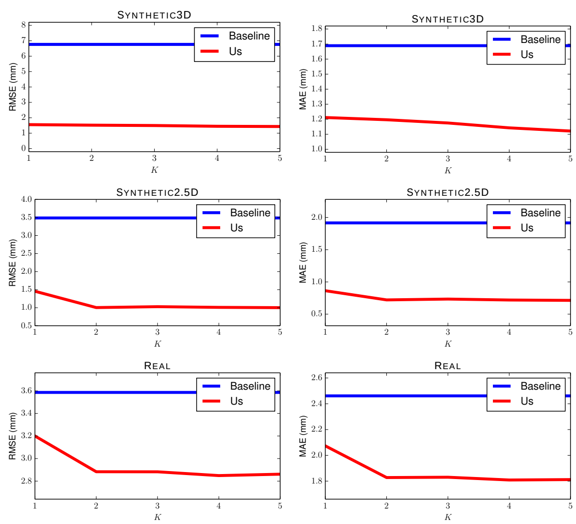

Figure 3. RMSE (left) and MAE (right) for the baseline (blue) and our model (red) with K basis components.

REAL, a real dataset of depth images extracted from the Kinect V2 sensor.

SYNTHETIC3D is generated using the popular modeling tool Poser [21]. The rigged hand model in Poser supports more than 100 blend shapes that can be used to generate a large variety of different and realistic hands. We thus randomly sampled weights for 50 distinctly shaped hands with 15 different poses each. In this dataset, all 3D vertices are used, even if some of them would be occluded in a real depth camera. SYNTHETIC2.5D, on the other hand, is generated by projecting the 3D data of SYNTHETIC3D using a virtual depth camera at a fixed location, removing any points not directly visible to the camera. Both SYNTHETIC3D and SYNTHETIC2.5D are noise-free data, which allows us to test the expressiveness of the model without worrying too much about getting stuck in local minima. We also investigate adding artificial noise to test its effect on the fitting process.

REAL was acquired using a Kinect V2 time of flight sensor. We recorded a diverse set of 50 different subjects: 17 women, 31 men, and 2 children, where each subject was asked to perform varied hand articulations in front of the depth camera. We selected 15 diverse hand poses for each subject on average. Unlike the synthetic datasets, REAL contains a very considerable amount of noise and outlier pixels due to depth discontinuities ('flying pixels') and multipath interference (see Figure 5).

5.2. Baseline

We compare against a baseline approach based on the 'personalization' procedure detailed in [24]. By separately applying personalization to S subjects, one can obtain a set of personalized meshes $\{V^s\}_{s=1}^S$, skeletons $\{L^s\}_{s=1}^S$ and scales $\{\beta_0^s\}_{s=1}^S$. For subject s, we can concatenate and flatten these matrices and remove the scale as $p^s = \frac{1}{\beta_0^s}\left(\vec{V^s}^\top\ \vec{L^s}^\top\right)^\top$. By applying PCA to the vectors $\{p^s\}_{s=1}^S$, we obtain a mean vector $\bar{p}$ and a set of principle directions $\{p_k\}$. In particular, each of the input meshes p, has a corresponding vector α^s, such that

$$p^s = \bar{p} + \sum_k \alpha_k^s p_k \, . \tag{25}$$

Note that if we truncate at K PCA directions this minimizes

$$\min_{\{p^s\}_{s=1}^S} \| \min_\alpha \left(\bar{p} + \sum_{k=1}^K \alpha_k p_k - p^s\right) \|^2 \, . \tag{26}$$

In contrast, our model minimizes the 3D error between the observed points and the model surface, through the subdivision surface, the skinning, and the linear shape basis.

5.3. Results

Across all datasets we used 30 subjects for training (learning the shape basis parameters Υ jointly with per-subject shape coefficients β^s and per-frame poses θ_{sf}) and 20 for testing (optimizing for the β^s and θ_{sf} parameters while keeping Υ fixed). All the reported quantitative and qualitative results, including the plots, are on this held-out test set.

Quantitatively, we calculate the 3D residuals between each data point in the test set and the surface and summarize these values using the root mean squared error (RMSE) and the mean absolute error (MAE). Since both SYNTHETIC3D and SYNTHETIC2.5D lack any noise, RMSE is a reasonable metric. However, for REAL MAE is significantly more robust to outliers in the data. Error values are reported in millimeters (mm).

In Figure 3 we report the RMS and MAE errors on the three datasets, showing the effect of the number of basis components. For the PCA baseline we fix K so as to explain at least 90% of the variance in the training meshes. This resulted in $K = 4$ for REAL and SYNTHETIC2.5D and $K = 5$ for SYNTHETIC3D, including the mean vector. Our model clearly outperforms the baseline even with $K = 1$. Additional basis components lower the error rate, but the accuracy appears to saturate beyond $K = 3$.

In Figure 4 we report the percentage of points with a squared error under a given threshold at convergence. Similarly, we outperform the baseline even with $K = 1$, despite the baseline using 4 or 5 basis components. Our accuracy improves with additional basis components but saturates beyond about $K = 2$.

We illustrate some qualitative fitting results on the REAL dataset in Figure 5. This also shows the diversity of hand shapes and poses our model can handle, as well as some of the outliers that it is robust to. Figure 6 shows the shape coefficients for all the subjects on the REAL dataset projected onto their first two principal components.[3] Figure 7 visual-

[3]NB this PCA over the β^s vectors is purely for visualization purposes and has nothing to do with the baseline.

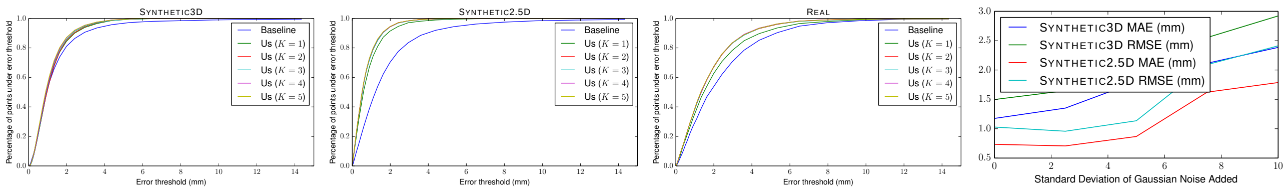

Figure 4. Left three panels: the percentage of data points whose error lies under a threshold for the baseline and our model with K basis components. Right: controlled noise experiments on the synthetic datasets.

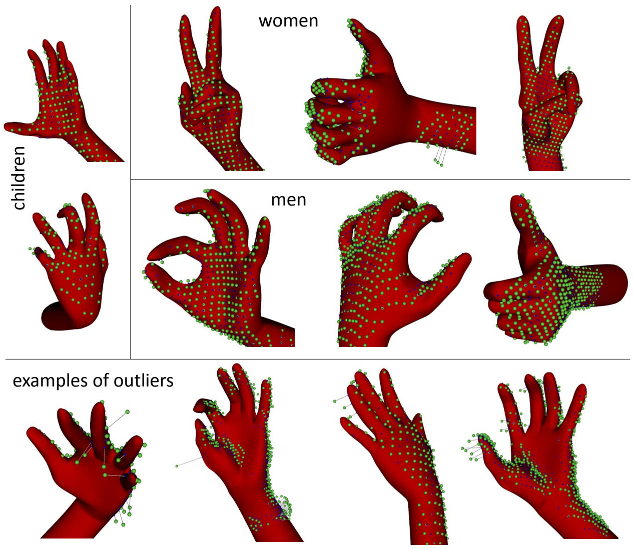

Figure 5. Our surface (red) model fit to data from several frames. Each data point $\mathbf{x}_{sfn} \in \mathbb{R}^3$ (green) has an associated surface point $S(\mathbf{u}_{sfn}; ...)$ (blue). The 'outlier' examples show robust fitting in the presence of considerable noise.

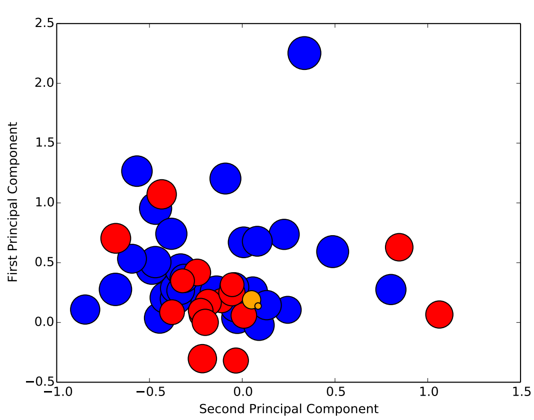

Figure 6. The set of shape coefficients $\{\beta_s\}_{s=1}^S$, learned when our model is fit to the entire REAL dataset, projected onto the first two principal directions. The size of each point encodes the scale coefficient (*i.e.* β_1). One can see that children (orange) have smaller hands while male subjects (blue) typically have larger hands.

izes the first two offset basis components ($k = 2, 3$) through the magnitudes of the vertex offsets. The basis component on the left has a larger effect on the index finger and wrist, while the basis component on the right affects the fingers

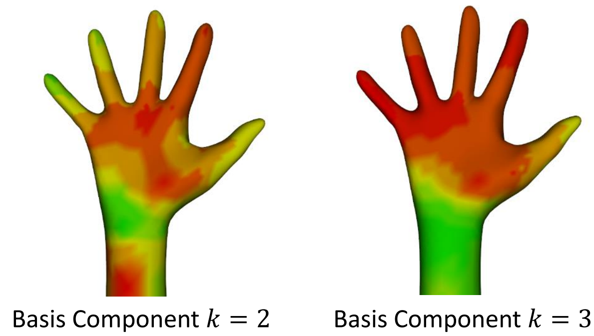

Figure 7. For the first two offset basis components learned on the REAL dataset, we color the surface by vertex offset magnitude. Green and red indicates small large offsets respectively.

farthest from the thumb, widening the entire hand.

Finally, we performed controlled noise experiments (see Figure 4) using the synthetic datasets SYNTHETIC3D and SYNTHETIC2.5D. We used a fixed number of basis components $K = 4$, but varied the standard deviation of the 3D Gaussian noise we added. Noise was added only to the 'training' data in this experiment, to show how robust our optimization scheme is.

6. Discussion

We have shown how a skeleton-driven morphable model can be learned from sparse and noisy data, and considerably outperform a baseline approach. To our knowledge, this is the first instance of a parametric shape and pose model for human hands. Our model is very efficient at test time, being linear in the number of basis components and requiring only very few components to accurately describe a wide variety of human hands. Indeed once the shape parameters β are inferred for a given user, the shape model could be 'baked in', thereby reducing to standard LBS plus subdivision.

As future work we would like to investigate efficient options for explicitly encoding the dependence of shape on pose. The current models are at a fairly coarse resolution, but it would be interesting to see if such a method could yield a super-resolved model. Another exciting avenue of future work is to 'personalize' a hand model interactively and in real time; we believe that given the learned shape basis, we can expect to accurately fit a personalized model from as few as one or two frames, a clear advantage over other approaches. Finally, we hope to apply our technique to other classes such as human body or animals.

References

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. https://code.google.com/p/ceres-solver/. 6

[2] I. Albrecht, J. Haber, and H.-P. Seidel. Construction and animation of anatomically based human hand models. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 98–109, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 2

[3] B. Allen, B. Curless, and Z. Popović. Articulated body deformation from range scan data. In *ACM Transactions on Graphics (TOG)*. ACM, 2002. 2

[4] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graphics*, 22(3), 2003. 2

[5] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape completion and animation of people. *ACM Trans. Graphics*, 24(3), 2005. 1, 2, 3

[6] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *Proc. ECCV*, 2012. 1, 2

[7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, 1999. 1, 2

[8] A. Buryanov and V. Kotiuk. Proportions of hand segments. *International Journal of Morphology*, 28(3):755–758, 2010. 2

[9] T. Cashman and A. Fitzgibbon. What shape are dolphins? building 3D morphable models from 2D images. *IEEE Trans. PAMI*, 2013. 2

[10] Y. Chen, Z. Liu, and Z. Zhang. Tensor-based human body modeling. In *Proc. CVPR*, June 2013. 2

[11] M. de La Gorce, N. Paragios, and D. J. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *Proc. CVPR*, 2008. 2

[12] O. Freifeld and M. J. Black. Lie bodies: A manifold representation of 3D human shape. In *Proc. ECCV*, 2012. 1

[13] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 2009. 1, 2

[14] T. Helten, A. Baak, G. Bharaj, M. Müller, H.-P. Seidel, and C. Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *3DV*, 2013. 2

[15] D. A. Hirshberg, M. Loper, E. Rachlin, and M. J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *Proc. ECCV*, 2012. 1

[16] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Proc. SGP*, 2008. 2

[17] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *Proc. ICCV*, 2009. 2

[18] C. Loop. Smooth Subdivision Surfaces Based on Triangles. Master's thesis, The University of Utah, Aug. 1987. 4

[19] T. Neumann, K. Varanasi, N. Hasler, M. Wacker, M. Magnor, and C. Theobalt. Capture and statistical modeling of arm-muscle deformations. In *Proc. Eurographics*, 2013. 2

[20] T. Rhee, U. Neumann, and J. P. Lewis. Human hand modeling from surface anatomy. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 27–34. ACM, 2006. 2

[21] S. M. Software. Poser. http://my.smithmicro.com/poser-3d-animation-software.html. 7

[22] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proc. SGP*, 2007. 5

[23] C. Stoll, Z. Karni, C. Rössl, H. Yamauchi, and H.-P. Seidel. Template deformation for point cloud fitting. In *Proc. Eurographics*, 2006. 2

[24] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *Proc. CVPR*, 2014. 1, 2, 4, 5, 7

[25] A. Tsoli, N. Mahmood, and M. J. Black. Breathing life into shape: capturing, modeling and animating 3D human breathing. *ACM Trans. Graphics*, 33(4), 2014. 2

[26] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graphics*, 2009. 2

[27] A. Weiss, D. Hirshberg, and M. J. Black. Home 3D body scans from noisy image and range data. In *Proc. ICCV*, 2011. 2