



Local Scale Adaptation to Hand Shape Model for Accurate and Robust Hand Tracking

P. Kalshetti¹  and P. Chaudhuri¹ 

¹Indian Institute of Technology Bombay, India



Figure 1: We introduce a new hand model, aMANO, by augmenting MANO's shape space with a local scale adaptation which enables calibrating to users with substantially different hand sizes than those covered by MANO. We also present a framework to calibrate and track aMANO by registering it to a sequence of depth frames. Here we see that aMANO registers more accurately than MANO for a child's hand sequence. We show the calibrated mesh in the rest pose for MANO and aMANO in the rightmost column.

Abstract

The accuracy of hand tracking algorithms depends on how closely the geometry of the mesh model resembles the user's hand shape. Most existing methods rely on a learned shape space model; however, this fails to generalize to unseen hand shapes with significant deviations from the training set. We introduce local scale adaptation to augment this data-driven shape model and thus enable modeling hands of substantially different sizes. We also present a framework to calibrate our proposed hand shape model by registering it to depth data and achieve accurate and robust tracking. We demonstrate the capability of our proposed adaptive shape model over the most widely used existing hand model by registering it to subjects from different demographics. We also validate the accuracy and robustness of our tracking framework on challenging public hand datasets where we improve over state-of-the-art methods. Our adaptive hand shape model and tracking framework offer a significant boost towards generalizing the accuracy of hand tracking.

CCS Concepts

• **Computing methodologies** → **Mesh models**; **Parametric curve and surface models**; **Motion capture**;

1. Introduction

Hand tracking plays a vital role in interacting with augmented and virtual reality systems. However, one of the significant challenges in the widespread adoption of these systems is their generalizability

across various demographics. We present a solution to this problem by presenting a hand shape model that can adapt to a wider variety of hand sizes than the most widely used hand shape model present in literature.

The importance of a user-specific (or calibrated) hand model for accurate tracking is well-established in the literature [TSR*14, TCT*16]. Most state-of-the-art hand tracking systems use a pre-calibrated user-specific hand model to track the pose from an input stream of images [TSR*14, TCT*16]. Recently, Tkach et al. [TTR*17] proposed an online calibration of a sphere-mesh model [TPT16] that is capable of handling a variety of hand shapes. However, it requires a large number of correlated parameters to model the shape, which is not readily amenable to optimization during registration [RTTP17].

Inspired by the recent literature in human body [LMR*15, OBB20, XBZ*20], face [EST*20] and hand [RTB17], our underlying hand model is a mesh-based model. Specifically, we use MANO [RTB17] which is the most widely used hand model [MDB*19, HVT*19, BBT19, ZHX*20, HTB*20, CPB*20, WMB*20, CPA*20]. However, this data-driven hand model cannot adapt to unseen hand shapes with significant deviations from the training set and thus adversely affects tracking.

We tackle this problem by introducing a new shape model *adaptive MANO (aMANO)* that augments MANO's shape space with local scale adaptation (see Sec. 3). This local scale adaptation enables calibrating the shape model to users with substantially different hand sizes than those covered by the original MANO shape space, as shown in Fig. 1. Specifically, we use a set of local scale parameters that scale each of the bones in the hand model and a modified skinning function to handle this local scale adaptation. aMANO gracefully adapts MANO to unseen hand sizes and thus aids for accurate tracking. We demonstrate the calibration ability of aMANO over MANO in Sec. 5.1.

Further, we present a framework to calibrate and track aMANO by registering it to a sequence of depth data in Sec. 4. Our registration method embeds a blend-shape model with the modified skinning function into an energy minimization formulation. We also reparameterize the pose, at each joint, in aMANO to achieve robust tracking; Sec. 5.3 highlights its effect. These ideas allow us to achieve competitive tracking accuracy compared to state-of-the-art methods, as reported in Sec. 5.2.

We highlight the primary contributions of this work below.

- We introduce a hand shape model, aMANO, that augments MANO's shape space with local scale adaptation and demonstrates its calibration ability over MANO on a captured dataset, including children's hands.
- We present a framework to calibrate and track aMANO by registering it to a sequence of depth data and achieve state-of-the-art tracking accuracy on many challenging datasets available in the literature.

2. Related Work

Over the past decade, especially due to the advent of deep learning, there has been a spur in hand-tracking research across academia and industry. The tracking methods either use a depth image or an RGB image as input. In this section, we focus on depth-based methods and refer the reader to Baek et al. [BKK19] for an overview of RGB-based hand tracking.

2.1. Hand tracking

One of the seminal works on depth-based hand tracking, proposed by Oikonomidis et al. [OKA11], used a primitive-based geometrical model for optimizing the pose using particle swarm optimization (PSO). Makris and Argyros [MA15] further extended it by incorporating an online shape adaptation of the hand model. However, it still uses a primitive model that lacks expressivity compared to our mesh-based model.

Taylor et al. [TSR*14] introduced user-specific hand modeling, and Sharp et al. [SKR*15] used it in their robust hand tracker. Khamis et al. [KTS*15] learned a shape space of hand from depth images of multiple users, which enabled Tan et al. [TCT*16] to extend their golden-energy tracking method to various users. Unfortunately, unlike ours, these implementations are not publicly available, restricting their usage for scientific research.

Tagliasacchi et al. [TST*15] revived the idea of using ICP-like algorithms for hand tracking by fitting a cylinder-based hand model using Levenberg-Marquardt (LM) non-linear least-squares optimization. We adapt the energy terms used in their method and use them in our tracking framework. The notable work of Taylor et al. [TBC*16] used a subdivided hand mesh to track using the LM algorithm. We borrow their idea for efficiently computing correspondences on a mesh. Further, Shen et al. [SCY*20] demonstrated that one could get away without subdividing the mesh using ideas from Phong shading. Our barycentric sampling of the mesh follows this approach and helps further increase the efficiency of our tracking framework.

Most tracking algorithms rely on a robust per-frame discriminative technique to avoid drifting. The early work by Qian et al. [QSW*14] introduced the idea of robust fingertip detection that exploits finger geometry. With the availability of large scale hand pose datasets (e.g. NYU [TSLP14], MSRA [SWL*15], Big-hand2.2M [YYs*17]) with 21 annotated keypoints (16 joints and 5 fingertips), state-of-the-art hand pose estimation networks [HRW*20, ZXCZ20, XZX*19, WPGY18] regress the 3D keypoints directly from depth. We use these per-frame keypoints (whenever available) to recover from tracking failure through the fingertip re-initializer energy term.

2.2. Shape modeling

Until recently, most methods used a fixed hand template model. Tkach et al. [TPT16] introduced the idea of sphere-meshes for generative hand modeling. Remelli et al. [RTTP17] further parameterized it for model personalization through local scaling. Tkach et al. [TTR*17] devised a method to optimize these parameters to personalize the hand model during online hand tracking. In contrast to these methods requiring a sphere-mesh hand model to work, our novel local scale adaptation can handle arbitrary skeleton-based meshes.

The recent advances in human body shape models (e.g. SMPL [LMR*15], STAR [OBB20], GHUM [XBZ*20]) have considerably pushed the boundaries in state-of-the-art human pose and shape tracking. Following this direction, we use the widely adopted MANO [RTB17] as our underlying hand model. However, it cannot adapt to unseen hand shapes with substantially large deviations

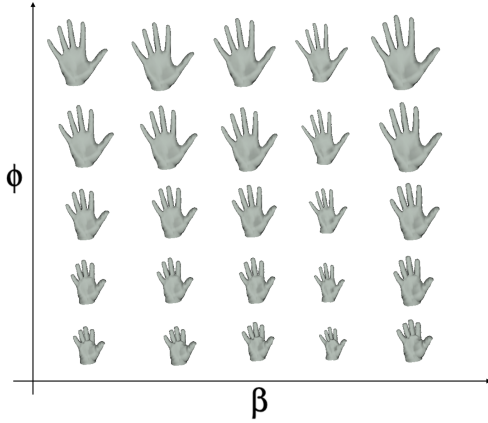


Figure 2: Our local scale parameters in aMANO compliment the shape space of MANO. Here we vary β and ϕ around the mean shape.

from the training set. Boukhayma et al. [BBT19] use a single global scale to handle the coarse size variation. Instead, we introduce local scale parameters for fine-grained control, which dramatically increases the span of our shape space.

3. Hand Model: aMANO

This section describes our proposed hand model, *adaptive MANO* (aMANO), that augments MANO's shape space with local scale adaptation.

MANO (hand Model with Articulated and Non-rigid defOrmations), introduced by Romero et al. [RTB17], is a hand model learned from 2018 scans of 31 subjects. It captures the user-specific vertex offsets via a shape blend-shape function and pose-specific vertex offsets to correct artifacts in skinning via a pose blend-shape function. A template vertex \bar{v}_i is offset as

$$v_i = \bar{v}_i + S_i \beta + P_i (r(\theta) - r(\bar{\theta})) \quad (1)$$

where $S_i \in \mathbb{R}^{3 \times 10}$ and $P_i \in \mathbb{R}^{3 \times 135}$ are the shape and pose blend shapes corresponding to the vertex v_i , $\beta \in \mathbb{R}^{10}$ is the shape parameter, $\theta \in \mathbb{R}^{15 \times 3}$ is the pose parameter capturing the axis angle rotation at each of the 15 joints, and $r(\theta) \in \mathbb{R}^{135}$ is the vectorized version of the stacked rotation matrices at each joint with pose θ ; $\bar{\theta}$ is the rest pose.

However, these PCA shape blend-shapes cannot capture hand shapes with significant deviations from training data. We augment the shape space of MANO by introducing local scale parameters ϕ that compliment the original shape parameters β and thus, increase the span of the shape space as shown in Fig. 2.

We now describe the local scale parameters in aMANO. Inspired by Jacobson and Sorkine [JS11], we assume a set of local scale parameters $\phi \in \mathbb{R}^{n_b}$ for each of the n_b bones in the hand (n_b is 20 in our hand model). We calculate the scale factor for each bone ϕ_j as

$$\phi_j = \frac{l_j^{(data)}}{l_j^{(template)}} \quad (2)$$

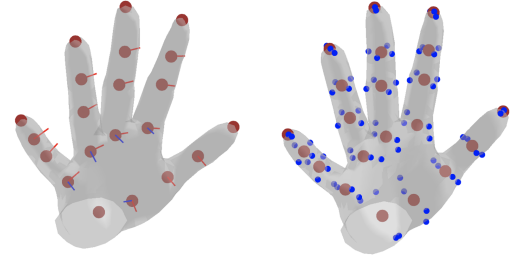


Figure 3: Left: We define 2 degrees of freedom at the MCP (metacarpophalangeal) joint, and 1 degree of freedom at each PIP (proximal interphalangeal) and DIP (distal interphalangeal) joints of a finger. The axis of rotation for each degree of freedom is plotted. Right: We select a ring of vertices (blue) around a keypoint (brown) to define K , which is then used to calculate the position of keypoints from a posed mesh.

where $l_j^{(data)} \in \mathbb{R}$ is the length of the j^{th} bone in the observed data, obtained from keypoints (each bone connects two keypoints), and $l_j^{(template)} \in \mathbb{R}$ is the length of the j^{th} bone in the template mesh model. The observed data keypoints are either available from the hand pose estimation network or marked manually on the depth image in the rest pose. We calculate the model keypoints $k \in \mathbb{R}^{21 \times 3}$ from the mesh vertices $v \in \mathbb{R}^{778 \times 3}$ using a sparse regression matrix $K \in \mathbb{R}^{21 \times 778}$ as

$$k = Kv \quad (3)$$

For each keypoint i , we select a ring of four vertices surrounding it and only fill the four corresponding columns in the i^{th} row of K . The vertices are selected such that the resulting keypoints lie at the anatomical joints and fingertips of the hand (see Fig. 3).

Existing methods use the standard linear blend skinning (LBS) to pose the mesh by applying a weighted combination of transformations for each bone. Let $a_j \in \mathbb{R}^3$ and $b_j \in \mathbb{R}^3$ be the start and end positions of j^{th} bone in rest pose mesh (after applying MANO shape-blends) respectively, and $R_j \in \mathbb{R}^{3 \times 3}$ be the rotation matrix that takes bone j 's rest vector $(b_j - a_j)$ to its pose vector $(b'_j - a'_j)$. Using LBS, the deformed vertex v'_i is given by

$$v'_i = \sum_{j=1}^{n_b} W_{b_{ij}} \{a'_j + R_j (-a_j + v_i)\} \quad (4)$$

where $W_b \in \mathbb{R}^{n_v \times n_b}$ is the bone weight matrix.

To incorporate our local scale parameters into the standard LBS, we can anisotropically scale the bone in the reference frame using the local scale parameter, ϕ_j as

$$v'_i = \sum_{j=1}^{n_b} W_{b_{ij}} \{a'_j + R_j (\phi_j (-a_j + v_i))\} \quad (5)$$

However, since ϕ_j is constant over the mesh, all points on each bone will stretch uniformly. Thus, if v_i lies beyond an endpoint of a bone, it will get overly stretched, as shown in Fig. 4. We can avoid

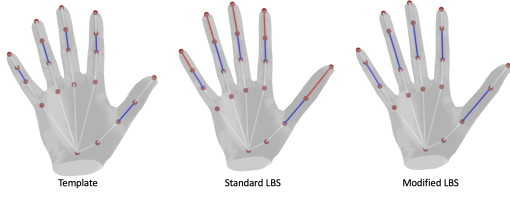


Figure 4: Consider we want to increase the scale of the bone (marked in blue) connecting the PIP joint and the DIP joint in the template mesh (left). Using the standard LBS (middle), we notice unwanted scaling of the bone (marked in red in middle image) connecting the DIP joint to the TIP (fingertip). However, the modified LBS [JS11] (right) handles this by limiting the influence of each bone using endpoint weights.



Figure 5: Endpoint weights W_e for the bones in index finger.

this problem if we know the position of v_i relative to bone j 's endpoints. We can then restrict the effect of scaling to part of the mesh associated with the scaled bone using an additional set of endpoint weights $W_e \in \mathbb{R}^{n_v \times n_b}$ for each bone. We compute W_e by minimizing the shape-aware smoothness functional (Laplacian energy) using the following constraints for the j^{th} bone weight $W_{e_{ij}}$, for all vertices i , with start position a_j and end position b_j , as follows:

- the weight is one at a_j and along all other bones connected to a_j ,
- the weight is zero at b_j and along all bones connected to b_j and,
- the weight varies linearly along a_j to b_j .

We optimize the discretized version of the functional, following Jacobson et al. [JBPS11]. We plot the endpoint weights for the bones along the index finger in Fig. 5.

We insert the local scale parameters and the endpoint weights in the standard LBS resulting in the modified LBS given by

$$v'_i = \sum_{j=1}^{n_b} W_{b_{ij}} \{a'_j + R_j (W_{e_{ij}} s_j + (-a_j + v_i))\} \quad (6)$$

where $s_j = (\phi_j - 1)(b_j - a_j)$.

4. Hand Tracking

We now present our framework to calibrate and track aMANO by registering it to a sequence of depth frames $\{D^{(f)}\}_{f=1}^{n_f} \subset \mathbb{R}^{H \times W}$ acquired from a sensor.

Our framework consists of two stages: calibration, in which we estimate $\{\beta^{(f)}, \theta^{(f)}\}$ for each frame, and tracking, in which we only estimate θ_f for each frame. We perform the calibration stage for the initial few frames of the sequence. Once the shape parameter

converges to $\hat{\beta}$, we transition to the tracking stage using the fixed shape parameter $\hat{\beta}$ for the subsequent frames. Before beginning the calibration stage, we calculate ϕ using the observed data keypoints and use it for all the frames. We follow this process for each new user.

The calibration and tracking stages minimize a registration energy to estimate the model parameters $\{\beta^{(f)}, \theta^{(f)}\}$ for each frame f . In the subsequent discussion, we drop the superscript f denoting the frame index for brevity. The registration energy is written as a weighted sum of several terms

$$E(\theta) = \sum_{\tau \in \mathcal{T}} \omega_\tau E_\tau(\theta) \quad (7)$$

where $\omega_\tau \in \mathbb{R}$ are the weights associated with each term. We use the same terms in both stages except the shape prior term, which is only used during the calibration stage. We now explain each energy term in \mathcal{T} .

4.1. 3D data term

The 3D data term ensures that the model explains the observed point cloud $x \in \mathbb{R}^{n_{3D} \times 3}$. We obtain the point cloud from the depth image and apply *furthest point downsampling* [QSMG17] for efficient downstream processing. Additionally, we also calculate the normal $x_i^\perp \in \mathbb{R}^3$ to the i^{th} point by locally fitting a plane [Rus09]. We use the point-to-plane ICP [CM91] to define the 3D data term

$$E_{data_{3D}}(\theta) = \sum_{i=1}^{n_{3D}} (x_i - y_i(\theta))^T y_i^\perp \quad (8)$$

where $y_i \in \mathbb{R}^3$ is the closest point on the mesh corresponding to x_i , and y_i^\perp is the normal at y_i . Computing the closest point on the mesh for every data point is expensive since we have to project each point on every face and find the closest point on that triangle. To compute y_i efficiently on a mesh, we evaluate a random subset of predefined n_s barycenters on the mesh and associate the one that minimizes the distance to the observed point x_i (see Figure 6). We obtain the closest point on the mesh, y_i as

$$y_i(\theta) = \arg \min_{b_j} (\|x_i - b_j(\theta)\|^2 + \omega^\perp \|x_i^\perp - b_j^\perp(\theta)\|^2) \quad (9)$$

where $b_j \in \mathbb{R}^3$ and $b_j^\perp \in \mathbb{R}^3$ are the position and normal of the j^{th} evaluated barycenter respectively, where $j \in \{1, 2, \dots, n_s\}$. The normal term prohibits selecting points that might be closer but face away from the camera, as shown by Taylor et al. [TBC*16].

4.2. 2D data term

We utilize additional information from the background region of the depth image, which suggests that the model should not project on it. We capture this information via the model's distance to data in the 2D image space. For efficiency, we evaluate another set of predefined barycenters to obtain the 2D image space positions $p \in \mathbb{R}^{n_{2D} \times 3}$ that are used to compute the 2D alignment energy as

$$E_{data_{2D}}(\theta) = \sum_{i=1}^{n_{2D}} \|q_i - p_i(\theta)\|^2 \quad (10)$$

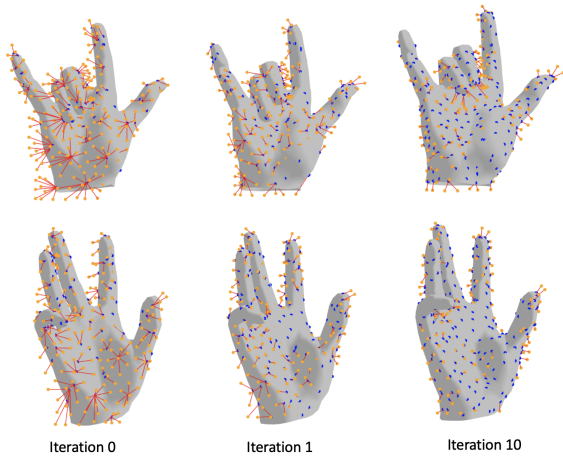


Figure 6: The two rows show two different hand pose fits to down-sampled, observed 3D point cloud (orange). We plot the closest point correspondences in 3D with short red lines that indicate the distance to the corresponding point (blue) on the mesh. As the iterations progress, the fit and the correspondences improve.

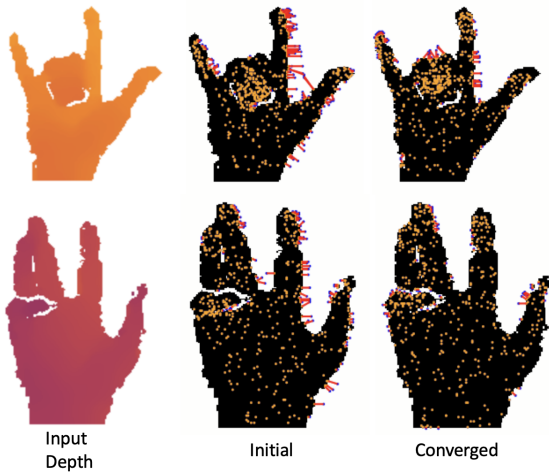


Figure 7: Each row shows a different example of updating correspondences in 2D. The leftmost image in each row is the input depth image. The hand in the top row is further away from the camera than the one in the bottom row, thus the difference in colors: orange and dark red. The correspondences between image space positions (in blue) and the closest points on the silhouette (in orange) improve from their initial estimate (middle image in each row) as the iterations progress and the fit converges (rightmost image in each row)

where $q_i \in \mathbb{R}^2$ is the 2D image space position of the closest point on the silhouette, computed using the distance transform [MQR03] of the cropped depth image; we only consider points that lie outside the silhouette (see Fig. 7).

4.3. Pose prior terms

Minimizing only the data terms leads to unrealistic poses due to noisy and partial data from the monocular depth sensor. To avoid overfitting to this insufficient data, we introduce two prior terms on the pose.

To naturally restrict physically implausible rotations, we explicitly re-parameterize the over-parameterized MANO pose parameters (3-DoF axis angles for each joint) by defining an axis of rotation for each degree of freedom (DoF) in the rest pose as shown in Fig. 3. This intuitive pose representation allows us to impose bounds $[\underline{\theta}_i, \bar{\theta}_i]$ on each articulation angle using an angle bound energy term

$$E_{bound}(\theta) = \sum_i \max(0, \theta_i - \underline{\theta}_i)^2 + \max(0, \bar{\theta}_i - \theta_i)^2 \quad (11)$$

We also model the correlation among the articulation angles using a data-driven prior. Specifically, we construct a PCA pose space from the recorded hand poses from Kinect v1 by Schröder et al. [SMRB14] and enforce the articulation angles to lie close to this low-dimensional linear subspace. Instead of introducing the PCA coefficients in the optimization, we use the projective PCA from Tagliasacchi et al. [TST*15] which allows rewriting the energy only in terms of the original pose parameters as

$$E_{pca}(\theta) = \|(\theta - \mu) - \Pi \Sigma^2 \Pi^T (\theta - \mu)\|^2 \quad (12)$$

where $\Pi \in \mathbb{R}^{|\theta| \times |\theta|}$ is the PCA basis matrix, $\mu \in \mathbb{R}^{|\theta|}$ is the mean pose, and $\Sigma \in \mathbb{R}^{|\theta| \times |\theta|}$ is the diagonal matrix containing the standard deviation of the data along the PCA basis. This term increases the robustness of our method against occlusion.

4.4. Intersection term

We add an intersection term to penalize inter-penetration among fingers. (The intersection between fingers and palm is handled by the angle bound term.) We define a set of proxy spheres [TBC*16] to approximate the fingers in the hand mesh as shown in Fig. 8. Each sphere has a heuristically defined constant radius $r \in \mathbb{R}$ and a center $c \in \mathbb{R}^3$ defined by a convex combination of the neighboring vertices. We adapt the cylinder-based intersection term from Tagliasacchi et al. [TST*15] to our sphere-based intersection energy. Specifically, the intersection energy minimizes the distance between the deepest penetration points $x_i \in \mathbb{R}^3$ and $x_j \in \mathbb{R}^3$ between two intersecting spheres i and j

$$E_{int}(\theta) = \sum_{(i,j) \in S} (x_i(\theta) - x_j(\theta))^T x_i^\perp(\theta) \quad (13)$$

where S is the set of sphere pairs excluding those belonging to the same finger and $x_i^\perp(\theta) = \frac{c_j(\theta) - c_i(\theta)}{\|c_j(\theta) - c_i(\theta)\|}$ and $x_i(\theta) = c_i(\theta) + r_i x_i^\perp(\theta)$.

4.5. Temporal smoothness

So far, all the terms depend only on a single frame, which leads to jittery tracking. Therefore, we enforce a velocity constraint on the model keypoints. We do not enforce this constraint directly on the pose angles because a small perturbation on angles closer to the root of the kinematic chain has a larger effect than those that are

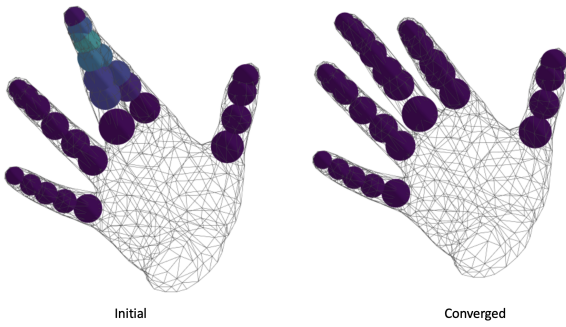


Figure 8: Intersection between proxy spheres helps detect finger intersection and is used to compute the intersection term. As the iterations converge, the intersecting fingers are pushed apart.

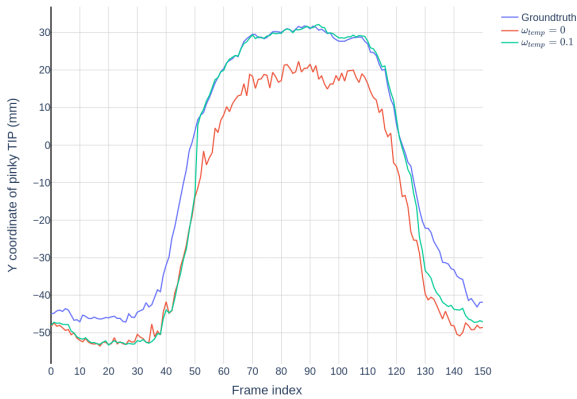


Figure 9: We plot the Y coordinate of the tip of the pinky finger for 150 frames of the BigHand dataset for user 1. When the temporal smoothness term is present (green), the plot follows the smooth trajectory of the ground-truth (blue) more accurately than when it is absent (red).

further away [TST*15]. Our temporal smoothness term for frame f is given by

$$E_{temp}(\theta) = \sum_{i=1}^{n_k} \|k_i^{(f)}(\theta) - k_i^{(f-1)}\|^2 \quad (14)$$

We show the effect of the temporal smoothing term in Fig. 9 where we plot the Y coordinate of the tip of the pinky finger for 150 frames of the BigHand dataset for user 1. The plot is noticeably smoother with the temporal smoothness term present.

4.6. Fingertip re-initializer

To further improve the robustness of our tracker, we incorporate a fingertip energy term, similar to Taylor et al. [TBC*16]. This term enforces the model's fingertips $t \in \mathbb{R}^{5 \times 3}$ to be close to the detected fingertips \hat{t} obtained from a discriminative hand pose es-

imation network [HRW*20, WPGY18] trained on annotated pairs of cropped depth image of a hand and corresponding keypoints.

$$E_{init}(\theta) = \sum_{i=1}^5 \|t_i(\theta) - \hat{t}_i\|^2 \quad (15)$$

This term allows us to recover from tracking failure caused due to fast motion or tracking error accumulated over multiple frames, which can lead to incorrect initialization for the optimization procedure if we only depend on the pose optimized in the previous frame.

4.7. Shape prior

To avoid drifting away from a human hand shape, we regularize the shape parameter β by enforcing it to lie close to the mean of the PCA shape space in MANO.

$$E_{shape}(\beta) = \|\beta\|^2 \quad (16)$$

4.8. Optimization

All the energy terms are written as a sum of squared residuals, which leads to a nonlinear least-squares optimization problem. We linearize each term and solve using the Levenberg-Marquardt algorithm [Lev44, Mar63]. Further, we use a discrete optimization over the 3D correspondences at each iteration: we sample a new set of barycenters and update the correspondences if any of the new correspondences are closer than the previous ones. This aids in faster convergence and allows a surprisingly small number of barycenters (n_s) to be used at each iteration for Equation 9.

We observe that naively sampling random barycenters on the hand mesh leads to more points on the palm region and fewer on the fingers. Since fingers play a crucial role in estimating the pose, we partition the mesh into parts and use a part-based sampling strategy that ensures samples are selected from each part, as shown in Fig. 10.

To initialize the pose for the current frame, we use the previous frame's optimized pose; for the first frame, we register the model to the keypoints obtained either from the dataset or marked manually.

5. Evaluation

Datasets In this section, we evaluate our proposed hand model, aMANO, and tracking framework on a variety of publicly available hand pose datasets viz., BigHand [YYs*17] (10 users, each with 6 viewpoints; total of around 2 million frames; Intel RealSense SR300), HANDS2019 [AGHB*20] (around 175000 frames; Intel RealSense SR300), GuessWho [TTR*17] (12 users; total of around 80000 frames; Intel RealSense SR300), NYU [TSLP14] (2 users; total of around 8000 frames; PrimeSense Depth Camera) and MSRA [SWL*15] (9 users, each with 17 gestures; total of around 75000 frames; Intel's Creative Interactive Gesture Camera). Further, we use a Kinect v2 sensor to capture multiple depth sequences of users from different demographics (4 children between the age of 7-9 years, 4 adult females, 2 adult males) performing six gestures (flexion/extension, adduction/abduction, open/close fist, global transforms, American sign language, random) with a total

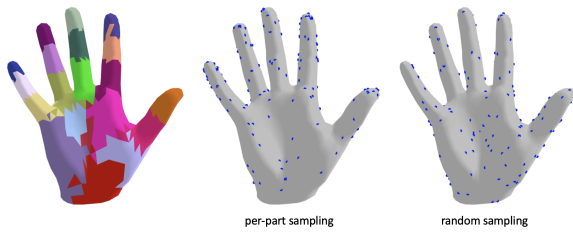


Figure 10: Right: Naively sampling random barycenters on the hand mesh leads to more points on the palm (large area, more faces) region and fewer on the fingers (small area, fewer faces). Left: We statically divide the hand mesh into parts. Middle: Our part-based sampling strategy ensures samples are selected from each part, resulting in more samples on the fingers which are important for pose registration.

of around 20000 frames. We show samples from our dataset along with the tracking result in Fig. 11.

Metrics We quantitatively evaluate the registration or fitting accuracy using two dense metrics: data-to-model error E_{3D} and model-to-data error E_{2D} , and a sparse metric: keypoint error E_k . Given an observed depth image $D^{(data)}$, a rendered depth image of the model $D^{(model)}$, E_{3D} is defined as

$$E_{3D} = \frac{1}{|D^{(data)}|} \sum_{p \in D^{(data)}} \|p - \Pi_{D^{(model)}}(p)\| \quad (17)$$

where $\Pi_{D^{(model)}}$ denotes the closest point correspondence of the observed data point p to the rendered model point cloud, and E_{2D} is defined as

$$E_{2D} = \frac{1}{|D^{(model)} \setminus \partial D^{(data)}|} \sum_{x \in D^{(model)} \setminus \partial D^{(data)}} \|x - \Pi_{\partial D^{(data)}}(x)\| \quad (18)$$

where $\partial D^{(data)}$ is the hand region's image space silhouette, and $D^{(model)} \setminus \partial D^{(data)}$ denotes the set of rendered model points that lie outside the observed data silhouette. More information about the metrics can be found in Tkach et al. [TPT16]. Given a set of observed keypoints $k^{(data)}$ and $k^{(model)}$, the keypoint error is given by

$$E_k = \frac{1}{|k|} \sum_{i \in \{1 \dots |k|\}} \|k_i^{(data)} - k_i^{(model)}\| \quad (19)$$

where $k_i \in \mathbb{R}^3$ is the i^{th} keypoint. We report keypoint error wherever $k^{(data)}$ is available in the annotated dataset.

Implementation We run all our experiments on a desktop machine with an Intel i7-7700, 3.60GHz processor, and 32GB of RAM. Our tracking framework runs entirely on the CPU and is implemented purely in Python. Our calibration and tracking stages run consistently at around 10 frames per second. We observe that 10 LM iterations per frame are sufficient for convergence in most cases.

Optimization Hyperparameters The hyperparameters used in the optimization play a significant role in the behaviour of the

Method	E_{3D} (in mm)
Remelli et al. [RTTP17] (offline)	2.5
Tkach et al. [TTR*17] (offline)	2.3
Our framework with aMANO (online)	3.1

Table 1: Tracking accuracy on the GuessWho dataset.

method. The weights used for the energy terms are $\omega_{3d} = 1$, $\omega_{2d} = 0.01$, $\omega_{bound} = 100$, $\omega_{pca} = 0.1$, $\omega_{int} = 100$, $\omega_{temp} = 0.1$, $\omega_{init} = 0.1$. The number of points used in the 3D and 2D data terms are $n_{3D} = 200$ and $n_{2D} = 400$ respectively. We list these parameters here to aid reproducibility of our results.

5.1. Calibration: aMANO v/s MANO

We demonstrate the capability of our proposed hand model, aMANO, over MANO by calibrating both models to our captured dataset. For both MANO and aMANO, we use our energy optimization-based calibration method described earlier. The results are shown in Fig. 12. We can see that aMANO successfully adapts and fits adult and children's hands, whereas MANO cannot adapt to hand sizes that are significantly far from its training data. For example, MANO cannot adapt to children's hand sizes, whereas aMANO gracefully adapts to unseen hand sizes.

5.2. Tracking accuracy

To validate the accuracy of our tracking framework, we register aMANO on the GuessWho, NYU, and MSRA datasets. We provide qualitative results on all these datasets in Fig. 13

In Table 1, we quantitatively compare our online calibration and tracking method on the GuessWho dataset with two state-of-the-art offline methods [RTTP17, TTR*17] that use a sphere-mesh model with a large number of shape parameters. These methods are offline because they fit the mesh to the input depth by simultaneously optimizing over multiple frames. In contrast, we perform much faster per frame online iterations. Our results are competitive with the offline methods, with much less computational effort.

We report our tracking accuracy on the NYU and MSRA datasets in Table 2 and Table 3 respectively. Here we compare our tracking method with state-of-the-art hand pose estimation methods for each dataset. For a fair comparison with just our tracking framework, first, we calibrate aMANO to each dataset. Then we register this mesh only to the estimated keypoints from these methods and measure the E_{3D} and E_{2D} metrics. For our framework, we register the same mesh to the depth point cloud and the fingertips obtained from these methods. Our tracking framework provides a more accurate fit, and thus it can be used as a refinement module over hand pose estimation methods.

We also compare the accuracy of tracking using our framework with MANO and aMANO on different demographics from our captured dataset in Table 4. For both models, MANO and aMANO, we use our shape calibration and tracking stages, to register the mesh to every input depth frame. The results conclusively show that aMANO captures hand shapes for all demographics better than MANO.

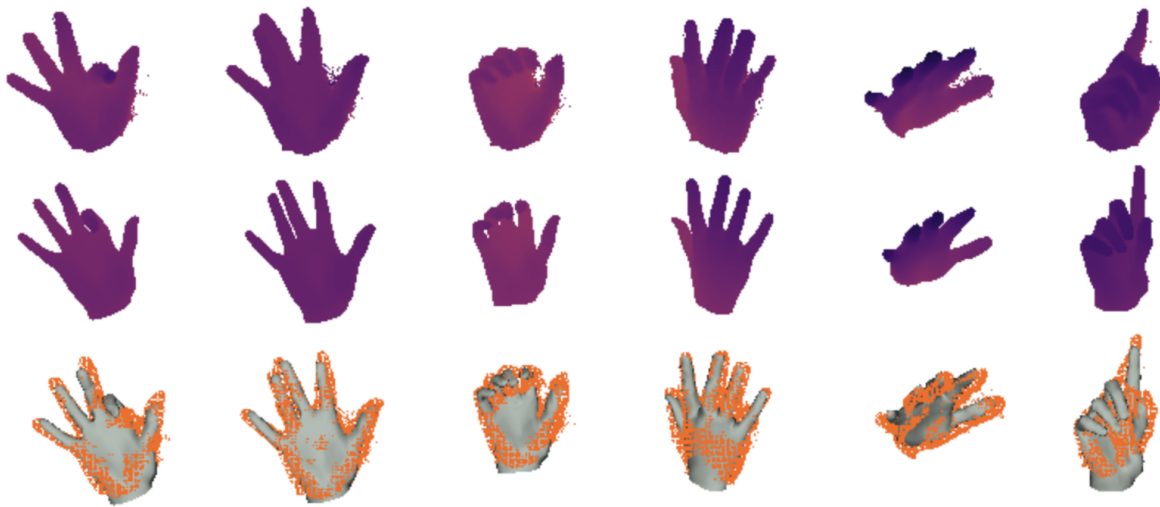


Figure 11: Samples from our captured dataset. The top row shows the captured input depth, the middle row shows the rendered depth of registered aMANO and, the bottom row shows the mesh model in grey, fit to the input point cloud in orange.

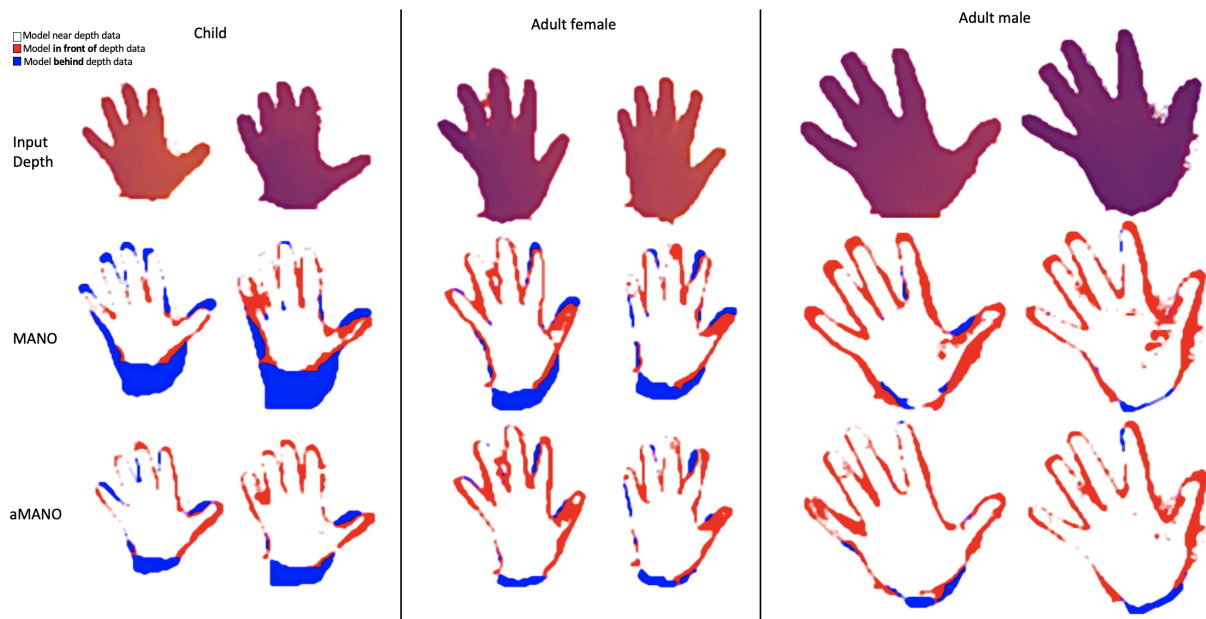


Figure 12: Comparison between MANO and aMANO by calibrating both models on our captured dataset with hands belonging to different demographics. Each of the six columns represents a different user. The fitting error plots in the second and third rows for each example show the quality of the fit. The regions where the two models nearly overlap in depth are colored white, the regions where the model is in front of depth data are colored red, and where the model is behind the depth data are colored blue.

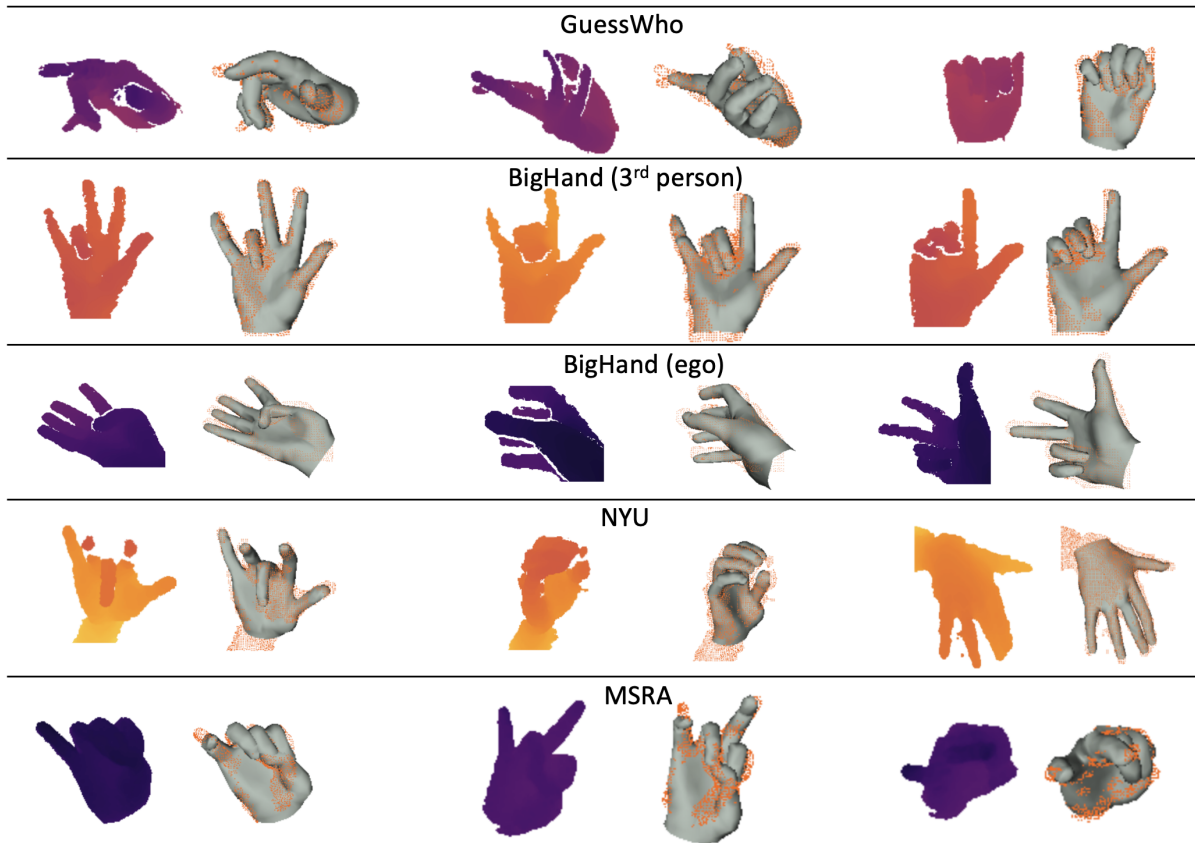


Figure 13: Each row shows qualitative results for tracking performed using our method on various dataset. The top row shows results from the GuessWho dataset. The 2nd and 3rd rows show results from the BigHand dataset (the 3rd row captures are from an ego view), 4th row is from the NYU dataset, and the final row is from the MSRA dataset. The odd columns are the input depth images from the dataset, the even columns are the corresponding mesh model fit to the point clouds (in orange) derived from the input depth images.

Method	E_{3D} (in mm)	E_{2D} (in pixels)
Huang et al. [HRW*20]	10.1	0.242
Our tracking framework	6.7	0.227

Table 2: Tracking accuracy on the NYU dataset. Both methods track using meshes calibrated with aMANO and our calibration method.

Method	E_{3D} (in mm)	E_{2D} (in pixels)
Wan et al. [WPGY18]	6.8	0.819
Our tracking framework	5.2	0.475

Table 3: Tracking accuracy on the MSRA dataset. Both methods track using meshes calibrated with aMANO and our calibration method.

5.3. Effect of pose parameterization

We now show the advantage of using our pose parametrization and constructed PCA pose prior instead of using MANO's pose param-

Demographics	E_{3D} (in mm)		E_{2D} (in pixels)	
	MANO	aMANO	MANO	aMANO
Children	5.4	4.5	1.491	0.758
Adult (female)	5.8	5.5	0.765	0.523
Adult (male)	5.9	5.4	0.471	0.363

Table 4: Tracking accuracy with MANO and aMANO on our captured datasets that contains data across various demographics.

eterization by evaluating the keypoint error on the Hands2019 challenge dataset [AGHB*20].

The dataset consists of MANO shape and pose parameters that are estimated using a gradient-based optimization [BKK19]. Table 5 shows the superior performance of our tracking framework with significantly lower keypoint error than those provided by the challenge organizers which uses a gradient-based optimization [BKK19]. We also plot the fraction of frames within a threshold for each of the metrics in Fig. 14.

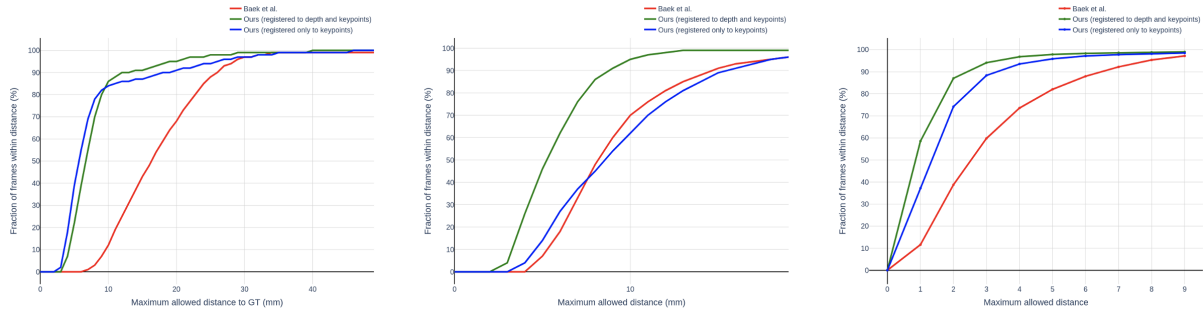


Figure 14: The fraction of frames within distance threshold is plotted against the maximum allowed distance for keypoint error (E_k), data-to-model error (E_{3D}) and model-to-data error (E_{2D}) metrics for the Hands2019 challenge dataset. Our method (green curve, registered per frame to both depth and keypoints) works better than the method used by the challenge organizers, Baek et al. [BKK19] (red curve). We also register only to the keypoints (blue curve) and observe that our pose parameterization is significantly better than the PCA pose space of MANO.

Method	E_k (in mm)
Baek et al. [BKK19]	16.4
Our framework (with aMANO)	5.9

Table 5: Keypoint error (E_k) comparison on the Hands2019 challenge dataset between our framework and Baek et al. [BKK19].

Left out energy term	E_{3D} (in mm)	E_{2D} (in pixels)	E_k (in mm)
3D data term	22.4	0.787	12.8
2D data term	4.1	0.518	12.1
Pose prior	7.8	0.420	13.6
Intersection penalty	5.7	0.301	12.2
Temporal smoothness	4.5	0.298	12.0
Fingertip reinitializer	3.3	0.274	21.490
None	4.0	0.295	11.9

Table 6: Ablation study of energy terms present in the optimization objective. This study was performed on 10000 frames of user 1 in viewpoint "1 75" of the BigHand dataset.

5.4. Effect of energy terms

We show the importance of the terms in our energy function as described in Section 4 by an ablation study on a sequence of the BigHand dataset.

In Table 6 we observe that the surface fitting (E_{3D}) and silhouette fitting (E_{2D}) metrics are minimum when all the energy terms are included in the optimization. In each row of the table, we remove one energy term from the optimization objective and report its error on the various error metrics.

6. Conclusion

We present an intuitive and mathematically robust extension to existing hand shape models to accommodate users with different hand sizes. We demonstrate that our hand model is capable of representing hand sizes of very different demographics, including that of

children, which was not possible until now. We also present a state-of-the-art framework that can register our model to depth data with competitive tracking accuracy. We plan to release our code and thus making it the only publicly available model-fitting hand tracking solution that uses triangle meshes that can be easily integrated with existing pipelines.

Our current local scale parameters can only adapt along the direction of bones; however, we can further increase the shape space by introducing the ability to vary the thickness of fingers and the palm. In this paper, we work only with depth data. However, the aMANO model and our tracking framework should be able to generalize to RGB data as well with appropriately designed energy terms.

References

- [AGHB*20] ARMAGAN A., GARCIA-HERNANDO G., BAEK S., HAMPALI S., RAD M., ZHANG Z., XIE S., CHEN M., ZHANG B., XIONG F., ET AL.: Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3d hand pose estimation under hand-object interaction. In *ECCV* (2020). 6, 9
- [BBT19] BOUKHAYMA A., BEM R. D., TORR P. H.: 3d hand shape and pose from images in the wild. In *CVPR* (2019), pp. 10843–10852. 2, 3
- [BKK19] BAEK S., KIM K. I., KIM T.-K.: Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *CVPR* (2019). 2, 9, 10
- [CM91] CHEN Y., MEDIONI G.: Object modeling by registration of multiple range images. In *ICRA* (1991). 4
- [CPA*20] CORONA E., PUMAROLA A., ALENYA G., MORENO-NOGUER F., ROGEZ G.: Ganhand: Predicting human grasp affordances in multi-object scenes. In *CVPR* (2020). 2
- [CPB*20] CHOUTAS V., PAVLAKOS G., BOLKART T., TZIONAS D., BLACK M. J.: Monocular expressive body regression through body-driven attention. In *ECCV* (2020), pp. 20–40. 2
- [EST*20] EGGER B., SMITH W. A. P., TEWARI A., WUHRER S., ZOLLHOEFER M., BEELER T., BERNARD F., BOLKART T., KORTYLEWSKI A., ROMDHANI S., THEOBALT C., BLANZ V., VETTER T.: 3d morphable face models—past, present, and future. *ACM TOG* 39, 5 (2020). 2

- [HRW*20] HUANG W., REN P., WANG J., QI Q., SUN H.: Awr: Adaptive weighting regression for 3d hand pose estimation. In *AAAI* (2020). 2, 6, 9
- [HTB*20] HASSON Y., TEKIN B., BOGO F., LAPTEV I., POLLEFEYS M., SCHMID C.: Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR* (2020). 2
- [HVT*19] HASSON Y., VAROL G., TZIONAS D., KALEVATYKH I., BLACK M. J., LAPTEV I., SCHMID C.: Learning joint reconstruction of hands and manipulated objects. In *CVPR* (2019). 2
- [JBPS11] JACOBSON A., BARAN I., POPOVIĆ J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM TOG* 30, 4 (2011), 78:1–78:8. 4
- [JS11] JACOBSON A., SORKINE O.: Stretchable and twistable bones for skeletal shape deformation. *ACM TOG* 30, 6 (2011), 165:1–165:8. 3, 4
- [KTS*15] KHAMIS S., TAYLOR J., SHOTTON J., KESKIN C., IZADI S., FITZGIBBON A.: Learning an efficient model of hand shape variation from depth images. In *CVPR* (2015). 2
- [Lev44] LEVENBERG K.: A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.* 2, 2 (1944), 164–168. 6
- [LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM TOG* 34, 6 (2015), 248:1–248:16. 2
- [MA15] MAKRISS A., ARGYROS A.: Model-based 3d hand tracking with on-line shape adaptation. In *BMVC* (2015), pp. 77:1–77:12. 2
- [Mar63] MARQUARDT D. W.: An algorithm for least-squares estimation of nonlinear parameters. *Journal of SIAM* 11, 2 (1963), 431–441. 6
- [MDB*19] MUELLER F., DAVIS M., BERNARD F., SOTNYCHENKO O., VERSCHOOR M., OTADUY M. A., CASAS D., THEOBALT C.: Real-time Pose and Shape Reconstruction of Two Interacting Hands With a Single Depth Camera. *ACM TOG* 38, 4 (2019). 2
- [MQR03] MAURER C. R., QI R., RAGHAVAN V.: A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE TPAMI* 25, 2 (2003), 265–270. 5
- [OBB20] OSMAN A. A. A., BOLKART T., BLACK M. J.: STAR: A sparse trained articulated human body regressor. In *ECCV* (2020), pp. 598–613. 2
- [OKA11] OIKONOMIDIS I., KYRIAZIS N., ARGYROS A. A.: Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC* (2011), vol. 1:2, p. 3. 2
- [QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR* (2017), pp. 652–660. 4
- [QSW*14] QIAN C., SUN X., WEI Y., TANG X., SUN J.: Realtime and robust hand tracking from depth. In *CVPR* (2014), pp. 1106–1113. 2
- [RTB17] ROMERO J., TZIONAS D., BLACK M. J.: Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG* 36, 6 (2017), 245:1–245:17. 2, 3
- [RTTP17] REMELLI E., TKACH A., TAGLIASACCHI A., PAULY M.: Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In *ICCV* (2017), pp. 2535–2543. 2, 7
- [Rus09] RUSU R. B.: *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009. 4
- [SCY*20] SHEN J., CASHMAN T. J., YE Q., HUTTON T., SHARP T., BOGO F., FITZGIBBON A., SHOTTON J.: The phong surface: Efficient 3d model fitting using lifted optimization. In *ECCV* (2020), pp. 687–703. 2
- [SKR*15] SHARP T., KESKIN C., ROBERTSON D., TAYLOR J., SHOTTON J., KIM D., RHEMANN C., LEICHTER I., VINNIKOV A., WEI Y., FREDMAN D., KRUPKA E., FITZGIBBON A., IZADI S., KOHLI P.: Accurate, robust, and flexible real-time hand tracking. In *CHI* (2015), pp. 3633–3642. 2
- [SMRB14] SCHRÖDER M., MAYCOCK J., RITTER H., BOTSCH M.: Real-time hand tracking using synergistic inverse kinematics. In *ICRA* (2014), pp. 5447–5454. 5
- [SWL*15] SUN X., WEI Y., LIANG S., TANG X., SUN J.: Cascaded hand pose regression. In *CVPR* (2015). 2, 6
- [TBC*16] TAYLOR J., BORDEAUX L., CASHMAN T., CORISH B., KESKIN C., SOTO E., SWEENEY D., VALENTIN J., LUFF B., TOPALIAN A., WOOD E., KHAMIS S., KOHLI P., SHARP T., IZADI S., BANKS R., FITZGIBBON A., SHOTTON J.: Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM TOG* 35 (2016). 2, 4, 5, 6
- [TCT*16] TAN D. J., CASHMAN T., TAYLOR J., FITZGIBBON A., TARLOW D., KHAMIS S., IZADI S., SHOTTON J.: Fits like a glove: Rapid and reliable hand shape personalization. In *CVPR* (2016). 2
- [TPT16] TKACH A., PAULY M., TAGLIASACCHI A.: Sphere-meshes for real-time hand modeling and tracking. *ACM TOG* 35, 6 (2016), 1–11. 2, 7
- [TSLP14] TOMPSON J., STEIN M., LECUN Y., PERLIN K.: Real-time continuous pose recovery of human hands using convolutional networks. *ACM TOG* 33 (2014). 2, 6
- [TSR*14] TAYLOR J., STEBBING R., RAMAKRISHNA V., KESKIN C., SHOTTON J., IZADI S., FITZGIBBON A.: User-specific hand modeling from monocular depth sequences. In *CVPR* (2014). 2
- [TST*15] TAGLIASACCHI A., SCHRÖDER M., TKACH A., BOUAZIZ S., BOTSCH M., PAULY M.: Robust articulated-icp for real-time hand tracking. In *Comput. Graph. Forum* (2015), vol. 34:5, pp. 101–114. 2, 5, 6
- [TTR*17] TKACH A., TAGLIASACCHI A., REMELLI E., PAULY M., FITZGIBBON A.: Online generative model personalization for hand tracking. *ACM TOG* 36, 6 (2017), 1–11. 2, 6, 7
- [WMB*20] WANG J., MUELLER F., BERNARD F., SORLI S., SOTNYCHENKO O., QIAN N., OTADUY M. A., CASAS D., THEOBALT C.: RGB2Hands: Real-Time Tracking of 3D Hand Interactions from Monocular RGB Video. *ACM TOG* 39, 6 (2020). 2
- [WPGY18] WAN C., PROBST T., GOOL L., YAO A.: Dense 3d regression for hand pose estimation. In *CVPR* (2018). 2, 6, 9
- [XBZ*20] XU H., BAZAVAN E. G., ZANFIR A., FREEMAN W. T., SUKTHANKAR R., SMINCHISESCU C.: Ghum & ghuml: Generative 3d human shape and articulated pose models. In *CVPR* (2020), pp. 6184–6193. 2
- [XZX*19] XIONG F., ZHANG B., XIAO Y., CAO Z., YU T., ZHOU TIANYI J., YUAN J.: A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *ICCV* (2019). 2
- [YYS*17] YUAN S., YE Q., STENGER B., JAIN S., KIM T.-K.: Big-hand2.2m benchmark: Hand pose dataset and state of the art analysis. In *CVPR* (2017). 2, 6
- [ZHX*20] ZHOU Y., HABERMANN M., XU W., HABIBIE I., THEOBALT C., XU F.: Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR* (2020). 2
- [ZXCZ20] ZHANG Z., XIE S., CHEN M., ZHU H.: Handaugment: A simple data augmentation method for depth-based 3d hand pose estimation. *arXiv* (2020), arXiv–2001. 2