Fits Like a Glove: Rapid and Reliable Hand Shape Personalization

David Joseph Tan[1,2] Thomas Cashman[1] Jonathan Taylor[1] Andrew Fitzgibbon[1]
Daniel Tarlow[1] Sameh Khamis[1] Shahram Izadi[1] Jamie Shotton[1]
[1]Microsoft Research [2]Technische Universität München

Abstract

We present a fast, practical method for personalizing a hand shape basis to an individual user's detailed hand shape using only a small set of depth images. To achieve this, we minimize an energy based on a sum of render-and-compare cost functions called the golden energy. *However, this energy is only piecewise continuous, due to pixels crossing occlusion boundaries, and is therefore not obviously amenable to efficient gradient-based optimization. A key insight is that the energy is the combination of a smooth low-frequency function with a high-frequency, low-amplitude, piecewise-continuous function. A central finite difference approximation with a suitable step size can therefore jump over the discontinuities to obtain a good approximation to the energy's low-frequency behavior, allowing efficient gradient-based optimization. Experimental results quantitatively demonstrate for the first time that detailed personalized models improve the accuracy of hand tracking and achieve competitive results in both tracking and model registration.*

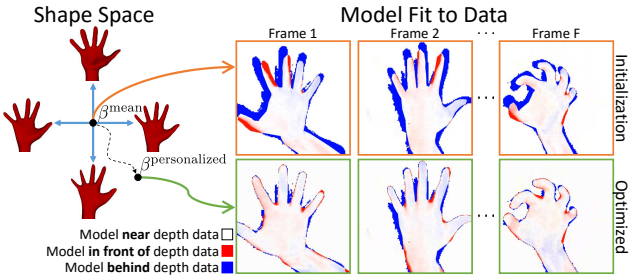

Figure 1: We show how to fit a deformable hand shape basis model [17] to a small set of depth images. Our method jointly optimizes over the shape $\beta \in \mathbb{R}^K$ and F poses θ_f to maximize the model's alignment to the data in F depth images. The initial hand poses are automatically determined by a hand tracker that uses the mean shape β^{mean}, but there is clearly poor alignment between model and data. After our optimization to obtain personalized shape $\beta^{\text{personalized}}$, the alignment is much better, with remaining errors largely due to sensor noise.

1. Introduction

The ability to accurately and efficiently reconstruct the motion of the human hand from images promises exciting new applications in immersive virtual and augmented realities, robotic control, and sign language recognition. There has been great progress in recent years, especially with the arrival of consumer depth cameras [16, 25, 26, 28, 29, 30, 32, 33, 36]. However, it remains a challenging task [31] due to unconstrained global and local pose variations, frequent occlusion, local self-similarity, and a high degree of articulation.

Most recent approaches combine the best of discriminative and generative approaches: the 'bottom-up' discriminative component attempts to make a prediction about the state of the hand directly from the image data, which then guides a 'top-down' generative component by deforming the parameters of a model to try to explain the data. Discriminative methods can be faster and typically require no temporal history. In contrast a good generative model can use its explanatory power and priors to produce what is usually a more accurate result, even in the presence of occlusion.

Generative models of hands are limited by their capacity to accurately explain the image observations. High-quality, though expensive and off-line, models have been shown to reliably fit both the pose and shape of complex sequences [4]. However, most interactive (real-time) hand tracking systems (*e.g.* [25, 30]) approximate the hand surface using primitives such as spheres or cylinders, parameterized to articulate the surface geometry. Others [28] use a detailed hand mesh model, though only attempt to fit the hand poses using a fixed template shape. To improve the model's capacity, some approaches [22, 30] allow shape deformations of primitive spheres and cylinders, but these models can only compensate for gross model-data mismatches.

Recent work [35] has investigated an off-line process for 'personalizing' a detailed 3D mesh model to an individual's hand shape using a set of depth images of the hand in varied poses, each paired with a manually-annotated initialization pose. The mesh shape is optimized to jointly explain the

depth data from each frame, yielding the user's personalized model. Unfortunately, this system is likely to be too brittle and slow for an online setting, as the parameterization of each mesh vertex yields a very high-dimensional optimization problem,

A promising alternative is to create a much lower-dimensional model that parameterizes the hand shape of an entire population of individuals. Khamis *et al*. [17] take a cue from the human body shape modeling literature [3, 14] and build a detailed 3D shape basis for human hands by parameterizing a mesh model using a small set of 'shape coefficients'. Each setting of these coefficients induces a hand model whose deformations are parameterized by a set of semantically meaningful pose parameters (*e.g*. joint angles). Unfortunately, even though Khamis *et al*. [17] show how to personalize their model for a new user, the lack of a 'background penalty' leaves local minima where the model has grown unrealistically in an attempt to explain the data. To avoid these local minima, they rely on a high-quality initialization that would be difficult to obtain reliably in an online setting. Further, they did not investigate whether the use of a personalized model was important for the accuracy of online hand tracking systems.

In this paper, we address these concerns and show how to use the trained shape basis from [17] to robustly personalize to an individual in a quick and easy calibration step. As illustrated in Fig. 1, our approach fits a single set of shape coefficients β and per-frame poses $\{\theta_f\}_{f=1}^{F}$ to a set of F depth images (each supplied with a rough initialization pose given by a template-based hand tracking system [28]). To do so, we exploit the 'golden energy' from [28], whose 'render-and-compare' formulation implicitly penalizes protrusions into free space. The energy appears to be the combination of a smooth low-frequency function with a high-frequency, low-amplitude, piecewise-continuous function (see Fig. 4). The discontinuities in the latter function are the result of occlusion boundaries travelling across locations being discretely sampled by each pixel. This seems to preclude gradient-based optimization, as following the exact gradient on either side of such a jump would not generally yield a good step direction.

One optimization option might be stochastic search (*e.g*. Particle Swarm Optimization) to avoid relying on derivatives, but this converges slowly and typically only works well for low-dimensional optimization problems. Our optimization space (one shape and F poses) is high-dimensional, however, and thus we would like to use a gradient-based optimizer. Although we could carefully work out the true derivatives of a continuous form of this energy [10], it is not obvious if we could compute them quickly. We thus choose to instead use an approximate derivative calculated using central differences. The step size must be right: large enough to jump over nearby occlusion boundaries, and small enough to capture the smooth global behavior of the function. We use a GPU-based tiled renderer to rapidly perform the extra function evaluations that this finite differencing requires. Given our ability to calculate the golden energy and calculate approximate derivatives, we are able to exploit Levenberg-Marquardt to minimize the energy in under a second for a small set of images (*e.g*. $F = 5$).

We can therefore demonstrate for the first time the potential for *detailed* personalization to quantifiably improve the accuracy of a real-time hand tracker. To this end, we adapt [28] to track using the personalized model, and compare template to personalized model tracking accuracy across several datasets. We show that our personalized hand tracking is able to achieve results that are competitive with the state of the art.

2. Related Work

A large amount of work has been done constructing detailed low-dimensional models of shape and pose variation for human bodies and faces [1, 2, 6, 9, 12, 13, 18, 19, 37, 38]. While hands may be similar to human bodies in the number of degrees of freedom, hands exhibit significantly more self-occlusion. They are also much smaller, which means images from current depth cameras contain fewer foreground pixels and suffer from more camera noise. Additionally, the space of hand poses is likely larger than that of the space of body poses. Consequently, it is only recently that similar detailed low-dimensional models were built for human hands [17]. Given various RGB-D sensor measurements, these approaches aim to find the low-dimensional shape and pose subspaces by fitting the entire set of observed data. This typically amounts to optimizing a very large number of parameters [7, 17, 21]. Despite the success of these approaches, the number of parameters prohibits their suitability for online *fitting*, although some systems may be close [21].

Recently, morphable subdivision surface models have been used to model other categories of deformation. Cashman and Fitzgibbon [8] demonstrate that extremely limited data (30 silhouette images) can be used to learn such a model for a variety of objects and animals. In more closely-related work, Taylor *et al*. [35] learn a personalized hand model from a set of noisy depth images for a single user, which was the approach adapted by Khamis *et al*. [17] to train a hand shape model on a large dataset of hands.

Other related work tackles differentiation for a render-and-compare energy function, which may at first seem unapproachable due to occlusion boundaries. When the image domain is kept continuous, however, one can show that such energies are naturally differentiable and their exact gradient can be laboriously worked out [11]. Nonetheless, current practical systems discretize the image domain by taking a point sample at each pixel, which introduces discontinuities in the energy caused by occlusion boundaries moving from

pixel to pixel. In order to avoid such difficulties, it is tempting to instead approximate the gradient by peering behind these boundaries [5]. Interestingly, Oberweger *et al.* [24] side-stepped this issue completely by training a convolutional neural network to render hands, as gradients are then easily obtainable using the standard back-propagation rules for such networks.

3. Shape and Pose Model

The model developed by Khamis *et al.* [17] parameterizes both hand pose $\theta \in \mathbb{R}^{28}$ and hand shape $\beta \in \mathbb{R}^K$ to deform an M-vertex triangular mesh, assumed to have a fixed triangulation and hierarchical skeleton. This deformation proceeds in three steps, the first two of which are illustrated in Fig. 2.

First, a vector β of shape coefficients produces a mesh of a hand in a neutral pose, but with a specific hand shape. Simultaneously, the shape also defines the position of the B bones of the skeleton. To be precise, given β, the locations of M vertices fill the columns of the $3 \times M$ matrix $V(\beta)$, and the set of bone locations fill the columns of the $3 \times B$ matrix $L(\beta)$:

$$V(\beta) = \sum_{k=1}^{K} \beta_k V_k \quad \text{and} \quad L(\beta) = \sum_{k=1}^{K} \beta_k L_k \,. \tag{1}$$

The matrices $\{V_k, L_k\}_{k=1}^K$ thus form a linear basis for the shape of the model. These are the same bases as [17] for all values of $K \in \{1, 2, 3, 4, 5\}$ for which they trained. Note that the regularization used during the training process encouraged the first dimension (V_1, L_1) to represent something akin to a mean hand and skeleton with the other dimensions serving as offsets. We therefore call $\beta^{\text{mean}} = [1, 0, \ldots, 0]^\top \in \mathbb{R}^K$ the 'mean' hand shape (see Fig. 1).

Second, the model applies a linear blend skinning (LBS) operator $P(\theta; V, L) \in \mathbb{R}^{3\times M}$ to a mesh V and skeleton L using a set of pose parameters $\theta \in \mathbb{R}^{28}$ that include global rotation, translation, wrist and finger joint rotations. LBS is a standard tool in computer animation; we refer the reader to [17] for details.

Third, and as a new addition to [17], we implement a final step $\Gamma : \mathbb{R}^{3\times M} \to \mathbb{R}^{3\times M'}$ that applies a single step of Loop subdivision [20] to the mesh to produce a denser mesh with M' vertices. This brings the resulting mesh into closer alignment with the true 'limit surface' that was fitted to the data in [17], while maintaining efficiency for what follows.

For notational clarity, we combine the steps together as

$$\Upsilon(\theta, \beta) = \Gamma(P(\theta; V(\beta), L(\beta)) \in \mathbb{R}^{3\times M'} \tag{2}$$

to denote the full deformation model that produces a subdivided mesh with shape β in pose θ.

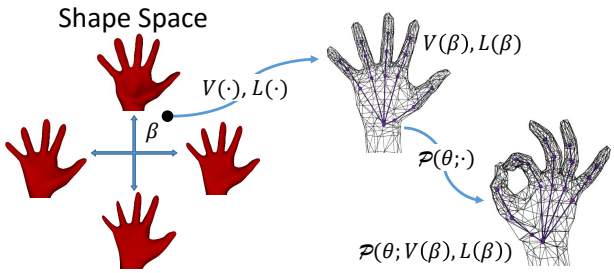

Figure 2: The model of hand shape deformation [17].

4. The Golden Energy

One way to evaluate whether a specific combination of shape β and pose θ give rise to an image is to simply render the deformed mesh $\Upsilon(\theta, \beta)$ and compare it to the image. If this evaluation can be formulated as an energy function that assigns a low value when the rendered and observed images are close, the problem is then reduced to function minimization.

To this end, we adapt the 'golden energy' from [28] in two ways: (i) we use an L^2 penalty (instead of L^1) to allow the use of standard least-squares optimization techniques; and, (ii) at least conceptually, we operate on a continuous pixel domain $\mathcal{I} \subseteq \mathbb{R}^2$ to model the idealized imaging process [10]. We thus define an idealized energy by simply integrating the difference between the observations and the rendering across the domain of the image $\mathcal{I}$

$$\hat{E}_{\text{gold}}(\theta, \beta) = \int_{(u,v)\in\mathcal{I}} \rho(\tilde{I}(u, v) - \tilde{R}(u, v; \Upsilon(\theta, \beta)))^2 \,\mathrm{d}u\,\mathrm{d}v \tag{3}$$

where $\rho(e) = \min(\sqrt{\tau}, |e|)$ with a constant truncation threshold τ. Here, $\tilde{I}(u, v)$ and $\tilde{R}(u, v; \Upsilon(\theta, \beta))$ give the observed and the rendered depth at the location (u, v), respectively. Note that we generally observe a discretized image and thus $\tilde{I}(u, v)$ will be piecewise constant.

In practice, the integral in (3) is difficult and expensive to evaluate so practical systems instead create a discretization by rendering an image of size $W \times H$. The (discretized) golden energy is thus given by

$$E_{\text{gold}}(\theta, \beta) = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} r_{ij}(\theta, \beta)^2 \tag{4}$$

with the residual $r_{ij}(\theta, \beta)$ for pixel (i, j) defined as

$$r_{ij}(\theta, \beta) = \rho(I_{ij} - R_{ij}(\Upsilon(\theta, \beta)) \tag{5}$$

where $I \in \mathbb{R}^{W\times H}$ is appropriately resampled from $\tilde{I}(\cdot, \cdot)$ and $R_{ij}(\Upsilon(\theta, \beta))$ yields the value of pixel (i, j) in the rendered depth image.

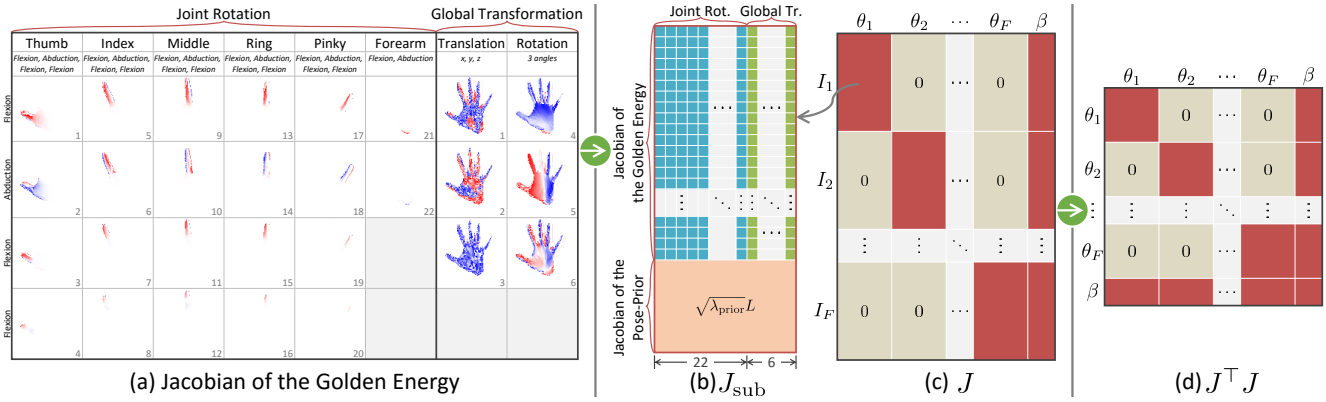

Figure 3: (a) Visualization of the Jacobian with respect to pose parameters θ. Each image is reshaped to form a column of J. (b) Rows in J_{sub} represent subterms in the energy; columns represent the pose parameters for one frame. (c) Jacobian of the full lifted energy E', including the shape parameters β. (d) Sparsity structure of $J^\top J$.

5. Shape Fitting Energy

We now have the tools that we need to attack the problem of inferring a user's hand shape β from a sequence of depth images $\{I_f\}_{f=1}^F$. To achieve this goal, we want to minimize

$$E(\beta) = \sum_{f=1}^{F} \min_{\theta} \left(E_{\text{gold}}(\theta, \beta; I_f) + \lambda_{\text{prior}} E_{\text{prior}}(\theta) \right). \quad (6)$$

To make the resulting value small, a pose θ must be found for each frame that yields both a low golden energy $E_{\text{gold}}(\theta)$ and a low pose prior energy $E_{\text{prior}}(\theta)$. The pose prior provides constraints on the pose in the form of the negative log-likelihood

$$E_{\text{prior}}(\theta) = (\theta - \mu)^\top \Sigma^{-1} (\theta - \mu) \quad (7)$$

of a multivariate normal $\mathcal{N}(\mu, \Sigma)$. The mean $\mu \in \mathbb{R}^{28}$ and covariance matrix $\Sigma \in \mathbb{R}^{28\times 28}$ were fitted to a selected set of valid hand poses $\{P_q^{\text{train}}\}_{q=1}^Q \subseteq \mathbb{R}^{22}$ captured using the hand tracker of [28], with the variance on the global pose set to ∞.

6. Optimization

Using the standard 'lifting' technique (see *e.g.* [17]), we define a new lifted energy

$$E'(\Theta, \beta) = \sum_{f=1}^{F} E_{\text{gold}}(\theta_f, \beta) + \lambda_{\text{prior}} E_{\text{prior}}(\theta_f) \quad (8)$$

where $\Theta = \{\theta_f\}_{f=1}^F$. As $E(\beta) \leq E'(\Theta, \beta)$ for any value of Θ, we seek to implicitly minimize the former by explicitly minimizing the latter. For simplicity, we assign $x = \begin{bmatrix}\text{vec}(\Theta) & \beta^\top\end{bmatrix}^\top \in \mathbb{R}^{28F+K}$ as the parameter vector.

Note that E' has $28F + K$ parameters, and thus would be very difficult to optimize using a stochastic optimizer like PSO [28]. Instead, we use Levenberg-Marquardt, a gradient-based optimizer that can yield second-order-like convergence properties when close to the minimum.

The optimizer requires the full Jacobian matrix J of the residuals with respect to the $28F + K$ parameters (see Fig. 3(a-c)). Given the independence of the pose parameters across the F depth images (we do not assume any ordering or temporal continuity in the depth images, only that they come from the same individual), it follows that $28F$ columns of J are sparsely filled by the results of the pixel-wise derivative of the golden energy from a single image I_f with respect to a pose parameter in θ_f (see Sec. 7). This is combined with the Jacobian matrix of the pose prior energy. The shape coefficients, however, are the same for all images, so the column that corresponds to a shape coefficient in J is the concatenation of the pixel-wise derivative of the golden energy from all images.

To find the Jacobian matrix associated to E_{prior}, we first use Cholesky decomposition on $\Sigma^{-1} = LL^\top$ and rewrite the energy as

$$E_{\text{prior}}(\theta) = \|L(\theta - \mu)\|^2. \quad (9)$$

Since we are computing the derivative of the residuals, the Jacobian matrix of $E_{\text{prior}}(\theta)$ with respect to the parameters is simply L. In addition to the pose prior, we also impose box constraints on the parameters θ to restrict the hand pose from unnatural or impossible deformations. These constraints take the form of limiting values $[P^{\min}, P^{\max}] \in \mathbb{R}^{28} \times \mathbb{R}^{28}$, which we impose using the projection Π such that $\Pi(x)_i = \min(\max(P_i^{\min}, x_i), P_i^{\max}))$.

Then, using the Levenberg-Marquardt method with a projected step [15], we propose the following update of the parameters

$$x^{\text{prop}} = \Pi(x - (J^\top J + \gamma\, \text{diag}(J^\top J))^{-1} J^\top r) \quad (10)$$

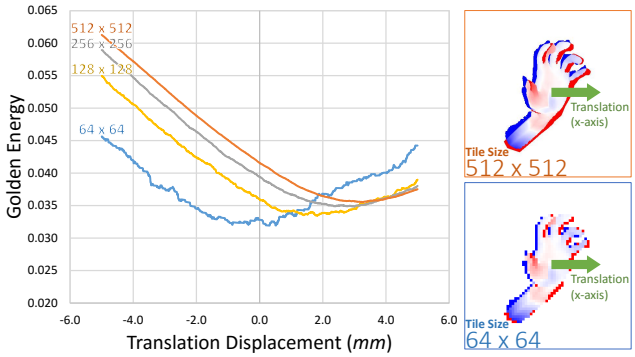

Figure 4: Golden energy as a function of x-axis translation, for different rendered tile sizes $W \times H$. Note the globally smooth nature but local discontinuities, which occur at an increasingly small scale with larger tile sizes.

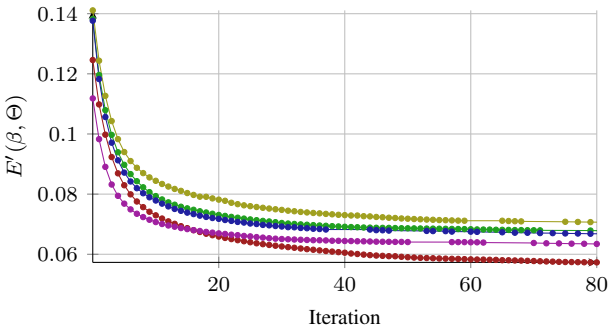

Figure 5: Convergence of E' for the five subjects in the FingerPaint dataset. Dots represent successful Levenberg-Marquardt iterations.

where $J^\top J$ is a sparse matrix as illustrated in Fig. 3(d). If $E(x^{\text{prop}}) < E(x)$, then the update is accepted $x \leftarrow x^{\text{prop}}$ and the damping is decreased $\gamma \leftarrow 0.1\gamma$. Otherwise, the damping is increased $\gamma \leftarrow 10\gamma$ and the proposal is recalculated. Eventually, progress will be made as this is effectively performing a back-tracking line search while interpolating from Gauss-Newton to gradient descent.

The importance of the pose prior in our energy becomes more evident in self-occluded poses where the fingers or forearm are not visible in the rendered image. When performing a finite difference with respect to transformation parameters, zero pixel residuals can occur. Thus, without the pose prior, J and $J^\top J$ become rank-deficient. By including the pose prior, the angles of the occluded joints approach the conditional mean of the occluded joints given the visible joints as they remain unobserved by the image.

7. Differentiating the Golden Energy

Note that (4) is only piecewise continuous (see Fig. 4), as moving occlusion boundaries cause jumps in the value of rendered pixels. Our desired optimization procedure requires gradients (see Sec. 6), but it is evident that the exact derivative of E_{gold} at any specific point of our approximation will generally not be helpful. One option would be to return to the idealized continuous energy [11]. However, the edge overdraw antialiasing used is considerably more expensive than a simple render on the GPU. Another approximation [5] is engineered to look behind the occlusion boundary to try to anticipate what will come into view. Nevertheless, we take a different approach that lets us exploit standard GPU-accelerated rendering techniques.

To this end, we note that the curves in Fig. 4 appear to be the combination of a well-behaved smooth function at a global scale and a low-amplitude non-smooth function at a local scale. If we could somehow recover the former, its gradient would provide a good candidate direction for minimizing (4). One option would be to try to smooth out the discontinuities in the approximation using a Gaussian kernel, but this would require the function's evaluation at positions across the entire basin of support of the kernel. For efficiency, we therefore attempt to approximate the function locally by fitting a line to two points that are sufficiently far from each other as to capture the dominant smooth behavior of the energy. Hence, we assign $\phi = \begin{bmatrix}\theta^\top & \beta^\top\end{bmatrix}$ with the parameters associated to an image and approximate the gradient using central differences

$$\frac{\partial E_{\text{gold}}(\phi)}{\partial \phi_k} \approx \frac{E_{\text{gold}}(\phi + \frac{\Delta_k}{2}) - E_{\text{gold}}(\phi - \frac{\Delta_k}{2})}{\epsilon_k} \tag{11}$$

where the constant step size ϵ_k is set empirically (see Table 1) and the value of the kth element of the vector $\Delta_k \in \mathbb{R}^{28+K}$ is set to ϵ_k while zero elsewhere.

As with (4), the residual at pixel (i, j) is only piecewise continuous, although with a sparser set of more dramatic jumps. Similarly then, we find that a central difference with a large step size allows us to approximate the derivative of the residual

$$\frac{\partial r_{ij}(\phi)}{\partial \phi_k} \approx \frac{r_{ij}(\phi + \frac{\Delta_k}{2}) - r_{ij}(\phi - \frac{\Delta_k}{2})}{\epsilon_k} . \tag{12}$$

Although one might be concerned about the various approximations above, our use of Levenberg-Marquardt provides a safeguard against catastrophic failure. When steps fail, the algorithm implicitly performs a back-tracking line search as it interpolates from Gauss-Newton to gradient descent. This means that in the worst case, the approximate gradient need only point uphill for progress to be made. In practice, however, we find the approximate derivatives to work quite robustly resulting in few rejected steps, indicated by the many dots (acceptances) in the convergence plots in both Fig. 5 and Fig. 6.

Parameter (each row maps to several ks)	Step size
X, Y and Z translations	10mm
X rotation	5°
Y and Z rotations	2.5°
Metacarpal-phalangeal joint flexions	5°
Metacarpal-phalangeal joint abductions	5°
Proximal interphalangeal joint flexions	10°
Distal interphalangeal joint flexions	15°

Table 1: Step sizes ϵ_k used in central differences (12).

8. Experimental Results

We use both synthetic and real data to elucidate our effectiveness at rapidly minimizing our shape-fitting energy. We show that this shape calibration gives us an accuracy improvement on three separate datasets and that our results are competitive with the state of the art. We refer the reader to the supplementary material for more experiments and a video of the live system in action.

For all experiments, we use the step sizes in Table 1 to calculate finite differences, a tile size of 256×256 pixels, which gave a good balance of global smoothness and performance (see Fig. 4), and a truncation threshold $\sqrt{\tau} = 10$cm. While one could minimize our energy using LM for *tracking* (as opposed to shape calibration), it performs only a fairly local optimization. Instead we use an implementation of [28][1], augmented with our own pose prior.

Synthetic Ground Truth. We begin with an experiment on synthetic data to evaluate our optimization strategy and its ability to find a good hand shape. To this end, we randomly choose a ground truth shape $\beta^{\text{gt}} \in \mathbb{R}^K$. We then sample a set of $F = 40$ poses $\Theta^{\text{gt}} = \{\theta_f^{\text{gt}}\}_{f=1}^F$ from our pose prior, and render a set of depth images $\{I_f\}_{f=1}^F$. We then initialize our energy minimization at the mean with $\beta = \beta^{\text{mean}}$. In Fig. 6, we show the convergence when we optimize $E(\Theta, \beta)$. One can see in Fig. 6 (left) that we rapidly descend the energy landscape in the first 20 iterations. This is clearly correlated with a rapid reduction of $|\beta_1 - \beta_1^{\text{gt}}|$ to near zero, which shows that we quickly obtain the correct scale. Due to the way the shape basis was trained in [17], β_1 is in a unit that roughly corresponds to the scale of the mean hand whereas the units of the other components are less interpretable. Nonetheless, one can see in the right of Fig. 6 that once scale (*i.e.* β_1) is taken care of, the error in these components is lowered to refine detail. Fig. 7 shows that minimizing the energy also gives strong agreement between the vertex positions $V(\beta)$ and the corresponding ground truth positions $V(\beta^{\text{gt}})$.

Marker Localization. We now begin exploring the usefulness of our shape calibration procedure in improving

[1]Despite statements to the contrary [22], [28] optimizes over pose only.

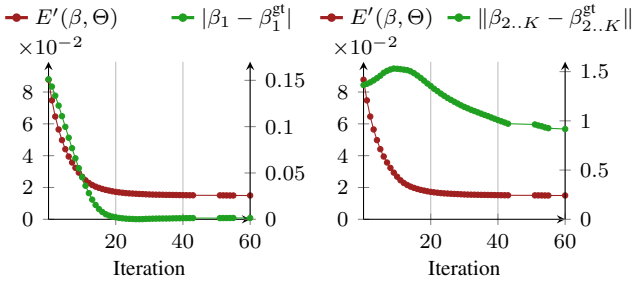

Figure 6: Left: Optimizing E' improves the estimate of β_1 which roughly corresponds to scale. Right: The same for the remaining coefficients of β. Dots show successful Levenberg-Marquardt steps.

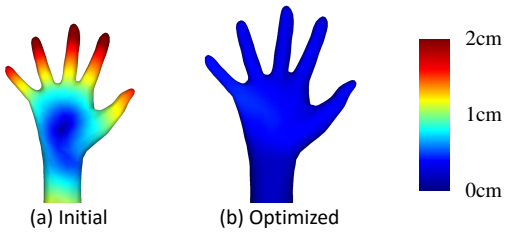

Figure 7: Heat maps showing the distance of each vertex to the corresponding ground truth position, for the (a) initial and (b) final iteration of the synthetic experiment (Fig. 6).

tracking accuracy, for which the most common metric is prediction error for a set of marker positions that localize semantic points on the hand. As these locations differ between datasets, we need to create a mapping from the combined shape-and-pose parameters ϕ to a marker position. To do so, for each marker $t = 1, \ldots, T$ we identify four vertices on the correct region of the model using a fixed picking matrix[2] $Y_t \in \mathbb{R}^{4\times M}$, and define an affine combination of these vertices using the barycentric coordinates $w_t \in \mathbb{R}^4$ with $\sum w_t = 1$. We then solve

$$w_t = \underset{w}{\text{argmin}} \sum_{f \in H} \|P(\theta_f; V(\beta), L(\beta))\, Y_t^\top w - G_{ft}\|^2$$

where $G_{ft} \in \mathbb{R}^3$ is the ground truth location of marker t in frame f, and $H \subseteq \{1, ..., N\}$ is an equally spaced 5% sampling of the N frames in the dataset.

NYU Dataset. We test our method on the popular NYU Hand Pose dataset [36], which comprises N = 8,252 test frames with captures of two different subjects (*i.e.* only two different shapes). Each frame is provided with ground truth locations G_{ft} for 36 positions of the hand. To compute 3D error for Tompson *et al.* [36] on this dataset, we follow recent papers [23, 24, 27] that augment the inferred 2D positions

[2]A picking matrix contains zeros except for a single unity entry per row.

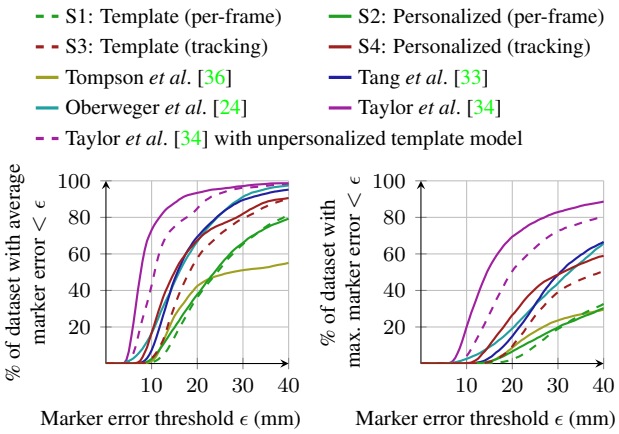

Figure 8: Marker localization error on NYU dataset.

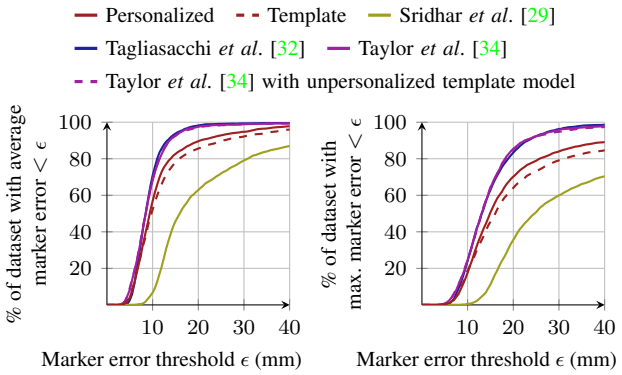

Figure 9: Marker localization error on Dexter dataset. The results for this dataset have been normalized so that each of the 7 sequences has equal weight.

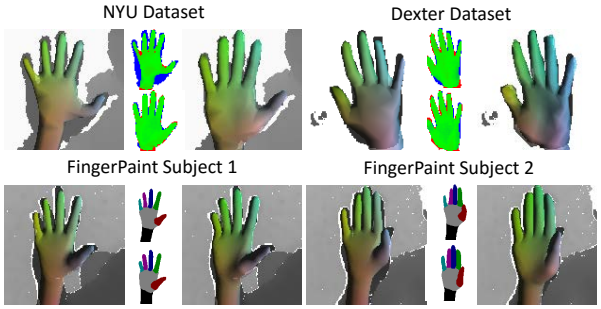

Figure 10: Qualitative example of fit difference between template (left and top-middle of each set) and personalized model (bottom-middle and right of each set) for one subject of the NYU (top left), the only subject of the Dexter (top right) and two subjects of the FingerPaint (bottom) datasets.

with the captured depth at each location where valid, and the ground truth depth otherwise. We also obtained inferred positions from Tang *et al*. [33], Oberweger *et al*. [24] and Taylor *et al*. [34], selecting a common subset of $T = 10$ positions (2 per digit) for comparison between all methods.

We give quantitative results for four different settings (S1-4) in Fig. 8. (S1) Since Tompson *et al*. use no temporal information to estimate hand pose, we also configure the tracker [28] to rely only on its discriminative initializer to seed each frame independently. (S2) We used $F = 20$ evenly distributed poses output by (S1) to initialize our calibration and create a $K = 5$ personalized model for each of the two subjects. We then re-ran (S1) using the appropriate personalized model. (S3) We re-enable temporal coherency in the hand tracker (a more realistic setting for tracking), and report the result using the template. (S4) We follow the same procedure as in (S2) but using poses from (S3) to create personalized models. Again, we report the result when tracking is run using the appropriate personalized model.

Notice first that our personalized tracker provides a result comparable to Tang *et al*. [33]. This machine learning approach was trained directly on the NYU training set, and thus benefits from the reduced search space induced by this largely front-facing, limited pose variation dataset. Second, the personalized tracker provides a much better result than the template tracker. We hypothesize that the superior fit of the personalized model (see Fig. 10) creates a much deeper 'correct' local minimum closer to the true pose, making it easier to find the deep 'correct' local minimum in the next frame. In contrast, personalization does not assist as much when temporal coherence is turned off. Nonetheless, our calibration tool lets us simply upgrade the performance of a compatible tracker with a personalized model. The recent result from Taylor *et al*. [34], using the same personalized models as our result, shows the the accuracy of a *gradient-based* hand tracker when combined with our personalization.

Dexter Dataset. We use the Dexter dataset [29] to further evaluate our personalized tracking against state-of-the-art results. We follow the recommendation of the authors and use $T = 5$ fingertip markers, excluding a small number of frames from the beginning of each sequence for the purpose of calculating error. The result is a total of $N = 2{,}931$ frames, of which we use an equally-spaced subset of $F = 20$ frames to personalize the model. Fig. 9 compares our tracker with Sridhar *et al*. [29] (see supplementary material for more detail on this comparison), as well as Tagliasacchi *et al*. [32] and Taylor *et al*. [34]. This time, personalization gives a lesser improvement in tracking accuracy as the template fits the single subject's hand quite well (see Fig. 10).

FingerPaint Dataset. To test our ability to perform detailed surface registration, we turn to the hand part segmentation task required for the FingerPaint dataset [28]. The dataset includes sequences from five different subjects, with pixels labelled as one of 7 parts (5 fingers, the palm, and the

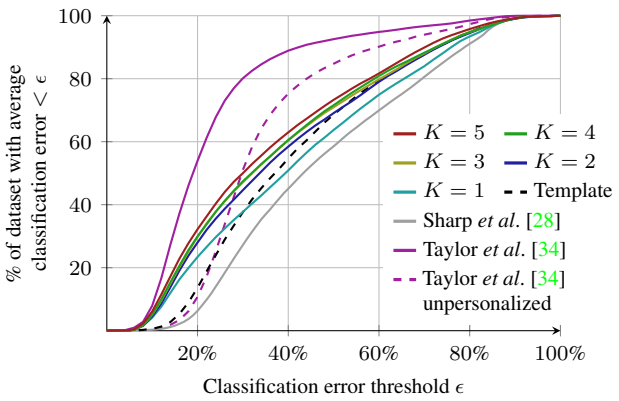

Figure 11: Classification error on FingerPaint dataset.

forearm). To personalize to each subject, we first run the template-based hand tracker across each sequence. Then, for each subject, we sample $F = 30$ frames, evenly distributed throughout the dataset, and use the poses to run our shape optimization. For this dataset, we try personalizing using $K = \{1, \ldots, 5\}$ for the 5 different shape models from [17]. We then run the tracker on the dataset using the appropriate personalized models, and compare the pixel classification accuracies (see supplementary material and Fig. 10 for examples of these personalized models). Fig. 11 shows, as expected, the average classification accuracy increases as we increase K as the deformation model can more accurately register itself to the data. Interestingly, the $K = 1$ curve which roughly corresponds to a scaled mean hand does not always perform better than the template. We hypothesize that in these areas of the curve, any benefits to personalization are not able to compensate for the bias caused by fitting to a different dataset; in contrast the template is implicitly not biased to any dataset as it was created by hand. Note that the pose prior explains the improvement in accuracy seen between template tracking and Sharp *et al*. [28] in Fig. 11.

Qualitative System. Finally, we show that our shape calibration procedure can be used in an online tracker to provide rapid and reliable detailed personalized tracking for any user (see Fig. 12). We augmented the live tracker of [28] to include the capability to perform an online personalization of the model (see Fig. 13). The system starts by using the template model to track the user's hand. The user moves their hand into F different poses, and when the user is comfortable that the tracker has a reasonable pose estimate, a button is pressed to capture both the depth frame and the pose estimate. When satisfied with these poses, the user presses a button to initiate shape calibration. Typically, this procedure takes less than a second, at which point the new personalized model is used for further tracking. This immediately allows applications to benefit from both improved surface-to-data

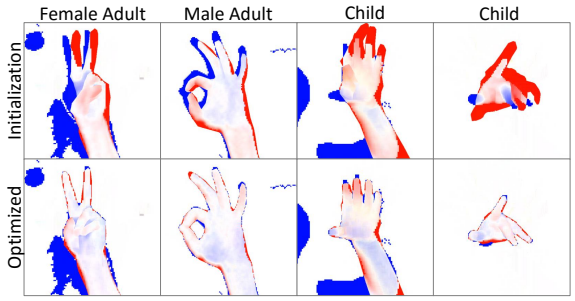

Figure 12: Calibration frames at initialization and after convergence of our personalization procedure. The template is the wrong shape for the female subject, too small for the male and wildly too large for the two children. After personalization, each model fits each user 'like a glove'. The truncated golden energy makes the system robust to errors in segmenting the background.

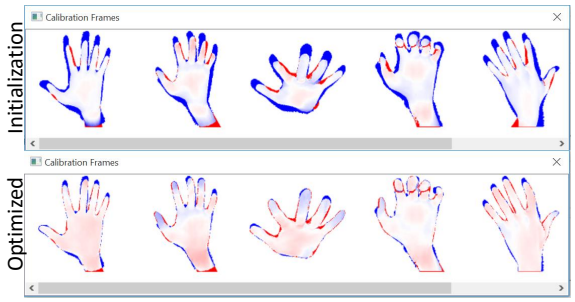

Figure 13: Our online calibration tool showing (top) that the alignment of the mean shape to calibration frames shows gross errors and (bottom) that the personalized model tightly aligns with the data after optimization.

registration (see 'FingerPaint Dataset') and tracking accuracy (see 'NYU Dataset'). See the supplementary material for videos of this system in live use.

9. Conclusion

We have presented the first online method for creating a *detailed* 'personalized' hand model for hand tracking. An easy-to-use calibration step allows a new user to rapidly transition from template to personalized tracking, yielding more robust tracking and better surface alignment that can be exploited by higher-level applications. We have experimentally verified both of these benefits on several standard datasets, showing the increase in both marker localization and dense pixel classification accuracy one obtains when a personalized model is used in place of a poorly-fit template model. Users found our calibration system easy to use and compelling to see a detailed hand avatar. We leave it as future work to address the question of how to remove the calibration step entirely and make personalization fully automatic.

References

[1] B. Allen, B. Curless, and Z. Popović. Articulated body deformation from range scan data. *ACM Trans. Graphics*, 21(3):612–619, 2002. 2

[2] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graphics*, 22(3):587–594, 2003. 2

[3] 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):408–416, 2005. 2

[4] 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*, pages 640–653, 2012. 1

[5] M. Black and M. Loper. OpenDR: An approximate differentiable renderer. In *Proc. ECCV*, pages 154–169, 2014. 3, 5

[6] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, pages 187–194, 1999. 2

[7] F. Bogo, M. J. Black, M. Loper, and J. Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Proc. ICCV*, pages 2300–2308, 2015. 2

[8] T. J. Cashman and A. W. Fitzgibbon. What shape are dolphins? Building 3D morphable models from 2D images. *IEEE Trans. PAMI*, 35(1):232–244, 2013. 2

[9] Y. Chen, Z. Liu, and Z. Zhang. Tensor-based human body modeling. In *Proc. CVPR*, pages 105–112, 2013. 2

[10] M. de La Gorce, D. J. Fleet, and N. Paragios. Model-based 3D hand pose estimation from monocular video. *IEEE Trans. PAMI*, 33(9):1793–1805, 2011. 2, 3

[11] M. de La Gorce, N. Paragios, and D. J. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *Proc. CVPR*, pages 1–8, 2008. 2, 5

[12] A. Delaunoy and E. Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3D reconstruction problems dealing with visibility. *IJCV*, 95(2):100–123, 2011. 2

[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*, 28(2):337–346, 2009. 2

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

[15] C. Kanzow, N. Yamashita, and M. Fukushima. Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *Journal of Computational and Applied Mathematics*, 172(2):375–397, 2004. 4

[16] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proc. ECCV*, pages 852–863, 2012. 1

[17] S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, and A. Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Proc. CVPR*, pages 2540–2548, 2015. 1, 2, 3, 4, 6, 8

[18] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum*, 27(5):1421–1430, 2008. 2

[19] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *Proc. ICCV*, pages 167–174, 2009. 2

[20] C. T. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Aug. 1987. 3

[21] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics*, 34(6):#248, 2015. 2

[22] A. Makris and A. Argyros. Model-based 3D hand tracking with on-line hand shape adaptation. In *Proc. BMVC*, pages 77.1–77.12, 2015. 1, 6

[23] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. In *Proc. Computer Vision Winter Workshop (CVWW)*, 2015. 6

[24] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *Proc. ICCV*, pages 3316–3324, 2015. 3, 6, 7

[25] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proc. BMVC*, pages 101.1–101.11, 2011. 1

[26] I. Oikonomidis, N. Kyriazis, and A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *Proc. CVPR*, pages 1862–1869, 2012. 1

[27] G. Poier, K. Roditakis, S. Schulter, D. Michel, H. Bischof, and A. A. Argyros. Hybrid one-shot 3D hand pose estimation by exploiting uncertainties. In *Proc. BMVC*, pages 182.1–182.14, 2015. 6

[28] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible realtime hand tracking. In *Proc. CHI*, pages 3633–3642, 2015. 1, 2, 3, 4, 6, 7, 8

[29] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Proc. CVPR*, pages 3213–3221, 2015. 1, 7

[30] S. Sridhar, H. Rhodin, H.-P. Seidel, A. Oulasvirta, and C. Theobalt. Real-time hand tracking using a sum of anisotropic Gaussians model. In *Proc. 3DV*, pages 319–326, 2014. 1

[31] J. S. Supančič III, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proc. ICCV*, pages 1868–1876, 2015. 1

[32] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust articulated-ICP for real-time hand tracking. *Computer Graphics Forum*, 34(5):101–114, 2015. 1, 7

[33] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proc. ICCV*, pages 3325–3333, 2015. 1, 7

[34] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton. Efficient and precise interactive

hand tracking through joint, continuous optimization of pose and correspondences. In *Proc. SIGGRAPH*, 2016. To appear. 7, 8

[35] 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*, pages 644–651, 2014. 1, 2

[36] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graphics*, 33(5):#169, 2014. 1, 6, 7

[37] A. Tsoli, N. Mahmood, and M. J. Black. Breathing life into shape: capturing, modeling and animating 3D human breathing. *ACM Trans. Graphics*, 33(4):#52, 2014. 2

[38] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Proc. CVPR*, pages 2353–2360, 2014. 2