Pacific Graphics 2020
E. Eisemann, A. Jacobson, and F.-L Zhang
(Guest Editors)

Volume 39 (2020), Number 7

Personalized Hand Modeling from Multiple Postures with Multi-View Color Images

Yangang Wang[1,2], Ruting Rao[1] and Changqing Zou[3†]

[1]School of Automation, Southeast University, Nanjing, China
[2]Shenzhen Research Institute of Southeast University, Shenzhen, China
[3] Huawei HMI Lab, Markham, Canada.

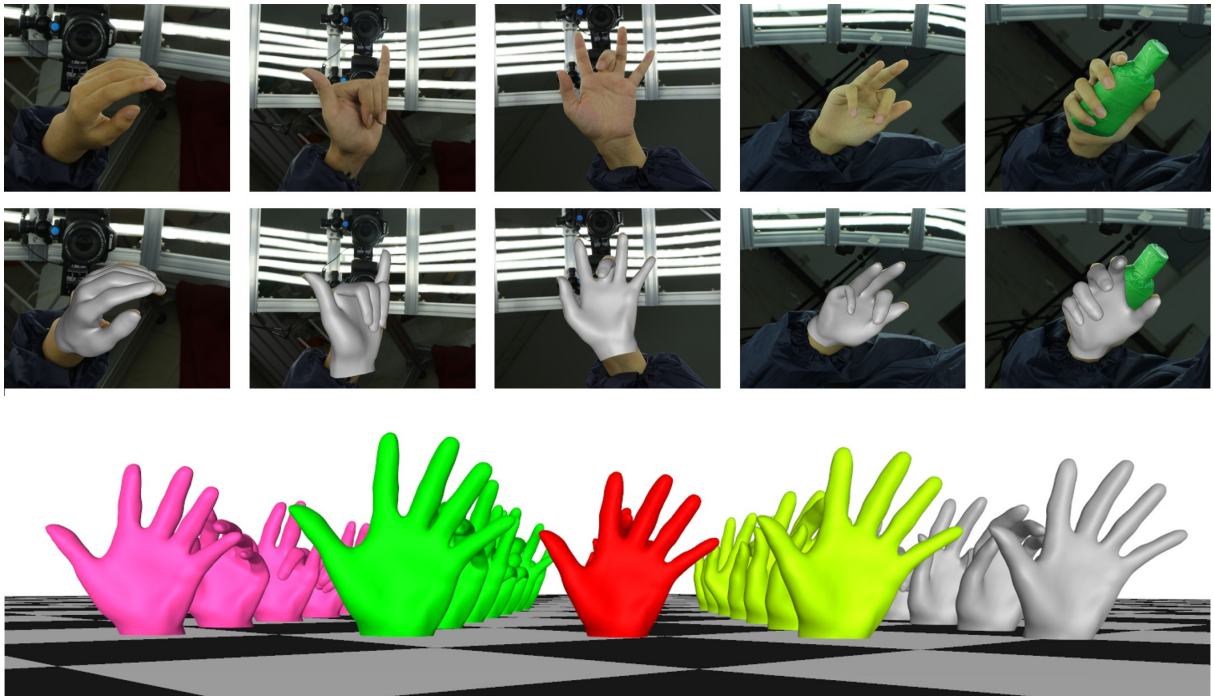


Figure 1: *Personalized hand modeling results from multiple hand postures. The top two rows show the estimated hand models which can be well overlaid with color images. The bottom row shows different personalized hand models of different hand postures. This paper proposes a novel method for personalized hand modeling from multiple hand postures with multi-view color images.*

Abstract

Personalized hand models can be utilized to synthesize high quality hand datasets, provide more possible training data for deep learning and improve the accuracy of hand pose estimation. In recent years, parameterized hand models, e.g., MANO, are widely used for obtaining personalized hand models. However, due to the low resolution of existing parameterized hand models, it is still hard to obtain high-fidelity personalized hand models. In this paper, we propose a new method to estimate personalized hand models from multiple hand postures with multi-view color images. The personalized hand model is represented by a personalized neutral hand, and multiple hand postures. We propose a novel optimization strategy to estimate the neutral hand from multiple hand postures. To demonstrate the performance of our method, we have built a multi-view system and captured more than 35 people, and each of them has 30 hand postures. We hope the estimated hand models can boost the research of high-fidelity parameterized hand modeling in the future. All the hand models are publicly available on www.yangangwang.com.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics;***

† Content of this paper should be specified as non-Huawei achievements.

1. Introduction

Hand pose estimation from color images and videos, whether it is 2D or 3D, is a long-standing problem in the area of computer graphics and computer vision. It can facilitate many imaginative applications, such as human-machine interaction, virtual reality, augmented reality and *etc*. In recent years, with the rapid development of deep learning techniques, hand pose estimation has gotten significant progress, *e.g*., [WZP20, ZHX*20]. Meanwhile, high quality hand datasets are becoming a crucial issue and attract lots of research interests in recent years. Some pioneers [MBS*18] have attempted to synthesize hand image datasets with the Generative Adversarial Network (GAN). However, there is still a huge gap between synthesized and real captured images. Reconstructing high-fidelity personalized hand models and rendering them with various backgrounds might be a strategy to promote the performance of hand pose estimation in the future.

In addition to provide more plausible high-quality training data to improve the accuracy of hand pose estimation, personalized hand modeling also plays an important role in the scenario of hand-object interactions. In some applications like robotics manipulation, reconstructing a personalized hand model is very important since a general hand model may encounter the problem of penetration or departure, which can lead to severe mistakes.

Previous works usually use two main routines to obtain personalized hand models. One relies on pre-defined parameterized hand models, *e.g*., MANO [RTB17], and infers proper model parameters by fitting the hand model with captured color images or depth data, or directly regresses the model parameters by neural networks [ZLM*19]. However, due to the low-resolution representation of existing parameterized hand models, it is still hard to obtain high-fidelity personalized hand models from highly compressed parameterized mathematical models. The other strategy is to adopt scanners or sensors [WMB19] to capture 3D hand information directly and reconstruct personalized hand models. Nevertheless, it is not trivial to register high quality 3D hand models. The main reason is that shape and posture deformation are always mixed together, and it is hard to require different people to perform the same posture. It remains a challenging topic to decompose the hand posture and shape for reconstructing high-fidelity personalized hand models.

In this paper, we introduce the idea of **personalized neutral hand**, as well as hand postures, to model the personalized hand. The differences of personalized hand models are addressed only by the neutral hand. A template hand model is adopted for the reference of building personalized neutral hands. For simplicity, we obtain hand models under different hand postures by linear blend skinning (other deformation techniques, such as [JBPS11] can be also adopted for better performance). In order to demonstrate the performance of the proposed method, we build a multi-view system with 15 color cameras as shown in Fig. 2. We captured the left and right hands of 35 persons and collected 30 hand postures for each person. A novel optimization strategy is proposed to estimate the neutral hand from multiple hand postures. With the estimated neutral hand as the initial guess, we iteratively optimize the hand postures and personalized neutral hand to achieve personalized hand modeling.

The main contributions of this paper are summarized as follows.

- We propose a novel method for personalized hand modeling from multiple color images.
- We propose an optimization strategy to obtain the personalized neutral hand from multiple hand postures.
- We have captured 35 people with 30 hand postures. The dataset is publicly available on our website.

2. Related Work

2.1. Registration

Registration plays an important role in shape estimation, tracking and 3D model building [LMR*15, LAGP09, SKR*15, CK15, BKL*16, BRPB17, VPAS19], which aligns 3D scans with each other or a template. Many methods [TCL*13] have been proposed in the last several decades, here we only discuss registration methods for the articulated objects like body and hands. Typical approaches exploit non-rigid ICP [LAGP09, CK15, Zha94, TST*15], but they may be sensitive to noisy data, causing alignment error in some cases. To make registration more accurate and robust, a template can be used for precise alignment [HLRB12, BRLB14, BBLR15], though temple-free methods [CZ11, WWV*16] are simpler. A novel mesh registration approach presented in FAUST [BRLB14] combines 3D shape and texture information. Bogo *et al*. [BRPB17] extend the registration in FAUST to 4D scans of moving people, and capture body shapes and poses over time, where both 2D texture and 3D geometry are exploited to align all scans with a standard template. Coregistration introduced in [HLRB12] defines an objective function to integrate the registering process with body model learning. A template is deformed with a new body model termed BlendSCAPE to address severe artifacts when rotating in coregistration. However, coregistration only learns the body shape of individual, so it is incapable of approximating all shapes.In contrast, SMPL in [LMR*15] is a skinned vertex-based model with better generalizations, which can widely capture correlations of various body shapes in different poses. It first registers a template mesh to each scan, and then uses principal component analysis (PCA) to obtain shape and pose blend shape.

Since the above registration methods pay more attention to body shape and pose, most of them tend to ignore hands. For example, hands in SMPL [LMR*15] are assumed to be open and rigid, lacking realism under some circumstances. Some researchers try to estimate bodies and hands at the same time [PCG*19, RTB17, JSS18] to solve this problem. [RTTP17] presented a hand model called MANO which can be combined with SMPL to capture bodies and hands simultaneously. [PCG*19] uses a same idea with [RTTP17] that combines SMPL with MANO, but additionally utilizes FLAM [LBB*17] model for facial expressions estimation.

Different from the mentioned methods, we first predict 2D hand keypoints with a convolutional neural network, and then perform registration by deforming a template hand mesh with linear blend skinning to fit the 2D hand keypoints in color images. The personalized neutral hand is then proposed to model the variations among different people. Combining with hand postures, we can finally obtain the personalized hand modeling results.

2.2. Personalized Hand Models

Personalized hand models [TSR*14, GLYT16, MBS*18, BTG*12, TTR*17, RTTP17, TBC*16] are widely applied in hand tracking, virtual reality and other fields. Personalization of hands means creating a precise template for a user. With development and popularization of RGB-D sensors, user-specific hand models can be fitted from RGB images [BTG*12, HVT*19, HHY*19, YLLY19, BDBT19]. Classical RGB-based methods focus on optimization for better model fitting. A smooth-surface model in [TBC*16] is utilized for non-linear optimization when iteratively fitting a hand model, and also reduces the computational cost. Although [TBC*16] can track hands efficiently and precisely for users, it still fails with heavy occlusions and complex backgrounds in the images. A generative model and discriminative points are coupled in [TBS*16] to offline capture moving rigid objects and articulated hands, dealing with the interaction and occlusions between the hand and object. [SJMS17] proposes multi-view bootstrapping to detect keypoints of hands, also dealing with occlusions. Noisy data in [SJMS17] caused by occlusions can be removed through triangulation iteratively, finally producing more reliable data. Thanks to the application of CNNs, personalized hand models can be learned in a more efficient and accurate way. A dense hand pose estimator (DHPE) [BKK19] is created by neural rendering, and is able to obtain the realistic 3D hand model from a single RGB image. MANO is also integrated into the training process to constrain the output 3D hand mesh in [BKK19]. Ge *et al*. [GRL*19] utilize Graph CNN for reconstructing a full 3D mesh containing both rich shape and pose information.

Apart from RGB images, depth data is also often exploited in personalized hand modeling. The online optimization algorithm in [TTR*17] focuses on estimating hands shape and pose from a sequence of depth frames, as well as the uncertainty of estimates. The uncertainty ensures the combination of estimates from each frame over time and improves the quality of the personalized hand model. Taylor *et al*. [TSR*14] propose a more efficient pipeline with automatic hand initialization, which also generates personalized hand models from depth sequences. The model proposed in [TCT*16] improves robustness of hands tracking as well as surface alignment for users via minimizing a cost function termed golden energy. Recently, learning-based methods are also developed in hand modeling from depth maps [MDB*19, ZXCZ20, MES18, CML18, YGS*18, HHY*19]. Mueller *et al*. [MDB*19] succeed in tracking two touching hands with only one depth camera by embedding a neural network into the energy minimization framework. Instead of directly estimating 3D coordinates from a depth map like common approaches do, V2V-PoseNet [CML18] first transforms the depth into voxelized grids and then predicts the possibility of voxels for each keypoint with a 3D CNN network. Obviously, using deep learning techniques in hand modeling is becoming a trend.

3. Method

An overview of the proposed method is shown in Fig. 3. We firstly collect hand postures with 15 synchronized and calibrated cameras (Sec. 3.1). For each color image of a single posture, we use a convolutional neural network to infer the 2D hand keypoints (Sec. 3.2), and then use a fitting-and-deforming strategy to obtain a hand template model (Sec. 3.3). Specifically, we first fit a template hand model with linear blend skinning to the estimated multi-view 2D hand keypoints (this step changes only the pose of the template hand model), and then utilize the embedded deformation [SSP07] to deform the template hand model to overlay the observed color images (this step may change the hand model geometry). Based on the deformed template models obtained from all the postures (i.e., postured 3D hand models), we then solve an optimization equation to obtain a personalized neutral hand model (Sec. 3.4). The final personalized neutral hand model can be achieved by using the personalized neutral hand model as the template hand model and iterating the above procedure. All the details are described in the following subsections.

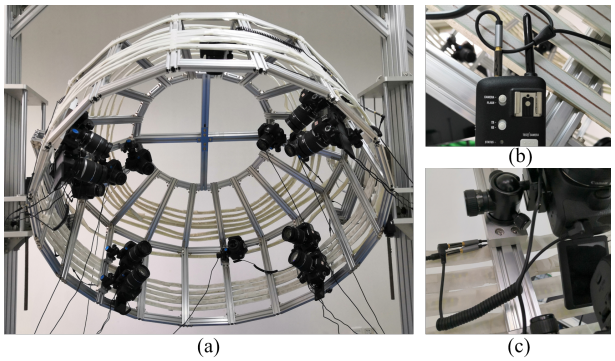


Figure 2: *Illustration of hardware system. (a) is a customized cage for hosting multiple cameras. (b) is a wireless receiver for performing the synchronization. (c) shows the linkage of* 2.5*mm audio cable for broadcasting shutter signals among all the cameras.*

3.1. Hardware System

To capture multiple hand postures, we build a 15-cameras system as shown in Fig. 2 (a). All the cameras are uniformly distributed in the customized holder, and we use one high speed wireless receiver as shown in Fig. 2 (b) to perform the synchronization of cameras. The shutter signal is linked by 2.5*mm* audio cables, which are shown in Fig. 2 (c), to broadcast among all the cameras. Our utilized audio-cable based synchronization strategy has the delay of less than 10*ms*.

We follow the calibration method proposed in [LHKP13] to calibrate all the cameras. It is noted that the distances between hand and each camera are very short in the cage and blurry images could influence the accuracy of camera calibration. Thus, we adopt a predefined image pattern attached to a cylinder, which is placed in the center of cage, to manually set the focus for each camera.

3.2. Hand 2D Keypoints Estimation

In recent years, with the rapid development of deep learning techniques, hand 2D pose estimation from color images has been discussed by a lot of researches [SJMS17]. Even so, achieving fast and accurate hand pose estimation results is still challenging due

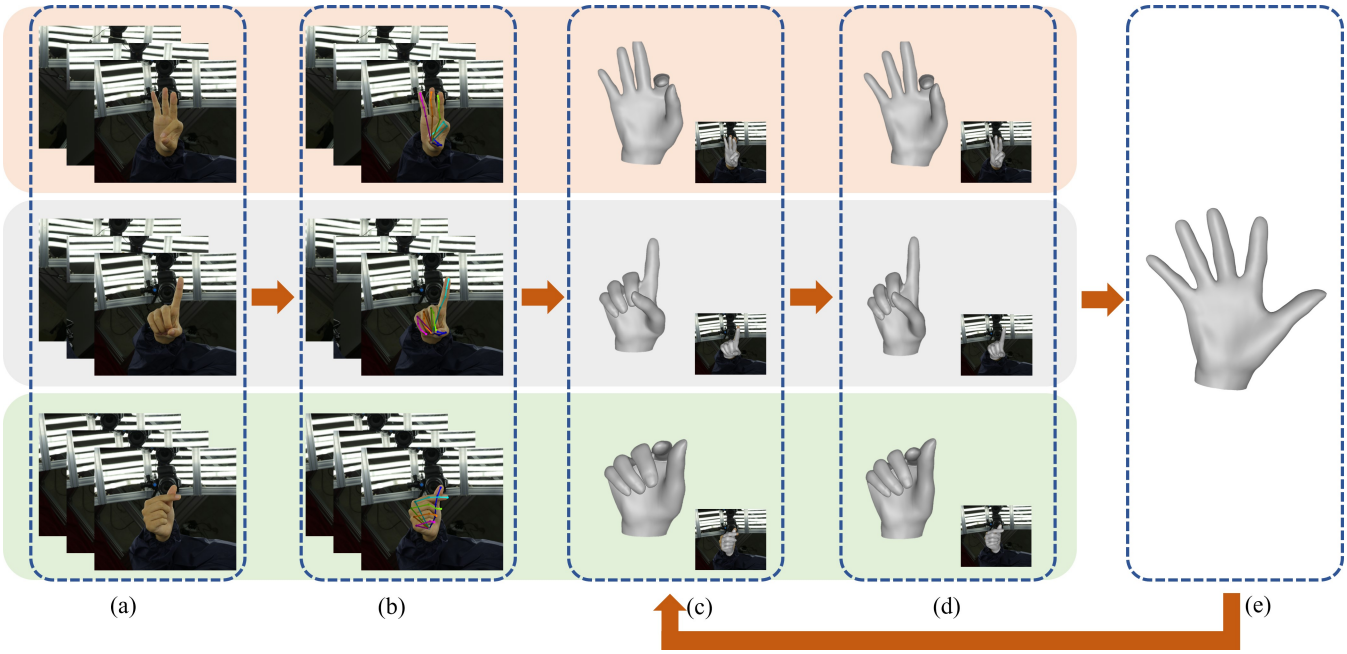


Figure 3: *Pipeline of the proposed method. (a) The input of our method are multi-view color images with different hand postures. (b) We then use the state-of-the-art 2D hand pose estimation method [WZP20] to obtain the 2D hand keypoints. (c) For each hand posture, a template hand is utilized to fit the detected hand 2D keypoints by Linear Blend Skinning. (d) After that, we use the embedded deformation [SSP07] to match the hand model with the observed 2D hand silhouettes. (e) Then we solve an optimization equation to obtain the personalized neutral hand model as shown in Sec. 3.4. We iteratively follow the steps (c) to (e) to refine the personalized neutral hand models. With the obtained personalized neutral hand models, we follow steps (c) and (d) to obtain the final personalized hand models.*

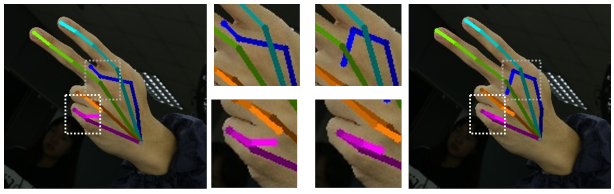


Figure 4: *Hand 2D keypoints estimation. (a) is the hand 2D keypoints estimated by network. (b) is optimized hand 2D keypoints by reprojection with multi-view information. The differences between two results are enlarged and shown in the middle.*

to flexible hand fingers movements, self-occlusions and appearance ambiguities in color images. We follow the state-of-the-art method [WZP20] to detect hand 2D poses. The hand 2D pose is represented by 21 2D keypoints, where each keypoint is described by a 2D Gaussian heatmap. An encoder-decoder network architecture is then utilized to estimate the 2D keypoints.

Specifically, we have 15-view synchronized color images for each hand pose. The network is fed with color images of each view to estimate the hand 2D keypoints separately. It is noted that not all views have accurate hand 2D keypoints estimation results. Our solution is to adopt multi-view information to boost the performance of 2D keypoints estimation results. We first compute hand 3D keypoints by optimizing the following equation with the camera parameters and the estimated 2D keypoints as

$$x = \underset{x}{\operatorname{argmax}} \sum_{v \in \mathcal{V}} \left|\left| [X_v]_\times \cdot K_v (R_v x + t_v) \right|\right|, \tag{1}$$

where x is the 3D keypoint, $\mathcal{V}$ is the set of visible cameras, K_v is the intrinsic parameter and $[R_v, t_v]$ is the extrinsic parameter for the v-th camera. $[X_v]_\times$ is the skew matrix of the homogenous coordinate of X_v, which describes the 2D coordinates of hand keypoints in the v-th camera.

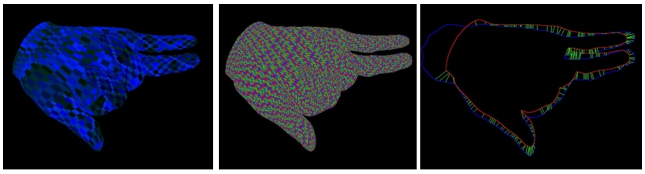
After we have obtained the 3D position for each keypoint by solving Eq. (1), we can reproject the 3D position into the 15 views again to obtain the updated 2D keypoints. This procedure can not only remove the random noise induced by network prediction, but also improve the confidence of keypoints estimation caused by invisible cameras.

3.3. Hand Postures Modeling

With the estimated multi-view hand 2D keypoints, we can deform a 3D hand skeleton and minimize the errors among projected skeleton joints and estimated 2D keypoints. The skeleton joint positions will be utilized in the following template deformation. We again follow Eq. (1) to build the optimization criteria, but replace x with a function of the hand pose, where the hand pose is represented

(a) (b) (c)

Figure 5: *Hand Postures Modeling. (a) is the initial template hand model. (b) is the deformed hand model by using linear blend skinning. (c) shows the results of embedded deformation.*



(a) (b) (c)

Figure 6: *Contour matching. (a) is the projected face index map. (b) is the projected barycentric map. (c) shows the contouring matching result, where blue line is original contour, red line is the projected contour and green line shows the computed correspondences.*

by a 27 dimensional vector θ, and the Degree of Freedoms (DoFs) of the hand pose are followed by the work [WMZ*13]. It is noted that a 3D hand skeleton can be treated as a kinematic chain and the position of all skeleton joints can be easily computed via kinematic chain rule [MLSS94]. We use Levenberg-marquardt method to solve the non-linear optimization. We found that explicit jacobian computation from 3D skeleton joints to the hand pose θ is important for the convergence of optimization.

After that, we utilize the technique of linear blend skinning to deform a template hand model M. Specifically, for the i-th vertex v_i in the mesh, the deformed new position $\tilde{v}_i$ is computed as

$$\tilde{v}_i = \sum_{J_k} \omega(v_i, J_k)[R_k(v_i - J_k) + J_k + t_k], \tag{2}$$

Here, J_k is the k-th skeleton joint position, $T_k = \{R_k, t_k\}$ is the rigid transformation of the k-th bone, which is only determined by the hand pose θ, the skinning weight $\omega(v_i, J_k)$ is computed by heat-based method [BP07], which measures the influence of the k-th bone to the i-th vertex.

Fig. 5 (b) shows the deformation result of linear blend skinning, which is projected onto image plane. From this figure, we can find that only rigid pose estimation may not fit the observed color images well. The reason comes from that the personalized template hand models are largely different from each other. This inspires us to perform the non-rigid deformation for the template hand mesh to fit the observed images.

We follow the representation in embedded deformation graph, where the non-rigid deformation is represented by affine transformations $\{A_k, t_k\}$ of K randomly selected vertices $\{x_k\}$ on the mesh. These vertices are treated as the nodes of graph. Again, for the i-th vertex v_i in the mesh, the deformed new position is computed as

$$\tilde{v}_i = \sum_{x_k \in \mathcal{N}(v_i)} \omega(v_i, x_k)[A_k(v_i - x_k) + x_k + t_k], \tag{3}$$

where $\mathcal{N}(v_i)$ is the neighbor nodes of the mesh vertex v_i. The neighborhood of all nodes in the deformation graph is defined in [SSP07]. $\omega(v_i, x_k)$ is the deformation weights of node x_k to v_i, which measures the influence of the node. The details of nodes extraction and weights computation are in [SSP07].

It is noted that Eq. (2) and Eq. (3) are very similar and the main difference is that embedded deformation does not require A_k to be a pure rotation. To compute the deformation in Eq. (2), we can easily obtain $T_K = \{R_k, t_k\}$ for the linear blend skinning by matching the 3D skeleton joints to all the observed 2D hand keypoints, which is mentioned earlier. However, it is not straightforward to compute $\{A_k, t_k\}$ in Eq. (3). We estimate $\{A_k, t_k\}$ by minimizing the following energy function

$$E = E_{fit} + \lambda_1 E_{rigid} + \lambda_2 E_{smooth}, \tag{4}$$

where

$$E_{fit} = \sum_{v} \sum_{v_i \in \mathcal{C}} \left|\left| [C_i]_{\times} \cdot K_v(R_v \tilde{v}_i + t_v) \right|\right|, \tag{5}$$

which forces the vertex v_i in the contour $\mathcal{C}$ to move to the corresponding contours C_i in the observed color images. $\mathcal{C}$ includes all the vertices that have corresponding contours in the color images. K_v is the intrinsic parameter and $[R_v, t_v]$ is the extrinsic parameter for the v-th camera.

There are two main issues for constructing the fit term in Eq. (5), one is how to find the matching 2D pixels in the image plane. We compute the 2D normal n for all the contour points in the image plane, and search the minimal distances when their normal dot

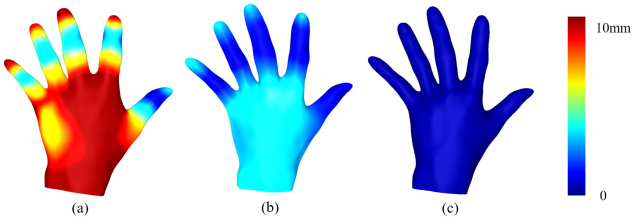


Figure 7: *Iterative neutral hand modeling results. (a) is the re-scaled template hand. (b) is the* 1*st hand modeling result and (c) is* 2*nd hand modeling result.*

product is larger than a threshold τ. In our current implementation, $\tau = 0.9$. The other issue is how to retrieve the 3D positions of 2D image coordinates. We adopt 2-pass rendering based strategy to retrieve the 3D positions of 2D image coordinates. Specifically, we first colorize the mesh triangles with its face index, for example, the *RGB* color value of the k-th triangle face is encoded as $R = ((k+1) \gg 16)\&0xFF$, $G = ((k+1) \gg 8)\&0xFF$, and $B = (k+1)\&0xFF$. Fig. 6 (a) shows the rendering result. Then, we set the triangle vertex color for each face as $(1,0,0),(0,1,0)$ and $(0,0,1)$, and re-render the mesh onto image plane as shown in Fig. 6 (b). After that, we could compute the barycentric coordinates for each face via reading the color information. With the face index and barycentric coordinate, we can directly obtain the corresponding 3D positions of 2D pixels.

E_{rigid} is the term to restrict the affine transformation to be as rigid as possible, which is the same in [SSP07] and is formulated as

$$E_{rigid} = \sum_i \Big((\mathbf{a}_{i,1}^T \mathbf{a}_{i,2})^2 + (\mathbf{a}_{i,1}^T \mathbf{a}_{i,3})^2 + (\mathbf{a}_{i,2}^T \mathbf{a}_{i,3})^2 + (1 - \mathbf{a}_{i,1}^T \mathbf{a}_{i,1})^2 + (1 - \mathbf{a}_{i,2}^T \mathbf{a}_{i,2})^2 + (1 - \mathbf{a}_{i,3}^T \mathbf{a}_{i,3})^2 \Big),$$

where $\mathbf{a}_{i,1}, \mathbf{a}_{i,2}$ and $\mathbf{a}_{i,3}$ are the column vectors of A_i. E_{smooth} ensures the smoothness of the deformation mesh and is computed for all the nodes of the embedded deformation graph as

$$E_{smooth} = \sum_{v_i} \sum_{v_j \in \mathcal{N}(v_i)} \omega(v_j, v_i) \Big|\Big| A_i(v_j - v_i) + v_i + t_i - (v_j + t_j) \Big|\Big|. \tag{6}$$

The minimization of Eq. (4) is performed iteratively. We found that 5 iterations are enough for convergence. Set $\lambda_1 = 2.0$ and $\lambda_2 = 10.0$ in all of our implementations.

3.4. Optimization for Personalized Neutral Hand

The goal of this step is to estimate the personalized neutral hand model (as illustrated in Fig. 8(e)) from several hand postures. It is formulated as an optimization problem. We utilize the obtained postured 3D hand models (as illustrated in Fig. 8(a)-(c)) as well as the pose information as input and solve a linear equation to obtain the template neutral hand model.

As mentioned in the previous subsection, T_k is a rigid transformation and only determined by the pose information (i.e., θ), we denote $T_k = [R_k, t_k]$, where R_k is the rotation and t_k is the translation. Thus, we can denote the deformation for the i-th vertex as $R_i(\theta), T_i(\theta)$, which has the form

$$R_i(\theta) = \sum_{k=0}^{K} \omega_k^i R_k,$$
$$T_i(\theta) = \sum_{k=0}^{K} \omega_k^i t_k.$$

Thus, for the whole n vertices in the mesh, we could build a diagonal matrix $R(\theta)$ as

$$R(\theta) = \begin{bmatrix} R_0(\theta) & & & \\ & R_1(\theta) & & \\ & & \ddots & \\ & & & R_n(\theta) \end{bmatrix} \tag{7}$$

and pack all $T_i(\theta)$ as a column vector $T(\theta)$. Then, the linear blend skinning for mesh deformation can be represented as

$$\tilde{\mathbf{v}} = R(\theta)\mathbf{v} + T(\theta), \tag{8}$$

where $\mathbf{v}$ and $\tilde{\mathbf{v}}$ are the packed column vectors for all mesh vertices, respectively.

With the obtained several hand postures described in Sec. 3.3, we can solve an optimization cost to obtain the personalized template hand model by

$$\mathbf{v} = \arg\min \sum_j ||R(\theta_j)\mathbf{v} + T(\theta_j) - \tilde{\mathbf{v}}||. \tag{9}$$

Here, $R(\theta_j)$, $T(\theta_j)$ and $\tilde{\mathbf{v}}$ are known and computed in Sec. 3.3.

To guarantee the smoothness of personalized template hand models, we additionally introduce a Laplacian smooth term into Eq. (9) and the optimization cost is replaced by the form

$$\mathbf{v} = \arg\min \sum_j ||R(\theta_j)\mathbf{v} + T(\theta_j) - \tilde{\mathbf{v}}|| + ||L\mathbf{v}||. \tag{10}$$

It is noted that all $R(\theta_j)$ and L are sparse matrices, and the proposed optimization cost function can be efficiently solved by sparse linear solver for its normal equation, that is

$$\left(\sum_j R(\theta_j)^T R(\theta_j) + L^T L \right) \mathbf{v} = \sum_j R(\theta_j)^T \left(\tilde{\mathbf{v}} - T(\theta_j) \right), \tag{11}$$

where $R(\theta_j)^T$ and L^T are the transposes of $R(\theta_j)$ and L respectively.

With the optimized personalized neural hand, we re-estimate the hand postures as described in Sec. 3.3 and iteratively optimize the personalized neutral hand. Fig. 7 shows the iterative estimation results. We found that 2 or 3 iterations are enough for obtaining stable personalized neutral hand.

4. Experiment

We captured left and right hands of 35 people with 15 multi-view color cameras. For each hand, we collected 30 postures and some of them are shown in Fig. 9. It is noted that there are no existing

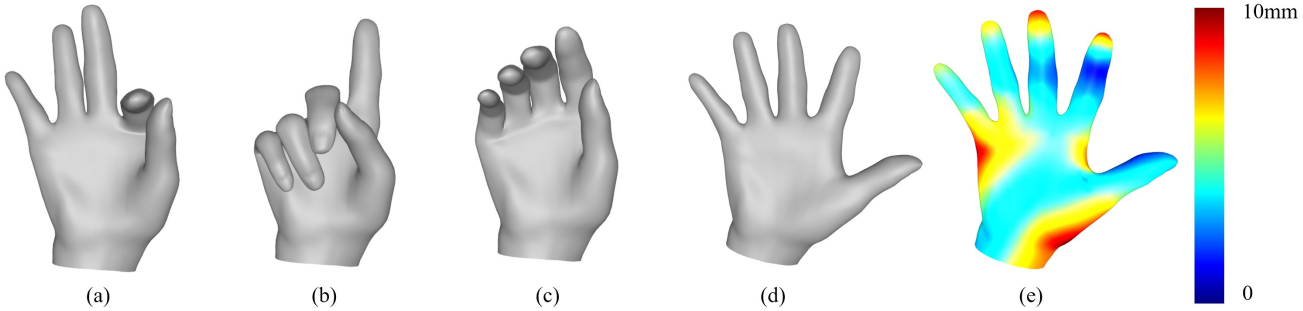


Figure 8: *Personalized hand optimization. (a)-(c) are three sampled hand postures. (d) is the template hand model and (e) shows the optimized personalized neutral hand model with the color-coded errors between the template and optimized hands.*

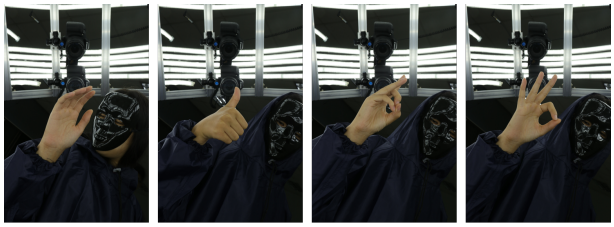


Figure 9: *Sampled images of different postures. The participate is requested to freely make different hand gestures and the hardware system captures the synchronized sequences for further processing.*

references for collecting the distinctive and meaningful hand postures. Thus, we requested a participator to freely make different hand gestures by referring sign languages [SLM06]. We then chose 30 distinctive hand postures as the target postures for data collection and asked the other participators to make them. We evaluate the proposed method on the captured data. Ablation studies of camera views and hand postures are also conducted. Running details of the proposed method are also presented in the following subsections.

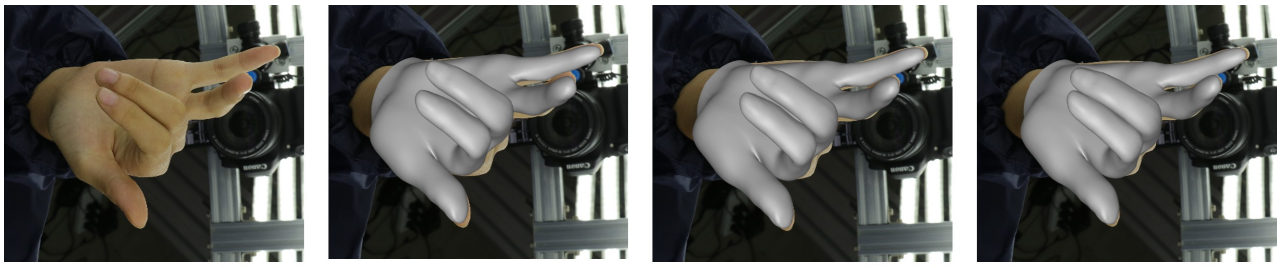
4.1. Datasets

Since our method relies on the deep learning based hand pose estimation, the data is important for the accurate and efficient network training. We first utilized the trained model given by SRHandNet [WZP20]. However, we found that the given model can not effectively estimate all the hand keypoints for our captured data. The reason is that the given model trained by **OneHand10K** is only available for visible hand keypoints, which is very different from our collected data. It may leave out several 3D hand keypoints with self-occlusions, and finally fail our multi-view hand postures modeling. To improve the network performance on samples with occlusions, two other datasets, including **STB** [ZJC*17] and **FreiHand** [ZCY*19] are then added into the training samples. Full hand keypoints are labeled in both of the two training datasets, which improves the robustness of the trained model towards self-occlusions. It is noted that the root keypoints of the hand poses in **STB** are at the palm centers, which are slightly different from the other datasets. To address the differences, we compute a mean vector $\mathbf{v}$ from 4 fingers to the palm center, and shift the original root position by adding $\mathbf{v}$ on the palm center to finally update the root keypoints.

4.2. Implementation Details

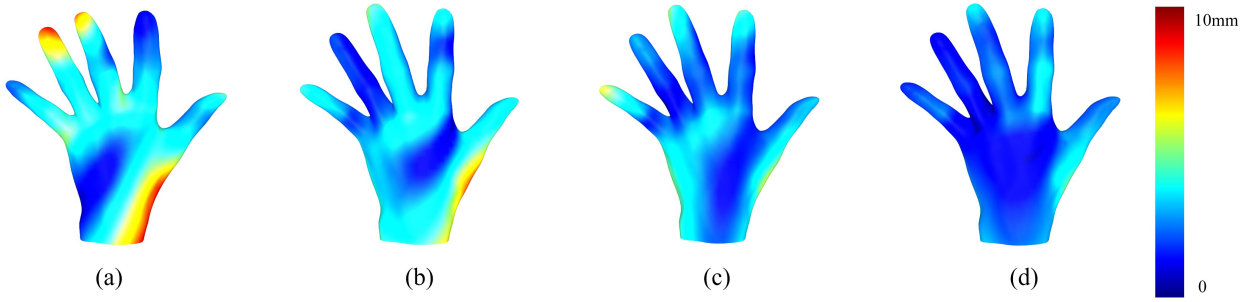
All the experiments were run on a desktop with one Intel 9900K CPU and 64GB RAMs. We trained the 2D hand pose estimation network with Pytorch 1.5 on one Nvidia GTX 2080Ti for 100 epochs, which costs about 5 days. **OneHand10K**, **STB** and **FreiHand** all contributed for the network training. We followed the network structure of SRHandNet [WZP20] except using the inception module [SVI*16] for extracting the features of input image, where it was resized into the resolution of 256×256 padding with zero values. Moreover, we used Leaky-Relu [MHN13] other than ReLu to improve the training efficiency since there are lots of zero values in the generated heatmaps. The Adam optimizer was utilized with a batch-size of 10 and the initial learning rate was set to 10^{-4}. The network inference costs about 20*ms* per image for 2D hand keypoints estimation, and 15 color images need to be processed for one hand posture.

Our employed template hand model has $6,829$ vertices and $13,552$ faces. As mentioned earlier, two-step optimization is used in our method, i.e., rigid hand pose estimation and non-rigid hand deformation with embedded deformation graph. Rigid hand pose estimation only relies on 2D hand keypoints, which are obtained by a neural network and refined by multi-view information boosting. We solved the linear optimization of rigid hand pose estimation by LDLT decomposition. As for the non-rigid deformation, hand masks and contours should be prepared for computation. We used background color subtraction to obtain the hand masks of captured color images. The contours of deformed template hand were computed by off-screen rendering. To accelerate the computation, we reduced the resolution of off-screen rendering to 600×400. The non-linear optimization of embedded deformation was solved by Ceres [AMO], where we used SPARSE_NORMAL_CHOLESKY decomposition. Finally, our method can run efficiently and accu-



(a) (b) (c) (d)

Figure 10: *Results of personalized hand modeling with different number of views. (a) is the novel view, which is not utilized in the optimization. (b), (c) and (d) show the overlaid results of the hand modeling result with* 5, 10 15 *views respectively.*



(a) (b) (c) (d)

Figure 11: *Results of personalized neutral hand modeling with different number of postures.* 1, 5, 15 *and* 25 *postures are utilized for optimization and the results are visualized (from left to right).*

rately. Typically, 2D keypoints refinement costs about 5*ms*, background color subtraction costs about 10*ms*, rigid hand pose estimation costs about 1*s* and non-rigid deformation with embedded deformation costs about 5*s*, where we run 4 iterations. After all postures are processed, we optimized Eq. (11) to obtain the personalized neural hand models for about 30*ms*.

4.3. Ablation Study

To demonstrate the performance of our proposed method, we evaluate the method by performing ablation studies including the number of views and the number of postures.

4.3.1. Number of views

We investigate the quality of personalized hand modeling with different number of views and results are shown in Fig. 10. From left to right, we show the personalized hand modeling results with 5, 10 and 15 camera views respectively. The cameras are sampled uniformly. As shown in Fig. 10, the reconstruction results become better when the number of camera views increase. To quantitatively compare the results, we have synthesized the hand postures and rendered the hand masks onto image plane, then we compare the computed hand models with the ground-truth hand meshes and the results are shown in Tab. 1. It can be observed that the estimation errors gradually decrease as the number of camera views increases. More number of camera views could provide more information for the personalized hand modeling, thus improve the accuracy of hand modeling, especially for the self-occlusion scenarios.

Table 1: *Quantitative comparisons with different number of views.*

Number of views	5 views	10 views	15 views
Mean (mm)	12.6	8.7	3.3
Std.	6.5	4.3	2.1

4.3.2. Number of postures

We have also investigated the influence of different numbers of hand postures to model the personalized hands and results are shown in Fig. 11. From left to right, we show the personalized hand modeling results with 1, 5, 15 and 25 hand postures respectively. As shown in the figure, we could find that the modeling results coincides with the number of hand postures. Since there is no reference of hand postures, it is a key challenge to choose representative hand postures. To find the representative hand poses, we first compute the principal component of all the 20-dimensional hand postures from captured video sequences (without global rotation, global translation and global scale) and project all the hand poses into the first principal component. Then we uniformly sample the hand postures from the projected values to obtain the representative hand postures. We can see that the variation of hand postures could improve the performance of hand modeling.

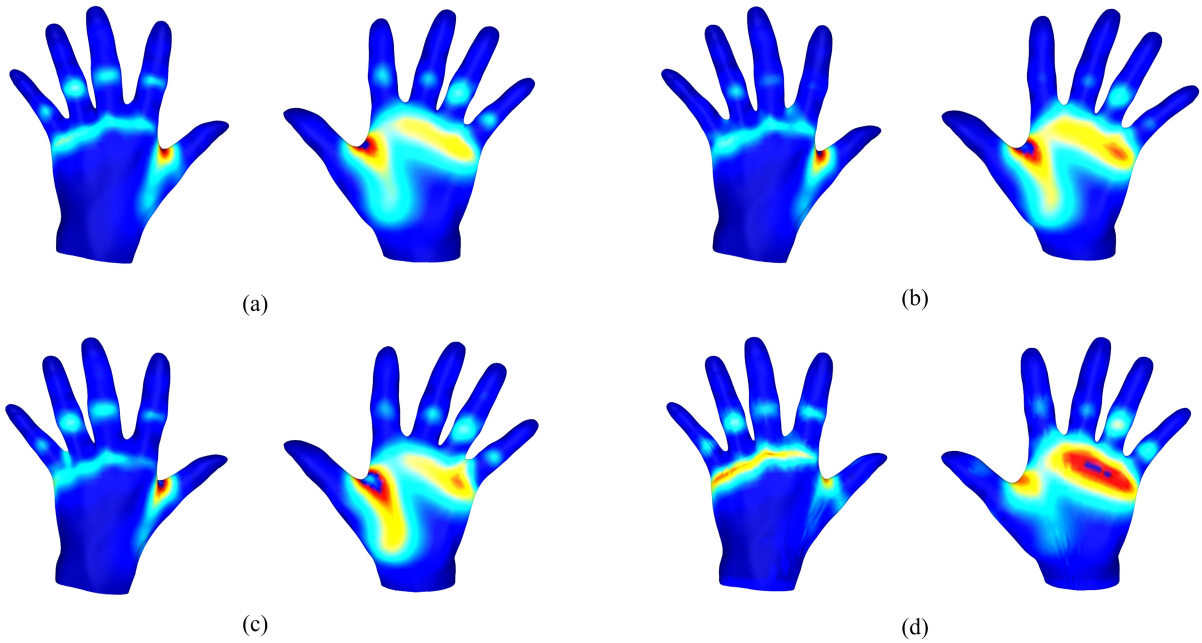


Figure 12: *Comparisons against neutral hand modeling from inverting parameters of hand postures. 4 different personal results are visualized in (a), (b), (c) and (d). The large errors come from palm and skeleton joints, where average of inverting the parameters of hand postures probably introduce errors caused by linear blend skinning.*

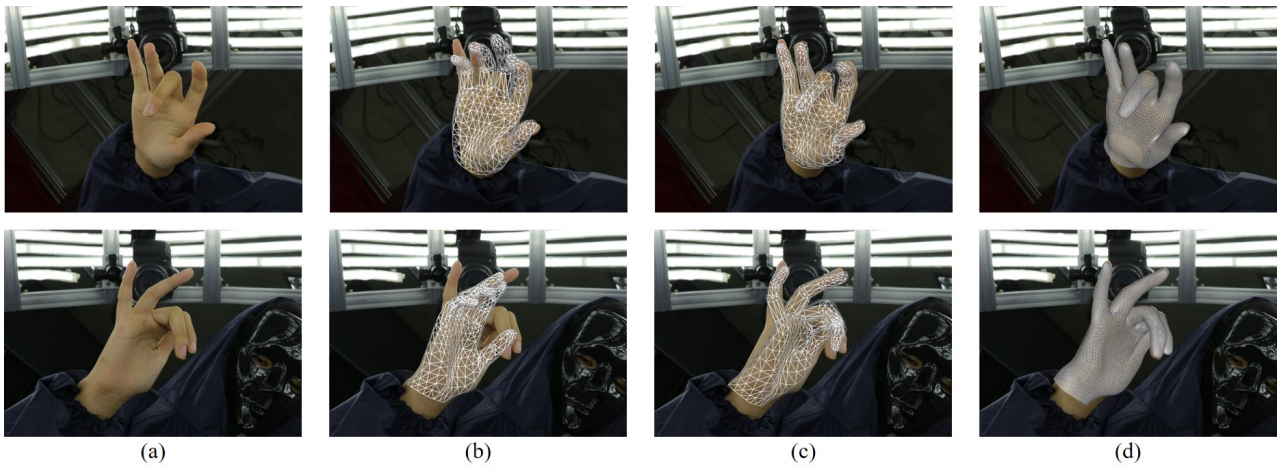


Figure 13: *Comparisons against MANO parameters estimation based methods. (a) is the original image; (b) is the result of MANO parameters estimation from neural network [ZLM*19]; (c) shows the result of MANO parameters estimation via non-linear optimization [RTB17] and (d) is our result. Our method can obtain high quality and better results.*

4.4. Evaluation and Comparison

To obtain personalized neutral hand models, there are several strategies to follow. We compare with two straightforward routines and evaluate the performance of our proposed method. The first one is, instead of directly matching 2D image cues, reconstructing point clouds from multi-view color images at first by the state-of-the-art multi-view reconstruction methods. Then, we perform the non-rigid ICP [LAGP09] by deforming the template hand mesh with the estimated point clouds to obtain different hand postures. We try the 3D reconstruction method of COLMAP [SZPF16] to estimate point clouds, which are shown in the 2nd and 3rd columns in Fig. 14. With the estimated point clouds, the results of deformed different hand postures are shown in the 4th column in Fig. 14. Compared with our method, it can be seen that the registered hand

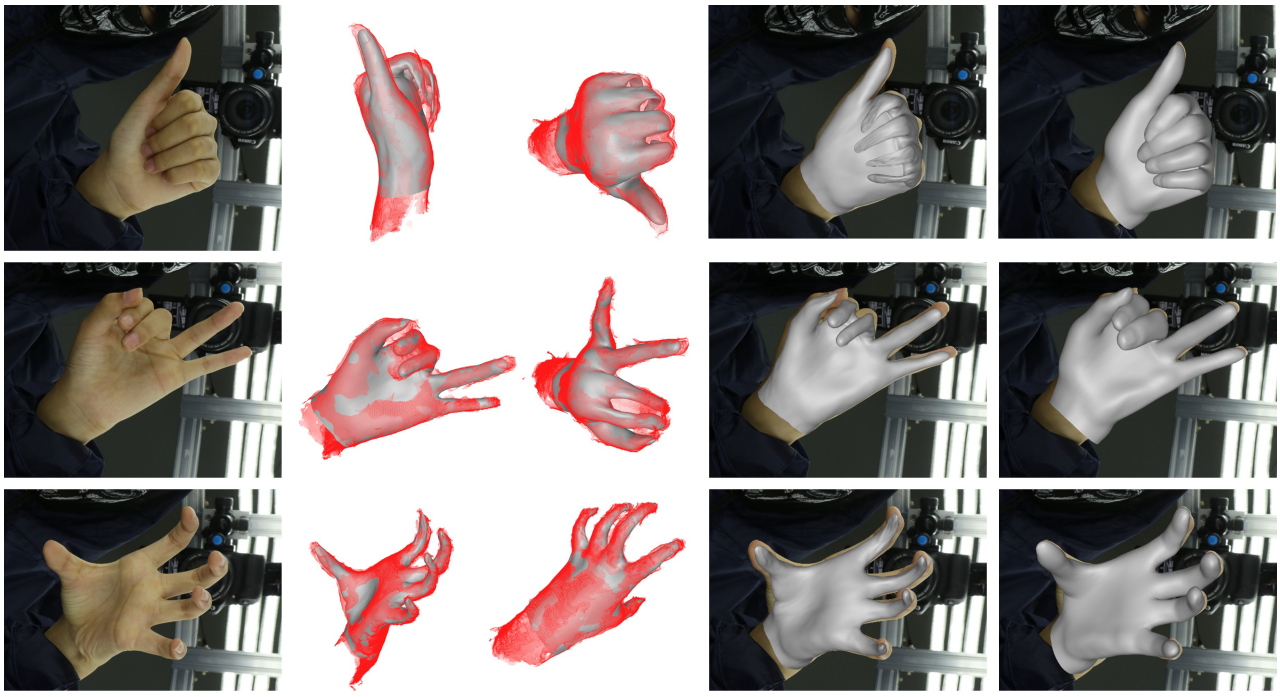


Figure 14: *Comparisons against hand modeling results from point clouds. (a) shows the testing color images; (b) shows point clouds (red) and fitted hand models; (c) shows the overlaid results of hand modeling from point clouds; (d) shows the results of the proposed method. Note that the registered hand meshes from point clouds are always shrunk, which produces bad results for personalized hand modeling.*

meshes from point clouds are always shrunk, which can produce bad results for personalized hand modeling.

Different from optimizing the hand postures from several hand postures, another straightforward strategy is to invert the parameters of hand postures and deform different hand postures to obtain the personalized neural hand models. With all the different hand postures, we can compute the average personalized hand models from all the inverted hand models. Fig. 12 shows the comparison results. From this figure, we can find that this result can also perform bad for personalized hand modeling. In order to quantitatively compare the results, we have also synthesized different hand postures and compute the error of two different strategies. We find that the proposed method can obtain the accuracy of 3.36*mm*, while inverting the hand pose parameters obtains the accuracy of 8.17*mm*, which also demonstrates the effectiveness of our method.

In addition, we also compare our method with two state-of-the-art strategies, including MANO parameters optimization [RTB17] and regressing MANO parameters [ZLM*19] via neural network. We utilize 2D hand keypoints from the 15 cameras to perform the MANO parameters optimization. As for the neural network estimation, we finetune the network parameters by adding the captured color images. Fig. 13 shows the comparison results. From this figure, we can find that MANO based methods can only obtain low personalized hand modeling results due to the fact that the original MANO model has only 778 vertices. And the meshes cannot be aligned with the images well. Because of the misaligned hand meshes, accurate hand poses are also unable to be achieved. However, our method can obtain high quality and better results compared with the above mentioned methods.

5. Conclusion

In this paper, we propose a new approach to reconstruct high-fidelity personalized hand models from multi-view color images of multiple postures. For each color image of a posture, 2D hand keypoints are estimated by a convolutional neural network and the estimated keypoints with estimation errors are further collected by using multi-view information. After that, a template hand model is fit to the 2D hand keypoints estimated from multiple views with linear blend skinning, and it is further deformed with embedded deformation to fit the observed color images from the posture. Next, an optimization strategy is leveraged to obtain a personalized neural hand model from the hand models constructed from the images of each postures. The personalized neural hand model can also be improved by iterating the above procedure. Experimental results show our approach can achieve much better results with much less computational time, compared to the state-of-the-art methods. All the hand models are publicly available on our website, and we hope the proposed method can enlarge the scale of personalized 3D hand datasets and thus improve the accuracy of parametric hand models, such as MANO [RTB17], in the future.

Acknowledgements

We would like to thank Zhenyi Zhao and Tianyao Wang for helping to capture the majorities of multiview hand images. This work was supported in part by the National Natural Science Foundation of China (No. 62076061, 61806054), in part by the Natural Science Foundation of Jiangsu Province (No. BK20180355), in part by the Shenzhen Fundamental Research Program (No. JCYJ20180306174459972) and "Zhishan Young Scholar" Program of Southeast University.

References

[AMO] AGARWAL S., MIERLE K., OTHERS: Ceres solver. http://ceres-solver.org. Accessed August 20, 2020.

[BBLR15] BOGO F., BLACK M. J., LOPER M., ROMERO J.: Detailed full-body reconstructions of moving people from monocular rgb-d sequences. 2300–2308.

[BDBT19] BOUKHAYMA A., DE BEM R., TORR P. H. S.: 3d hand shape and pose from images in the wild. 10843–10852.

[BKK19] BAEK S., KIM K. I., KIM T.: Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. 1067–1076.

[BKL*16] BOGO F., KANAZAWA A., LASSNER C., GEHLER P. V., ROMERO J., BLACK M. J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. 561–578.

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *tog 26*, 3 (2007).

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: Faust: Dataset and evaluation for 3d mesh registration. 3794–3801.

[BRPB17] BOGO F., ROMERO J., PONSMOLL G., BLACK M. J.: Dynamic faust: Registering human bodies in motion. 5573–5582.

[BTG*12] BALLAN L., TANEJA A., GALL J., VAN GOOL L., POLLEFEYS M.: Motion capture of hands in action using discriminative salient points. 640–653.

[CK15] CHEN Q., KOLTUN V.: Robust nonrigid registration by convex optimization. 2039–2047.

[CML18] CHANG J. Y., MOON G., LEE K. M.: V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. 5079–5088.

[CZ11] CHANG W. S. C., ZWICKER M.: Global registration of dynamic range scans for articulated model reconstruction. *ACM Transactions on Graphics 30*, 3 (2011), 26.

[GLYT16] GE L., LIANG H., YUAN J., THALMANN D.: Robust 3d hand pose estimation in single depth images: From single-view cnn to multi-view cnns. 3593–3601.

[GRL*19] GE L., REN Z., LI Y., XUE Z., WANG Y., CAI J., YUAN J.: 3d hand shape and pose estimation from a single rgb image. 10833–10842.

[HHY*19] HE Y., HU W., YANG S., QU X., WAN P., GUO Z.: Graph-posegan: 3d hand pose estimation from a monocular rgb image via adversarial learning on graphs. *arXiv: Computer Vision and Pattern Recognition* (2019).

[HLRB12] HIRSHBERG D. A., LOPER M., RACHLIN E., BLACK M. J.: Coregistration: simultaneous alignment and modeling of articulated 3d shape. 242–255.

[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. 11807–11816.

[JBPS11] JACOBSON A., BARAN I., POPOVIC J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4 (2011), 78.

[JSS18] JOO H., SIMON T., SHEIKH Y.: Total capture: A 3d deformation model for tracking faces, hands, and bodies. 8320–8329.

[LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. 175.

[LBB*17] LI T., BOLKART T., BLACK M. J., LI H., ROMERO J.: Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics 36*, 6 (2017), 194.

[LHKP13] LI B., HENG L., KOSER K., POLLEFEYS M.: A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *IROS* (2013), IEEE.

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONSMOLL G., BLACK M. J.: Smpl: a skinned multi-person linear model. 248.

[MBS*18] MUELLER F., BERNARD F., SOTNYCHENKO O., MEHTA D., SRIDHAR S., CASAS D., THEOBALT C.: Ganerated hands for real-time 3d hand tracking from monocular rgb. 49–59.

[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 Transactions on Graphics 38*, 4 (2019), 11761–11766.

[MES18] MALIK J., ELHAYEK A., STRICKER D.: Structure-aware 3d hand pose regression from a single depth image. 3–17.

[MHN13] MAAS A. L., HANNUN A. Y., NG A. Y.: Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (2013), vol. 30, p. 3.

[MLSS94] MURRAY R. M., LI Z., SASTRY S. S., SASTRY S. S.: *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[PCG*19] PAVLAKOS G., CHOUTAS V., GHORBANI N., BOLKART T., OSMAN A. A., TZIONAS D., BLACK M. J.: Expressive body capture: 3d hands, face, and body from a single image. 10975–10985.

[RTB17] ROMERO J., TZIONAS D., BLACK M. J.: Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics 36*, 6 (2017), 245.

[RTTP17] REMELLI E., TKACH A., TAGLIASACCHI A., PAULY M.: Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. 2554–2562.

[SJMS17] SIMON T., JOO H., MATTHEWS I., SHEIKH Y.: Hand keypoint detection in single images using multiview bootstrapping. 4645–4653.

[SKR*15] SHARP T., KESKIN C., ROBERTSON D., TAYLOR J., SHOTTON J., KIM D., RHEMANN C., LEICHTER I., VINNIKOV A., WEI Y., ET AL.: Accurate, robust, and flexible real-time hand tracking. 3633–3642.

[SLM06] SANDLER W., LILLO-MARTIN D.: *Sign language and linguistic universals*. Cambridge University Press, 2006.

[SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. *tog 26*, 3 (2007).

[SVI*16] SZEGEDY C., VANHOUCKE V., IOFFE S., SHLENS J., WOJNA Z.: Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2818–2826.

[SZPF16] SCHÖNBERGER J. L., ZHENG E., POLLEFEYS M., FRAHM J.-M.: Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)* (2016).

[TBC*16] TAYLOR J., BORDEAUX L., CASHMAN T. J., CORISH B., KESKIN C., SHARP T., SOTO E., SWEENEY D., VALENTIN J., LUFF B., ET AL.: Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. 143.

[TBS*16] TZIONAS D., BALLAN L., SRIKANTHA A., APONTE P., POLLEFEYS M., GALL J.: Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision 118*, 2 (2016), 172–193.

[TCL*13] TAM G. K. L., CHENG Z., LAI Y., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X., ROSIN P. L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics 19*, 7 (2013), 1199–1217.

[TCT*16] TAN D. J., CASHMAN T. J., TAYLOR J., FITZGIBBON A., TARLOW D., KHAMIS S., IZADI S., SHOTTON J.: Fits like a glove: Rapid and reliable hand shape personalization. 5610–5619.

[TSR*14] TAYLOR J., STEBBING R., RAMAKRISHNA V., KESKIN C., SHOTTON J., IZADI S., HERTZMANN A., FITZGIBBON A.: User-specific hand modeling from monocular depth sequences. 644–651.

[TST*15] TAGLIASACCHI A., SCHRODER M., TKACH A., BOUAZIZ S., BOTSCH M., PAULY M.: Robust articulated-icp for real-time hand tracking. 101–114.

[TTR*17] TKACH A., TAGLIASACCHI A., REMELLI E., PAULY M., FITZGIBBON A.: Online generative model personalization for hand tracking. 243.

[VPAS19] VENKAT A., PATEL C., AGRAWAL Y., SHARMA A.: Human-meshnet: Polygonal mesh recovery of humans.

[WMB19] WANG B., MATCUK G., BARBIČ J.: Hand modeling and simulation using stabilized magnetic resonance imaging. *ACM Trans. Graph. 38*, 4 (July 2019).

[WMZ*13] WANG Y., MIN J., ZHANG J., LIU Y., XU F., DAI Q., CHAI J.: Video-based hand manipulation capture through composite motion control. *tog 32*, 4 (2013).

[WWV*16] WANG R., WEI L., VOUGA E., HUANG Q., CEYLAN D., MEDIONI G., LI H.: Capturing dynamic textured surfaces of moving targets. 271–288.

[WZP20] WANG Y., ZHANG B., PENG C.: Srhandnet: Real-time 2d hand pose estimation with simultaneous region localization. *IEEE Transactions on Image Processing 29*, 1 (2020), 2977 – 2986.

[YGS*18] YUAN S., GARCIAHERNANDO G., STENGER B., MOON G., CHANG J. Y., LEE K. M., MOLCHANOV P., KAUTZ J., HONARI S., GE L., ET AL.: Depth-based 3d hand pose estimation: From current achievements to future goals. 2636–2645.

[YLLY19] YANG L., LI S., LEE D., YAO A.: Aligning latent spaces for 3d hand pose estimation. 2335–2343.

[ZCY*19] ZIMMERMANN C., CEYLAN D., YANG J., RUSSELL B., ARGUS M., BROX T.: Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 813–822.

[Zha94] ZHANG Z.: Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision 13*, 2 (1994), 119–152.

[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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 0–0.

[ZJC*17] ZHANG J., JIAO J., CHEN M., QU L., XU X., YANG Q.: A hand pose tracking benchmark from stereo matching. In *2017 IEEE International Conference on Image Processing (ICIP)* (2017), IEEE, pp. 982–986.

[ZLM*19] ZHANG X., LI Q., MO H., ZHANG W., ZHENG W.: End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 2354–2364.

[ZXCZ20] ZHANG Z., XIE S., CHEN M., ZHU H.: Handaugment: A simple data augmentation method for depth-based 3d hand pose estimation. *arXiv: Computer Vision and Pattern Recognition* (2020).