# HTML: A Parametric Hand Texture Model for 3D Hand Reconstruction and Personalization

Neng Qian[1,2], Jiayi Wang[1], Franziska Mueller[1(✉)], Florian Bernard[1,3], Vladislav Golyanik[1], and Christian Theobalt[1]

[1] Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
{nqian,jwang,frmueller,fbernard,golyanik,theobalt}@mpi-inf.mpg.de
[2] RWTH Aachen University, Aachen, Germany
[3] Technical University of Munich, München, Germany

**Abstract.** 3D hand reconstruction from images is a widely-studied problem in computer vision and graphics, and has a particularly high relevance for virtual and augmented reality. Although several 3D hand reconstruction approaches leverage hand models as a strong prior to resolve ambiguities and achieve more robust results, most existing models account only for the hand shape and poses and do not model the texture. To fill this gap, in this work we present HTML, the first parametric texture model of human hands. Our model spans several dimensions of hand appearance variability (*e.g.*, related to gender, ethnicity, or age) and only requires a commodity camera for data acquisition. Experimentally, we demonstrate that our appearance model can be used to tackle a range of challenging problems such as 3D hand reconstruction from a single monocular image. Furthermore, our appearance model can be used to define a neural rendering layer that enables training with a self-supervised photometric loss. We make our model publicly available.

**Keywords:** Hand texture model · Appearance modeling · Hand tracking · 3D hand reconstruction

## 1 Introduction

Hands are one of the most natural ways for humans to interact with their environment. As interest in virtual and augmented reality grows, so does the need for reconstructing a user's hands to enable intuitive and immersive interactions with the virtual environment. Ideally, this reconstruction contains accurate hand

https://handtracker.mpi-inf.mpg.de/projects/HandTextureModel/.

**Electronic supplementary material** The online version of this chapter (https://doi.org/10.1007/978-3-030-58621-8_4) contains supplementary material, which is available to authorized users.

© Springer Nature Switzerland AG 2020
A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12356, pp. 54–71, 2020.
https://doi.org/10.1007/978-3-030-58621-8_4

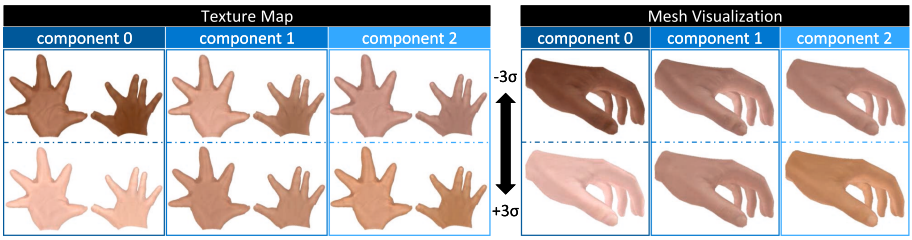


**Fig. 1.** We present the first parametric hand texture model. Our model successfully captures appearance variations from different gender, age, and ethnicity.

shape, pose, and appearance. However, it is a challenging task to capture a user's hands from just images due to the complexity of hand interactions and self-occlusion. In recent years, there has been significant progress in hand pose estimation from monocular depth [1,8,12,25,30,53,54] and RGB [7,22,46,55,57] images. Although most of these works estimate only joint positions, a few recent works attempt to reconstruct the hand geometry as well [2,6,26,27,56].

Despite these recent advances, there is little work that addresses the reconstruction of hand appearance. However, hand appearance personalization is important for increasing immersion and the sense of "body-ownership" in VR applications [19], and for improved tracking and pose estimation through analysis-by-synthesis approaches. Without a personalized appearance model, existing pose estimation methods must use much coarser hand silhouettes [2,6,56] as an approximation of appearance. One approach to obtain a personalized hand texture is to project the tracked geometry to the RGB image and copy the observed color to the texture map [23]. However, only a partial appearance of the observed hand parts can be recovered with this method and tracking errors can lead to unnatural appearances. In addition, without explicit lighting estimation, lighting effects will be baked into the results of these projection-based methods.

To address this gap, we present *HTML*, the first data-driven parametric **H**and **T**exture **M**ode**L** (see Fig. 1). We captured a large variety of hands and aligned the scans in order to enable principal component analysis (PCA) and build a textured parametric hand model. PCA compresses the variations of natural hand appearances to a low dimensional appearance basis, thus enabling a more robust appearance fitting. Our model can additionally produce plausible appearance of the entire hand from fitting to partial observations from a single RGB image. Our main **contributions** can be summarized as follows:

- We introduce a novel parametric model of hand texture, HTML, that we make publicly available. Our model is based on a dataset of high-resolution hand scans of 51 subjects with variety in gender, age, and ethnicity.
- We register our scans to the popular MANO hand model [39] in order to create a statistical hand appearance model that is also compatible with MANO.

- We demonstrate that our new parametric texture model allows to obtain a personalized 3D hand mesh from a single RGB image of the user's hand.
- We present a proof-of-concept neural network layer which uses the MANO shape and pose model in combination with our proposed texture model in an analysis-by-synthesis fashion. It enables a self-supervised photometric loss, directly comparing the textured rendered hand model to the input image.

## 2 Related Work

The use of detailed, yet computationally efficient, hand models for hand tracking applications is well studied [31,35,41,48]. Nevertheless, many such methods require time-consuming expert adjustments to personalize the model to a user's hand, making them difficult to deploy to the end-user. Therefore, we focus our review to methods that can automatically generate personalized articulated hand models from images. However, we will see that almost all these methods exclusively consider shape personalization and do not include texture or appearance.

*Modeling Hand Geometry.* Two types of personalizable hand models exist in the literature, *i.e.*, heuristic parameterizations that directly move and scale the geometric primitives of the models [23,37,47,51,52], and data-driven statistical parameterizations that model the covariance of hand geometry [20,39]. Although heuristic approaches are expressive, infeasible hand-shape configurations can arise when fitting such models to single images due to ambiguities between shape and pose. Thus, existing approaches must perform the personalization offline over a set of depth images [37,47,51], or design additional heuristic constraints [52] to resolve these ambiguities. On the other hand, data-driven parameterizations [20] provide a low-dimensional shape representation and natural priors on hand poses. The recent MANO model [39] additionally provides learned data-driven pose-dependent shape corrections to the geometry to avoid artifacts in posing a hand model through *linear blend skinning* (LBS). This model has been applied in many recent hand pose estimation methods [2,6,16,56] and has been used to annotate hand pose estimation benchmarks [15,16,58].

Nonetheless, and despite the popularity of the MANO model of hand geometry, there exists no data-driven parametric texture model for providing realistic appearance. As such, in this work we present for the first time a hand appearance model that is fully compatible with MANO. Although MANO has a rather low-resolution mesh (778 vertices), our appearance model is defined in the texture space so that a much higher texture resolution is available.

*Modeling Appearance.* With a few exceptions [23,24], the previously mentioned works do not model hand texture. The works of de La Gorce *et al.* [23,24] incorporate heuristic texture personalization for hand-tracking using an analysis-by-synthesis approach. Their approach obtains only a partial estimate of the hand texture using the current pose estimate, and relies on a smoothness prior to transfer color to unobserved parts by a diffusion process on a per-frame basis.

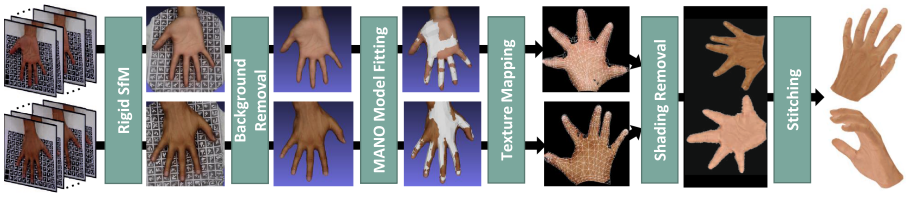


**Fig. 2.** Overview of our hand texture acquisition pipeline. We run rigid structure from motion (SfM) on a set of input images to obtain a scanned mesh for back and palm side of the hand, respectively. After removing background vertices, we fit the MANO template mesh to extract the texture from the scan. We remove lighting effects and seamlessly stitch the front and back texture, resulting in a complete texture for the captured hand (visualized on the 3D hand mesh from 2 virtual views on the right).

Romero *et al.* [39] provide the raw RGB scans used to register the MANO model, but they contain strong lighting effects like shadows and over-exposed regions. Hence, it is not possible to recover accurate appearance from these scans as we show in the supplementary document. Despite the lack of a parametric hand texture model, the benefits of having such a model can be readily seen in face modeling literature. For example, 3D morphable face models (3DMM) [5,9,13,18,33] provide parametric geometry and appearance models for faces that have been used to drive research in many recent works in diverse applications [10]. For example, these 3DMMs were used within analysis-by-synthesis frameworks for RGB tracking [38,50], and as unsupervised loss for learning-based methods [49]. Our proposed parametric hand appearance model HTML has the potential to drive similar advances in the hand pose estimation and modeling community.

## 3 Textured Parametric Hand Model

Our hand texture acquisition pipeline is summarized in Fig. 2. First, we record two image sequences observing the palm side and the back side of the hand, respectively. Subsequently, we run rigid structure from motion (SfM) [3,40] to obtain a 3D reconstruction of the observed hand side (Sect. 3.1). Next, we remove the scene background, and register both (partial) hand scans to the MANO model [39] based on nonlinear optimization. Afterwards, the texture of the partial hand scans is mapped to the registered mesh. We then remove shading effects from the textures and stitch them to obtain a complete hand texture (Sect. 3.2). The parametric texture model in subsequently generated using PCA (Sect. 3.3).

### 3.1 Data Acquisition

In total, we captured 51 subjects with varying gender, age, and ethnicity (see Fig. 3). To minimize hand motion during scanning, we record the palm side and backside of the hand separately, so that the subjects can rest their hand on a

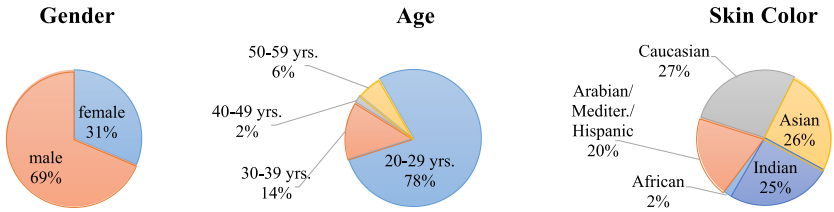


**Fig. 3.** Distribution of age, gender, and skin color for our 51 captured subjects. We use the Goldman world classification scale [42] for classifying skin color.

flat surface. As such, for each subject we obtain four scans, *i.e.*, back and palm sides for both left and right hands. The scanning takes ∼90 s for one hand side, so that the total scanning time of ∼6 min is required per person.

To obtain 3D hand scans, we use SONY's 3DCreator App [44]. The 3D reconstruction pipeline includes three stages, *i.e.*, initial anchor point extraction, simultaneous localization and mapping (SLAM) with sparse points [21], and online dense 3D reconstruction (sculpting) [45]. The output is a textured high-resolution surface mesh (of one hand side as well as the background), which contains $\sim 6.2k$ vertices and $\sim 11k$ triangles in the hand area on average. By design, our hand texture model is built for the right hand. For model creation, we mirror the left hand meshes, so that we use a total of 102 "right" hands for modeling. We note that by mirroring we can also use the texture model of "right" hand for the left hand. In the following, we will abstract away this technical detail and describe our texture modeling approach for a single hand.

### 3.2 Data Canonicalization

To learn the texture variations in a data-driven manner it is crucial that the acquired 3D scans are brought into a common representation. Due to the popularity and the wide use of the MANO model of hand geometry, we decided to build the hand texture in the MANO space. This has the advantage that existing hand reconstruction and tracking frameworks that are based on MANO, such as [6,16,29], can be directly extended to also incorporate hand texture. We point out that our texture model can also be used with other models by defining the respective UV mapping. Our data canonicalization comprises several consecutive stages, *i.e.*, *background removal*, *MANO model fitting*, *texture mapping*, *shading removal*, and *seamless stitching*, which we describe next.

*Background Removal.* For each hand we have reconstructed two textured meshes, one that shows the hand palm-down on a flat surface, and one that shows the hand palm-up on a flat surface (cf. Sect. 3.1). In both cases, the background, *i.e.*, the flat surface that the hand is resting on, is also reconstructed as part of the mesh. Hence, in order to remove the background, we perform a robust plane fitting based on RANSAC [11], where a plane is fitted to the flat background surface. To this end, we sample 100 random configurations of three vertices, fit

a plane to the sampled points, and then count the number of inliers. Any point that has a distance to the fitted plane that is smaller than the median edge length of the input scanned mesh is considered as inlier. Eventually, the plane that leads to the largest inlier count is considered the background plane. We have empirically found that this approach is robust and able to reliably identify the flat surface in all cases. Eventually, we use a combination of distance-based and color-based thresholding to discard background vertices in the scanned mesh. In particular, we discard a vertex if its distance from the background plane is less than 1cm and the difference between the red and green channel of the vertex color is smaller than 30 (RGB $\in [0, 255]^3$). This yields better preservation of hand vertices that are close to the background plane.

*MANO Model Fitting.* Subsequently, we fit the MANO hand model to the filtered hand scan mesh (*i.e.,* the one without background). To this end, we first obtain the MANO shape and pose parameters based on the hand tracking approach of Mueller *et al.* [29]. The approach uses a Gauss-Newton optimization scheme that makes use of additional information based on trained machine learning predictors (*e.g.,* for correspondence estimation). Since their method was developed for 3D reconstruction and tracking of hands in *depth images*, we render synthetic depth images from our partial hand scan meshes. Note that the approach [29] was partially trained on synthetic depth images and thus we have found that it is able to produce sufficiently good fits of the MANO geometry to our data.

However, since the MANO model is relatively coarse (778 vertices), and more importantly, it has a limited expressivity of hand shape (it only spans the variations of their training set of 31 subjects), we have found that there are still some misalignments. To also allow for deformations outside the shape space of the MANO model, we hence use a complementary non-rigid refinement of the previously fitted MANO mesh to the hand scan. To this end, we use a variant of non-rigid *iterative closet point* (ICP) [4] that optimizes for individual vertex displacements that further refine the template, which in our case is the fitted MANO model. As our objective function, we use 3D point-to-point and point-to-plane distances together with a spatial smoothness regularizer [14]. An accurate alignment is especially important at salient points, like fingertips, to ensure high perceptual quality. Hence, we add prior correspondences for the fingertips and the wrist to the non-rigid ICP fitting. We automatically obtain these correspondences in the input scanned mesh using OpenPose [43]. The influence of the prior correspondences is shown in our evaluation (see Sect. 5.1).

*Texture Mapping.* After having obtained an accurate alignment of the hand template, *i.e.*, the fitted MANO model plus non-rigid deformation for refinement, to our textured high-resolution hand scan, we transfer the scan texture to a texture map. To this end, we have manually defined UV coordinates for the MANO model template by unwrapping the mesh to a plane (see texture mapping step in Fig. 2). We project each vertex in the high-resolution hand scan to the closest point on the surface of the fitted MANO hand template. Using the barycentric coordinates of this projected point together with the UV coordinates of the

template mesh, we transfer the color to the texture map. After performing this procedure for all vertices of our high-resolution hand scan, there can still be some texels (pixels in the texture map) that are not set (we have found that about 6.5% of the hand interior does not have a defined texture). To deal with that, holes are filled based on inpainting with neighboring texels.

*Shading Removal.* We ensured that our scans have low-frequency shading by using controlled lighting. Thus, we implicitly made the assumptions of having a mostly Lambertian surface and no casted shadows. Since the smooth shading effects have low frequency (see Fig. 4a), they can be separated and removed using a frequency-based method like the Laplacian image pyramid. To this end, we first build a Laplacian pyramid with five levels from the texture map that we obtained in the previous step. We observe that the deepest level separates the (almost) constant skin color as well as the smooth shading from the texture details that are kept on earlier levels of the pyramid. We replace this deepest level with a constant skin color for palm and back side, respectively, effectively removing the smooth shading. We obtain this constant skin color by averaging in the well-lit area (see blue rectangles in Fig. 4). Note how the texture details from higher levels are preserved in the modified texture map (see Fig. 4b).

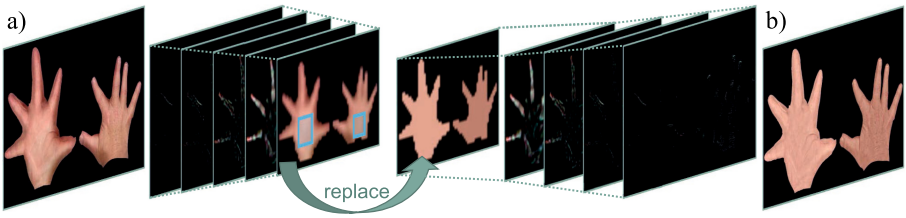


**Fig. 4. Shading removal.** (a) Original texture and its Laplacian pyramid decomposition. (b) The shading effects are removed by modifying the deepest level. (Color figure online)

*Seamless Texture Stitching.* Since so far this texture mapping is performed both for the palm-up and palm-down facing meshes, we eventually blend both partial texture maps to obtain a complete texture map of the hand. To this end, we use a recent gradient-domain texture stitching approach that directly operates in the texture atlas domain while preserving continuity induced by the 3D mesh topology across atlas chart boundaries [34].

### 3.3 Texture Model Creation

Let $\{T_i\}_{i=1}^n$ be the collection of 2D texture maps that we obtain after data canonicalization as described in Sect. 3.2. In order to create a parametric texture model we employ PCA. We vectorize each $T_i$ to obtain the vector $t_i \in \mathbb{R}^{618,990}$ that

stacks the red, green and blue channels of all hand texels. PCA first computes the data covariance matrix

$$C = \frac{1}{n-1} \sum_{i=1}^{n} (t_i - \bar{t}) (t_i - \bar{t})^\top , \tag{1}$$

for $\bar{t} = \frac{1}{n} \sum_{i=1}^{n} t_i$ being the average texture. Subsequently, eigenvalue decomposition of $C = \Phi \Lambda \Phi^T$ is used to obtain the principal components $\Phi$ and the diagonal matrix of eigenvalues $\Lambda$. With that we obtain the parametric texture model for the parameter vector $\alpha \in \mathbb{R}^k, k = 101$ as

$$t(\alpha) = \bar{t} + \Phi \, \alpha \, . \tag{2}$$

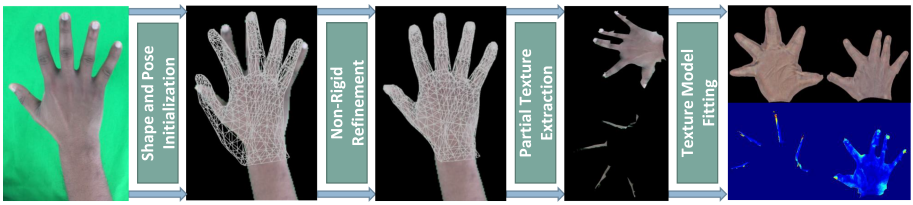


**Fig. 5. 3D hand personalization from a single image.** Starting from a single RGB input image (left), we first initialize the mesh using the method by Boukhayma *et al.* [6]. Next, we refine the fit non-rigidly and extract the partial hand texture. By fitting our parametric texture model, we are able to obtain a complete texture which minimizes the error to the input texture (right).

## 4 Applications

To demonstrate possible use cases of our parametric hand appearance model, we present two applications. First, we consider 3D hand reconstruction and personalization from a single monocular RGB image. Subsequently, we show the usage as a neural network layer enabling a self-supervised photometric loss.

### 4.1 3D Hand Personalization from a Single Image

Given a single monocular RGB image of a hand, we aim to reconstruct a 3D hand mesh that is personalized to the user's shape and appearance. This application consists of four steps: (1) initialization of shape and pose parameters of the MANO model, (2) non-rigid shape and pose refinement, (3) partial texture extraction, and (4) estimation of appearance parameters of our model.

*Shape and Pose Initialization.* We use the method of Boukhayma *et al.* [6] to obtain an initial pose and shape estimate of the MANO template mesh from a single RGB image. As discussed before, the MANO shape space is not always expressive enough to perfectly fit the user's hand shape. In addition, the results from the method by Boukhayma *et al.* do not yield sufficiently accurate reprojection of the mesh onto the image plane as shown in Fig. 5 (second from the left). Hence, this initial mesh is further refined.

*Non-Rigid Refinement of the Initial Mesh.* We non-rigidly refine the initial mesh estimate to better fit the hand silhouette in the image. Therefore, we optimize the 3D displacement of each vertex using ICP constraints on the boundary vertices. We define the set of boundary vertices of the hand mesh $\bar{\mathcal{V}} \subset \mathcal{V}$, *i.e.*, the set of vertices on the silhouette. Let $\Pi : \mathbb{R}^3 \rightarrow \Omega$ be the camera projection converting from 3D world coordinates to 2D pixel locations. For each boundary vertex $\bar{\boldsymbol{v}}_i$, we first find the closest hand silhouette pixel $\bar{\boldsymbol{p}}_i$ in the image domain $\Omega$ as

$$\bar{\boldsymbol{p}}_i = \underset{\boldsymbol{p} \in \Omega}{arg\,min}\, ||\Pi(\bar{\boldsymbol{v}}_i) - \boldsymbol{p}||_2 \quad \text{s.t.} \quad n(\boldsymbol{p})^\top \Pi(n(\bar{\boldsymbol{v}}_i)) > \eta\,. \tag{3}$$

Here, $n(\boldsymbol{p})$ is the 2D boundary normal at pixel $\boldsymbol{p}$ (calculated by Sobel filtering), and $\Pi(n(\bar{\boldsymbol{v}}_i))$ is the 2D image-plane projection of the 3D vertex normal at $\bar{\boldsymbol{v}}_i$. The threshold $\eta = 0.8$ discards unsuitable pixels based on normal dissimilarity. We then use this closest hand silhouette pixel $\bar{\boldsymbol{p}}_i$ as correspondence for boundary vertex $\bar{\boldsymbol{v}}_i$ if it is closer than $\delta$ (= 4% of the image size):

$$\bar{\boldsymbol{c}}_i = \begin{cases} \bar{\boldsymbol{p}}_i, & \text{if } ||\Pi(\bar{\boldsymbol{v}}_i) - \bar{\boldsymbol{p}}_i||_2 < \delta \\ \emptyset, & \text{otherwise} \end{cases} . \tag{4}$$

We can then optimize for the refined 3D vertex positions using the computed correspondences in the following objective function:

$$E(\mathcal{V}) = \frac{1}{|\bar{\mathcal{V}}|} \sum_{\bar{\boldsymbol{v}}_i \in \bar{\mathcal{V}}} ||\Pi(\bar{\boldsymbol{v}}_i) - \bar{\boldsymbol{p}}_i||_2^2 + w_{\text{smth}} \sum_{\boldsymbol{v}_j \in \mathcal{V}} \sum_{\boldsymbol{v}_k \in \mathcal{N}_j} \frac{1}{|\mathcal{N}_j|} ||(\boldsymbol{v}_j - \boldsymbol{v}_k) - (\boldsymbol{v}_j^0 - \boldsymbol{v}_k^0)||_2^2, \tag{5}$$

where $\mathcal{N}_j$ is the set of neighboring vertices of $\boldsymbol{v}_j$, and $\mathcal{V}^0 = \{\boldsymbol{v}_\bullet^0\}$ are the vertex positions from the previous ICP iteration. In total, we use 20 ICP iterations and initialize $\mathcal{V}, \mathcal{V}^0$ from the shape and pose initialization step as described above.

*Partial Texture Extraction.* For each fully visible triangle, *i.e.*, when all its 3 vertices are visible, we extract the color from the input image and copy it to the texture map. This yields a partial texture map where usually at most half the texels have a value assigned and all other texels are set to $\emptyset$. We then obtain the vectorized target texture map $t^{\text{trgt}}$ (as for model creation in Sect. 3.3).

*Estimation of Appearance Parameters.* Subsequently, we find the appearance parameters of our model that best fit the user's hand by solving the least-squares problem with Tikhonov regularization:

$$\underset{\alpha \in \mathbb{R}^k}{arg\,min} \sum_{t_i^{\text{trgt}} \neq \emptyset} (t_i^{\text{trgt}} - t(\alpha)_i)^2 + w_{\text{reg}} ||\alpha||_2^2 \,. \tag{6}$$

Note that our proposed parametric appearance model enables us to obtain a complete texture. In contrast to the extracted partial texture, the result is free of lighting effects and artifacts caused by small misalignments of the hand model.

### 4.2 Self-supervised Photometric Loss

Previous works have trained neural networks to regress joint positions or MANO model parameters from RGB images [7,28,46,55,57,58]. The most common loss is the Euclidean distance between the regressed and ground truth joint positions. Some works have also explored a silhouette loss between the mesh and the hand region in the image [2,6,56]. Our HTML enables the use of a *self-supervised* photometric loss, which complements the existing fully supervised losses. With that, when training a network to predict shape and pose with such an approach, we additionally obtain a hand texture estimate. To this end, we introduce a *textured hand model layer*, which we explain now.

*Textured Hand Model Layer.* Given a pair of MANO shape and pose parameters $(\beta, \theta)$, as well as the texture parameters $\alpha$, our model layer computes the textured 3D hand mesh $\mathcal{M}(\beta, \theta, \alpha)$. An image of this mesh is then rendered using a scaled orthographic projection. As such, this rendered image can directly be compared to the input image $\mathcal{I}$ using a photometric loss in an analysis-by-synthesis manner. We formulate the photometric loss as

$$\mathcal{L}_{\text{photo}}(\beta, \theta, \alpha) = \frac{1}{|\Gamma|} \sum_{(u,v) \in \Gamma} ||\text{render}(\mathcal{M}(\beta, \theta, \alpha))(u, v) - \mathcal{I}(u, v)||_2 \,, \tag{7}$$

where $\Gamma$ is the set of pixels which the estimated hand mesh projects to. The use of a differentiable renderer makes the photometric loss $\mathcal{L}_{\text{photo}}$ fully differentiable and enables backpropagation for neural network training.

*Network Training.* We train a residual network with the architecture of ResNet-34 [17] to regress the shape $\beta$, pose $\theta$, and texture parameters $\alpha$ from a given input image. In addition to the self-supervised photometric loss, we employ losses on 2D joint positions, 3D joint positions, and L2-regularizers on the magnitude of the shape, pose, and texture parameters. The network is trained in PyTorch [32], using the differentiable renderer provided in PyTorch3D [36]. We assume a single fixed illumination condition for training. We leave the joint estimation of additional lighting and material properties to future work.

## 5 Experiments

In this section, we evaluate our proposed parametric hand texture model, explore different design choices in our texture acquisition pipeline, and present results of our two example applications.

### 5.1 Texture Model Evaluation

*Compactness.* Figure 6 (left) shows the compactness of our texture model. The plot describes how much the explained variance in the training dataset increases with the number of used principal components. The first few components already explain a significant amount of variation since they account for more global changes in the texture, *e.g.*, skin tone. However, adding more components continuously increases the explained variance.

*Generalization.* For evaluating generalization, we use a leave-one-subject-out protocol. We remove the data of one subject, *i.e.*, the two texture samples from left and right hand, and rebuild the PCA model. Then, we reconstruct the left-out textures using the built model and measure the reconstruction error as the mean absolute distance (MAD) of the vectorized textures. As shown in Fig. 6 (middle), the reconstruction error decreases monotonically for an increasing number of components for both of the two models.

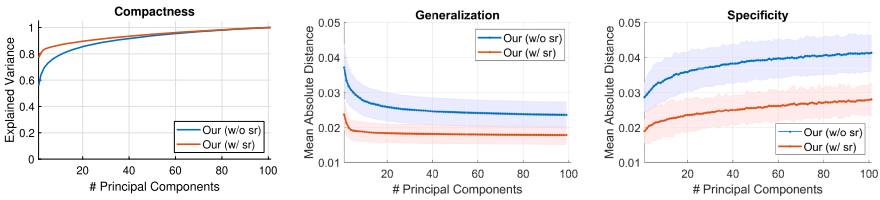


**Fig. 6.** Evaluation of compactness, generalization and specificity. Using shading removal ("w/ sr") substantially outperforms not using shading removal ("w/o sr").

*Specificity.* We also report the specificity, which quantifies the similarity between random samples from the model and the training data. To this end, we first sample a texture instance from our model based on a multivariate standard Normal distribution (due to the Gaussian assumption of PCA). Then, we find the nearest texture in our training dataset in terms of the MAD. We repeat this procedure 200 times, and report the statistics of the MAD in Fig. 6 (right).

*Influence of Shading Removal.* Figure 6 also shows compactness, generalization, and specificity for a version of the texture model that was built without shading removal ("w/o sr"). It can be seen that the version without shading removal performs worse compared to the one with shading removal ("w/ sr") in all metrics. When the lighting effects are not removed, they increase the variance in the training dataset. Hence, more principal components are necessary to explain variation and the reconstruction of unseen test samples has a higher error. In the supplemental material, we also show visually that the principal components for the model without shading removal have to account for strong lighting variation.

*Influence of Prior Correspondences.* To ensure a good alignment of the hand template mesh and the scanned mesh, as explained in Sect. 3.2, for the non-rigid ICP-based refinement step in our model building stage we make use of prior correspondences for the fingertips and the wrist. Figure 7 compares the textures obtained by running the non-rigid ICP fitting with and without them. Especially for the thumb, the tip is often not well-aligned, resulting in a missing finger nail in the texture. Using explicit prior correspondences alleviates this issue.

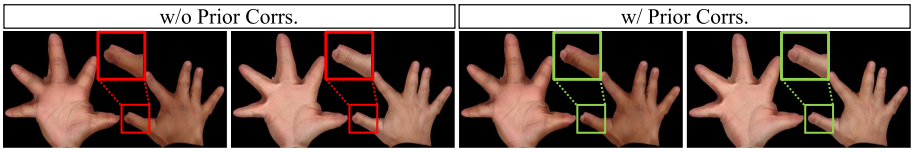


**Fig. 7.** Using non-rigid ICP-based refinement with prior correspondences for fingertips and the wrist improves the alignment of the hand template mesh to the scanned mesh, yielding better textures (right). (Textures shown before shading removal.)

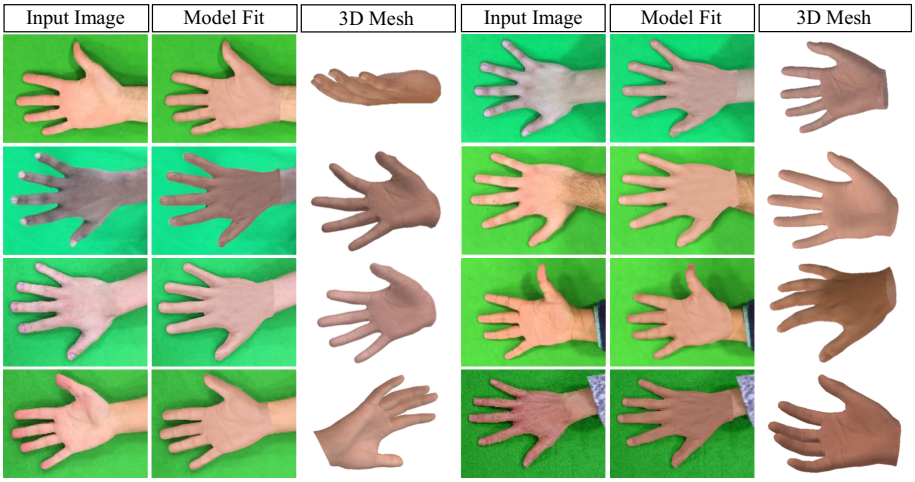


**Fig. 8.** Hand personlization from a single RGB input image for different subjects. (Color figure online)

### 5.2 Application Results: 3D Hand Personalization

Here, we show results for obtaining a personalized 3D hand model from a single RGB image (see Sect. 4.1). As previously discussed, since the output meshes of state-of-the-art regression approaches [6] do not have a low reprojection error, we use non-rigid refinement based on silhouettes. To simplify segmentation in our example application, we captured the images of the users in front of a green

screen. In future work, this could be replaced by a dedicated hand segmentation method. Figure 8 shows hand model fits and complete recovered textures from a single RGB image for several subjects. Since we use a low-dimensional PCA space to model hand texture variation, we can robustly estimate a plausible and complete texture from noisy or partially corrupted input (see Fig. 9). In contrast, a texture that is directly obtained by projecting the input image onto a mesh obtained by the method of Boukhayma *et al.* [6] contains large misalignments and a significant amount of background pixels, and thus is severely corrupted.

### 5.3 Application Results: Photometric Neural Network Loss

Our self-supervised photometric loss (see Sect. 4.2) enables to not only obtain shape and pose estimates as in previous work, but in addition to also estimate hand appearance. To demonstrate this we train our network on the recently proposed FreiHAND dataset [58]. For details of the experimental setup, please see the document. In Fig. 10, we show hand model fits predicted by a neural network trained with and without our photometric loss (cf. Sect. 4.2). We note that the pose and shape prediction with the photometric loss are quantitatively similar to the predictions without (the mean aligned vertex errors (MAVE) are 1.10 cm vs 1.14 cm respectively, and mean aligned keypoint errors (MAKE) are 1.11 cm vs 1.14 cm respectively). In addition, these results are comparable to the current state of the art [58] with a MAVE of 1.09 cm and MAKE of 1.10 cm. We stress that our method with the photometric loss additionally infers a high resolution, detailed texture of the full hand, which the other methods do not.

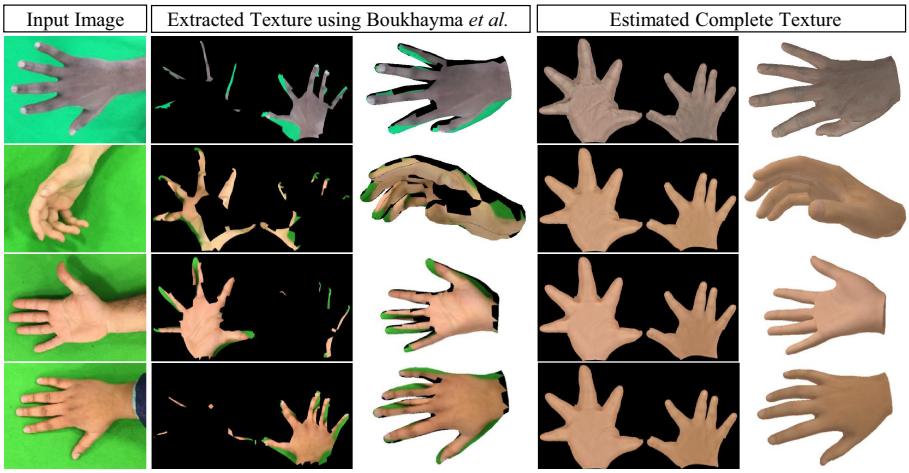


**Fig. 9.** Fitting to noisy or corrupted input textures is robust and yields a realistic and complete texture estimate due to the low-dimensional PCA space built by our model.

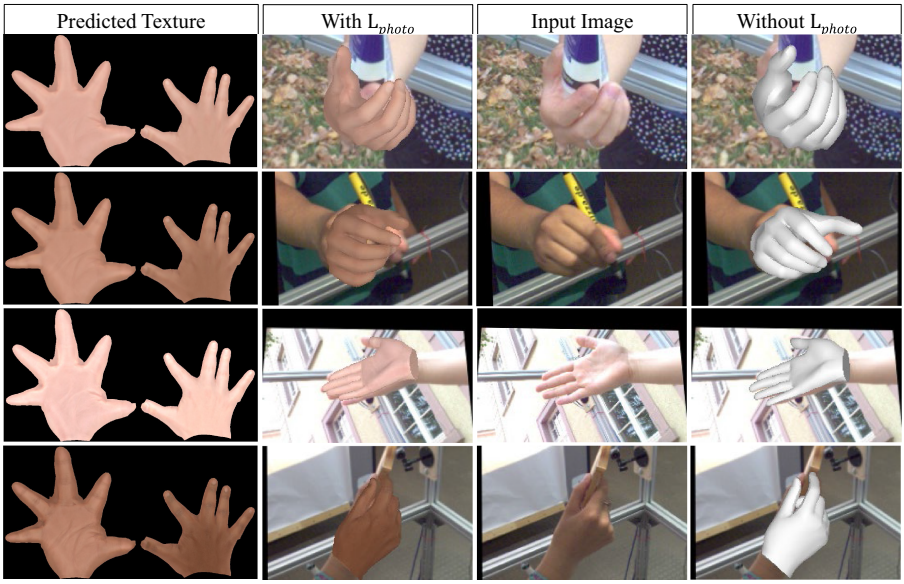


**Fig. 10.** We show the predicted pose and texture from a neural network trained using a photometric loss $L_{photo}$ enabled by our parametric hand texture model.

## 6 Limitations and Discussion

Our experiments have shown that HTML can be used to recover personalized 3D hand shape and appearance. Although our model provides detailed texture, the underlying geometry of the MANO mesh is coarse (778 vertices). This could be improved by using a higher-resolution mesh and extending the MANO shape space with more detailed geometry. Non-linear models, *e.g.,* an autoencoder neural network, can be explored for capturing variations that a linear PCA model cannot. As hand appearance varies during articulation, modeling pose-dependent texture changes can increase the realism. This would need a more complicated capture and registration setup and a significantly larger dataset to capture the whole pose space and diverse users. In terms of applications, estimating lighting in addition to or jointly with the texture parameters can better reconstruct input observations. Correctly modeling lighting for hands, where shadow casting often occurs, is a challenge that would need to be addressed. Other applications of our model, such as exploring how self-supervision can alleviate the need for keypoint annotations or improve pose estimation, can be directions for future research.

## 7 Conclusion

In this work, we introduced HTML—the first parametric texture model of hands. The model is based on data that captures 102 hands of people with varying

gender, age and ethnicity. For model creation, we carefully designed a data canonicalization pipeline that entails background removal, geometric model fitting, texture mapping, and shading removal. Moreover, we demonstrated that our model enables two highly relevant applications: 3D hand personalization from a single RGB image, and learning texture estimation using a self-supervised loss. We make our model publicly available to encourage future work in the area.

**Acknowledgments.** The authors thank all participants of the data recordings. This work was supported by the ERC Consolidator Grant 4DRepLy (770784).

## References

1. Baek, S., In Kim, K., Kim, T.K.: Augmented skeleton space transfer for depth-based hand pose estimation. In: Computer Vision and Pattern Recognition (CVPR) (2018)
2. Baek, S., Kim, K.I., Kim, T.K.: Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering. In: Computer Vision and Pattern Recognition (CVPR) (2019)
3. Bailer, C., Finckh, M., Lensch, H.P.A.: Scale robust multi view stereo. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 398–411. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33712-3_29
4. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. Trans. Pattern Anal. Mach. Intell. (TPAMI) **14**(2), 239–256 (1992)
5. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: SIGGRAPH, pp. 187–194 (1999)
6. Boukhayma, A., de Bem, R., Torr, P.H.: 3D hand shape and pose from images in the wild. In: Computer Vision and Pattern Recognition (CVPR) (2019)
7. Cai, Y., Ge, L., Cai, J., Yuan, J.: Weakly-supervised 3D hand pose estimation from monocular RGB images. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11210, pp. 678–694. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01231-1_41
8. Chen, Y., Tu, Z., Ge, L., Zhang, D., Chen, R., Yuan, J.: SO-HandNet: self-organizing network for 3D hand pose estimation with semi-supervised learning. In: International Conference on Computer Vision (ICCV) (2019)
9. Dai, H., Pears, N., Smith, W.A., Duncan, C.: A 3D morphable model of craniofacial shape and texture variation. In: International Conference on Computer Vision (ICCV), pp. 3085–3093 (2017)
10. Egger, B., et al.: 3D morphable face models - past, present and future. ACM Trans. Graph. **39**, 157:1–157:38 (2020)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
12. Ge, L., Cai, Y., Weng, J., Yuan, J.: Hand PointNet: 3D hand pose estimation using point sets. In: Computer Vision and Pattern Recognition (CVPR) (2018)
13. Gerig, T., et al.: Morphable face models-an open framework. In: International Conference on Automatic Face & Gesture Recognition (FG), pp. 75–82 (2018)

14. Habermann, M., Xu, W., Zollhoefer, M., Pons-Moll, G., Theobalt, C.: LiveCap: real-time human performance capture from monocular video. ACM Trans. Graph. (TOG) **38**(2), 1–17 (2019)
15. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: HOnnotate: a method for 3D annotation of hand and object poses. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3196–3206 (2020)
16. Hasson, Y., et al.: Learning joint reconstruction of hands and manipulated objects. In: Computer Vision and Pattern Recognition (CVPR) (2019)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
18. Huber, P., et al.: A multiresolution 3D morphable face model and fitting framework. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP) (2016)
19. Jung, S., Hughes, C.: Body ownership in virtual reality. In: International Conference on Collaboration Technologies and Systems (CTS), pp. 597–600 (10 2016)
20. Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., Fitzgibbon, A.: Learning an efficient model of hand shape variation from depth images. In: Computer Vision and Pattern Recognition (CVPR) (2015)
21. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: International Symposium on Mixed and Augmented Reality (ISMAR) (2007)
22. Kovalenko, O., Golyanik, V., Malik, J., Elhayek, A., Stricker, D.: Structure from articulated motion: accurate and stable monocular 3D reconstruction without training data. Sensors **19**(20), 4603 (2019)
23. de La Gorce, M., Fleet, D.J., Paragios, N.: Model-based 3D hand pose estimation from monocular video. Trans. Pattern Anal. Mach. Intell. (PAMI) **33**(9), 1793–1805 (2011)
24. de La Gorce, M., Paragios, N., Fleet, D.J.: Model-based hand tracking with texture, shading and self-occlusions. In: Computer Vision and Pattern Recognition (CVPR) (2008)
25. Li, S., Lee, D.: Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In: Computer Vision and Pattern Recognition (CVPR) (2019)
26. Malik, J., et al.: HandVoxNet: deep voxel-based network for 3D hand shape and pose estimation from a single depth map. In: Computer Vision and Pattern Recognition (CVPR) (2020)
27. Malik, J., et al.: DeepHPS: end-to-end estimation of 3D hand pose and shape by learning from synthetic depth. In: International Conference on 3D Vision (3DV) (2018)
28. Mueller, F., et al.: GANerated hands for real-time 3D hand tracking from monocular RGB. In: Computer Vision and Pattern Recognition (CVPR) (2018)
29. Mueller, F., et al.: Real-time pose and shape reconstruction of two interacting hands with a single depth camera. ACM Trans. Graph. (TOG) **38**(4), 1–13 (2019)
30. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: International Conference on Computer Vision (ICCV) (2015)
31. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3D tracking of hand articulations using Kinect. In: British Machine Vision Conference (BMVC) (2011)
32. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 8024–8035 (2019)

33. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 296–301 (2009)
34. Prada, F., Kazhdan, M., Chuang, M., Hoppe, H.: Gradient-domain processing within a texture atlas. ACM Trans. Graph. (TOG) **37**(4), 154:1–154:14 (2018)
35. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: Computer Vision and Pattern Recognition (CVPR) (2014)
36. Ravi, N., et al.: PyTorch3D (2020). https://github.com/facebookresearch/pytorch3d
37. Remelli, E., Tkach, A., Tagliasacchi, A., Pauly, M.: Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In: International Conference on Computer Vision (ICCV) (2017)
38. Romdhani, S., Vetter, T.: Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. Comput. Vis. Pattern Recogn. (CVPR) **2**, 986–993 (2005)
39. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: modeling and capturing hands and bodies together. ACM Trans. Graph. (Proc. SIGGRAPH Asia) **36**(6), 245:1–245:17 (2017)
40. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Computer Vision and Pattern Recognition (CVPR) (2016)
41. Sharp, T., et al.: Accurate, robust, and flexible real-time hand tracking. In: ACM Conference on Human Factors in Computing Systems (CHI) (2015)
42. Shiffman, M.A., Mirrafati, S., Lam, S.M., Cueteaux, C.G.: Simplified Facial Rejuvenation. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71097-4
43. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: Computer Vision and Pattern Recognition (CVPR) (2017)
44. SONY 3D Creator. https://3d-creator.sonymobile.com/
45. Sony Corporation: 3D Creator App (White Paper), version 3: August 2018 (2018). https://dyshcs8wkvd5y.cloudfront.net/docs/3D-Creator-Whitepaper.pdf
46. Spurr, A., Song, J., Park, S., Hilliges, O.: Cross-modal deep variational hand pose estimation. In: Computer Vision and Pattern Recognition (CVPR) (2018)
47. Sridhar, S., Oulasvirta, A., Theobalt, C.: Interactive markerless articulated hand motion tracking using RGB and depth data. In: International Conference on Computer Vision (ICCV) (2013)
48. Taylor, J., et al.: Articulated distance fields for ultra-fast tracking of hands interacting. ACM Trans. Graph. (TOG) **36**(6), 1–12 (2017)
49. Tewari, A., et al.: MoFA: model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In: International Conference on Computer Vision (ICCV) (2017)
50. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2Face: real-time face capture and reenactment of RGB videos. In: Computer Vision and Pattern Recognition (CVPR) (2016)
51. Tkach, A., Pauly, M., Tagliasacchi, A.: Sphere-meshes for real-time hand modeling and tracking. ACM Trans. Graph. (ToG) **35**(6), 1–11 (2016)
52. Tkach, A., Tagliasacchi, A., Remelli, E., Pauly, M., Fitzgibbon, A.: Online generative model personalization for hand tracking. ACM Trans. Graph. (ToG) **36**(6), 1–11 (2017)
53. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph. **33**, 169:1–169:10 (2014)

54. Wan, C., Probst, T., Van Gool, L., Yao, A.: Crossing nets: combining GANs and VAEs with a shared latent space for hand pose estimation. In: Computer Vision and Pattern Recognition (CVPR) (2017)
55. Yang, L., Li, S., Lee, D., Yao, A.: Aligning latent spaces for 3D hand pose estimation. In: International Conference on Computer Vision (ICCV) (2019)
56. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular RGB image. In: International Conference on Computer Vision (ICCV) (2019)
57. Zimmermann, C., Brox, T.: Learning to estimate 3D hand pose from single RGB images. In: International Conference on Computer Vision (ICCV) (2017)
58. 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: International Conference on Computer Vision (ICCV) (2019)