Model-based 3D Hand Reconstruction via Self-Supervised Learning

Yujin Chen[1*] Zhigang Tu[1†] Di Kang[2] Linchao Bao[2]
Ying Zhang[3] Xuefei Zhe[2] Ruizhi Chen[1] Junsong Yuan[4]
[1]Wuhan University [2]Tencent AI Lab [3]Tencent [4]State University of New York at Buffalo
{yujin.chen, tuzhigang, ruizhi.chen}@whu.edu.cn {di.kang, zhexuefei}@outlook.com
linchaobao@gmail.com yinggzhang@tencent.com jsyuan@buffalo.edu

Abstract

Reconstructing a 3D hand from a single-view RGB image is challenging due to various hand configurations and depth ambiguity. To reliably reconstruct a 3D hand from a monocular image, most state-of-the-art methods heavily rely on 3D annotations at the training stage, but obtaining 3D annotations is expensive. To alleviate reliance on labeled training data, we propose S^{2}HAND*, a self-supervised 3D hand reconstruction network that can jointly estimate pose, shape, texture, and the camera viewpoint. Specifically, we obtain geometric cues from the input image through easily accessible 2D detected keypoints. To learn an accurate hand reconstruction model from these noisy geometric cues, we utilize the consistency between 2D and 3D representations and propose a set of novel losses to rationalize outputs of the neural network. For the first time, we demonstrate the feasibility of training an accurate 3D hand reconstruction network without relying on manual annotations. Our experiments show that the proposed self-supervised method achieves comparable performance with recent fully-supervised methods. The code is available at* https://github.com/TerenceCYJ/S2HAND.

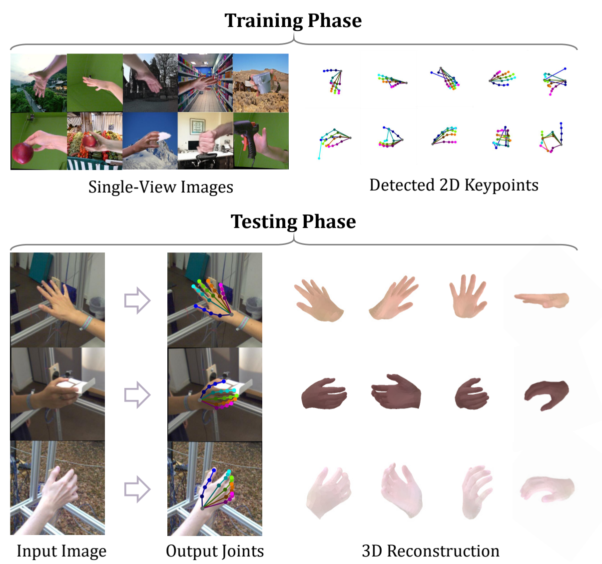

Figure 1: Given a collection of unlabeled hand images, we learn a 3D hand reconstruction network in a self-supervised manner. Top: the training uses a collection of unlabeled hand images and their corresponding noisy detected 2D keypoints. Bottom: our model outputs accurate hand joints and shapes, as well as vivid textures.

1. Introduction

Reconstructing 3D human hands from a single image is important for computer vision tasks such as hand-related action recognition, augmented reality, sign language translation, and human-computer interaction [21, 33, 43]. However, due to the diversity of hands and the depth ambiguity in monocular 3D reconstruction, image-based 3D hand reconstruction remains a challenging problem.

In recent years, we have witnessed fast progress in recovering 3D representations of human hands from images. In this field, most methods were proposed to predict 3D hand pose from the depth image [1, 10, 15, 22, 49] or the RGB image [2, 8, 24, 37, 52]. However, the surface information is needed in some applications such as grasping an object by a virtual hand [21], where the 3D hand pose represented by sparse joints is not sufficient. To better display the surface information of the hand, previous studies predict the triangle mesh either via regressing per-vertex coordinate [16, 29] or by deforming a parametric hand model [19, 20]. Outputting such high-dimensional representation from 2D input is challenging for neural networks to learn, thus resulting in the training process relying heavily on 3D annotations such as dense hand scans, model-fitted parametric hand mesh, or human-annotated 3D joints. Besides, the hand texture is important in some applications, such as vivid hands reconstruction in immersive virtual reality. But only recently has a study exploring parametric texture estimation in a learning-based hand recovery system [35], while most

*Work done during an internship at Tencent AI Lab.
†Corresponding author: tuzhigang@whu.edu.cn

Approach	Supervision	Outputs
[35]	3DM, 3DJ, 2DKP, I, TI	3DM, 3DJ, Tex
[20, 30]	3DM, 3DJ	3DM, 3DJ
[48]	3DM*, 3DJ, 2DKP, 2DS, Syn	3DM, 3DJ
[16]	3DM*, 3DJ, 2DKP, D*	3DM, 3DJ
[29]	3DM*, D2DKP	3DM, 3DJ
[51]	3DJ, 2DKP, Mo	3DM, 3DJ
[3, 7, 50]	3DJ, 2DKP, 2DS	3DM, 3DJ
[52]	3DJ, 2DKP, 2DS	3DJ
[24]	3DJ, 2DKP	3DJ
[4]	3DJ*, 2DKP, 2DS	3DM, 3DJ, Tex
[37]	3DJ*, 2DKP	3DJ
[8]	2DKP, D	3DJ
Ours	D2DKP, I	3DM, 3DJ, Tex

Table 1: A comparison of some representative 3D hand recovery approaches with highlighting the differences between the supervision and the outputs. We use the weakest degree of supervision and output the most representations. 3DM: 3D mesh, 3DJ: 3D joints, I: input image, TI: an additional set of images with clear hand texture, Tex: texture, 2DKP: 2D keypoints, 2DS: 2D silhouette, D: depth, D2DKP: detected 2D keypoints, Syn: extra synthetic sequence data, Mo: extra motion capture data. * indicates that the study uses multiple datasets for training, and at least one dataset used the supervision item.

previous works do not consider texture modeling.

Our key observation is that the 2D cues in the image space are closely related to the 3D hand model in the real world. The 2D hand keypoints contain rich structural information, and the image contains texture information. Both are important for reducing the use of expensive 3D annotations but have not been investigated much. In this way, we could directly use 2D annotations and the input image to learn the structural and texture representations without using 3D annotations. However, it is still labor-consuming to annotate 2D hand keypoints. To completely save the cost of manual annotation, we propose to extract some geometric representations from the unlabeled image to help shape reconstruction and use the texture information contained in the input image to help texture modeling.

Motivated by the above observations, this work seeks to train an accurate and robust 3D hand reconstruction network only using supervision signals obtained from the input images and eliminate all manual annotations, which is the first attempt in this task. To this end, we use an off-the-shelf 2D keypoint detector [9] to produce some noisy 2D keypoints and supervise the hand reconstruction by these noisy detected 2D keypoints and the input image. To better achieve this goal, there are several issues that need to be addressed. First, how to efficiently use joint-wise 2D keypoints to supervise the ill-posed monocular 3D hand reconstruction? Second, since our setting does not use any ground truth annotation, how do we handle the noise in the 2D detection output?

To address the first issue, a model-based autoencoder is presented to estimate 3D joints and shape, where the output 3D joints are projected into image space and forced to align with the detected keypoints during training. However, if we only align keypoints in image space, invalid hand poses often occur. This may be an invalid 3D hand configure that could be projected to be the correct 2D keypoints. Also, 2D keypoints cannot reduce the scale ambiguity of the predicted 3D hand. Thus, we design a series of priors embedded in the model-based hand representations to help the neural network output hand with a reasonable pose and size.

To address the second issue, a trainable 2D keypoint estimator and a novel 2D-3D consistency loss are proposed. The 2D keypoint estimator outputs joint-wise 2D keypoints and the 2D-3D consistency loss links the 2D keypoint estimator and the 3D reconstruction network to make the two mutually beneficial to each other during the training. In addition, we find that the detection accuracy of different samples varies greatly, thus we propose to distinguish each detection item to weigh its supervision strength accordingly.

In summary, we present a $\mathbf{S^2HAND}$ (self-supervised 3D hand reconstruction) model which enables us to train a neural network that can predict 3D pose, shape, texture, and camera viewpoint from a hand image without any ground truth annotation, except that we use the outputs from a 2D keypoint detector (Fig. 1).

Our main contributions are summarized as follows:

- We present the first self-supervised 3D hand reconstruction network, which accurately outputs 3D joints, mesh, and texture from a single image, without using any annotated training data.
- We exploit an additional trainable 2D keypoint estimator to boost the 3D reconstruction through a mutual improvement manner, in which a novel 2D-3D consistency loss is proposed.
- We introduce a hand texture estimation module to learn vivid hand texture through self-supervision.
- We benchmark self-supervised 3D hand reconstruction on some currently challenging datasets, where our self-supervised method achieves comparable performance to previous fully-supervised methods.

2. Related Work

In this section, we review previous works that are related to our approach. Our focus is on model-based 3D hand pose and shape estimation, and 3D reconstruction with limited supervision. For more work on 3D pose estimation, please refer to [1, 16, 37]. Below and in Table 1, we compare our contribution with prior works.

Model-based Hand Pose and Shape Estimation. Many hand models have been proposed to approximate hand shape via a parametric model [5, 26, 36, 42]. In this paper, we employ a hand model named MANO [36] that maps pose and shape parameters to a triangle mesh [7, 11, 20, 53].

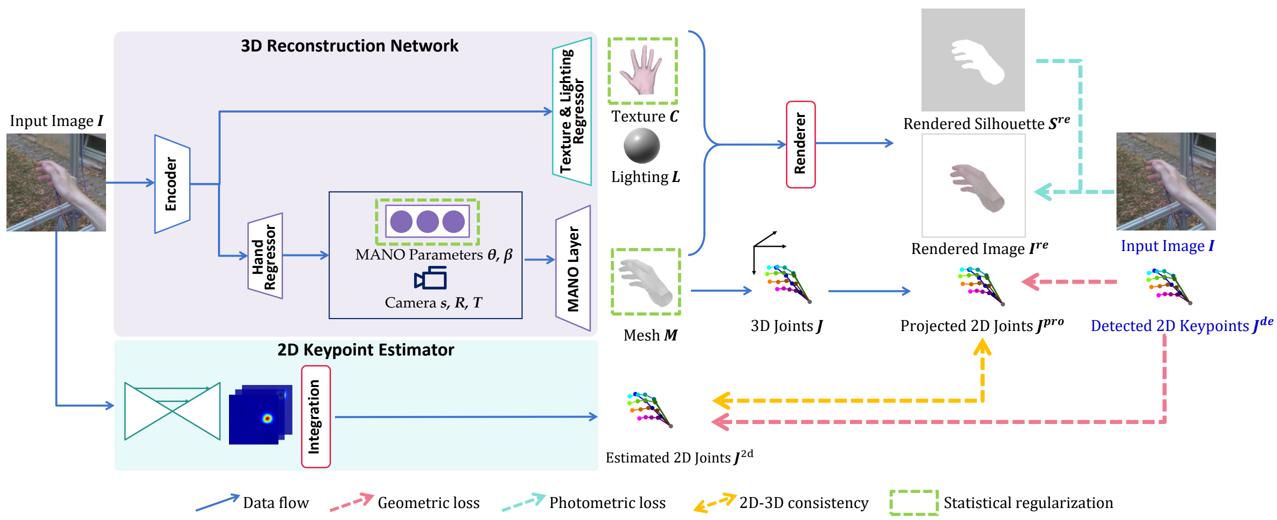

Figure 2: Overview of the proposed framework. Our 3D reconstruction network decomposes an input image I into pose, shape, viewpoint, texture, and lighting. The network is trained to reconstruct the input hand image and align with the detected 2D keypoints without extra ground truth annotation. We also adopt an additional trainable 2D keypoint estimator for joint-wise 2D keypoint estimation, which is supervised by the detected 2D keypoints as well. If the 2D keypoint estimator is enabled, a 2D-3D consistency function is introduced to link the 2D and 3D components for mutual improvement. During the inference, only the 3D reconstruction network is utilized. The "Input Image I" and "Detected 2D Keypoints J^{de}" on the right side of this figure are used to calculate losses.

Because the parametric model contains abundant structure priors of human hands, recent works integrate hand model as a differentiable layers in neural networks [3, 4, 7, 19, 20, 45, 51, 53]. Among them, [3, 45, 51] output a set of intermediate estimations, like segmentation mask and 2D keypoints, and then maps these representations to the MANO parameters. Different from them, we aim at demonstrating the feasibility of a self-supervised framework using an intuitive autoencoder. We additionally output 2D keypoint estimation from another branch and use it only during training to facilitate 3D reconstruction. More generally, recent methods [4, 7, 19, 20, 53] directly adopt an autoencoder that couples an image feature encoding stage with a model-based decoding stage. Unlike [19, 20], we focus on hand recovery and do not use any annotation about objects. More importantly, the above methods use 3D annotations as supervision, while the proposed method does not rely on any ground truth annotations.

3D Hand Pose and Shape Estimation with Limited Supervision. 2D annotation is cheaper than 3D annotation, but it is difficult to deal with the ambiguity of depth and scale. [8] use a depth map to perform additional weak supervision to strengthen 2D supervision. [37] proposes biomechanical constraints to help the network output feasible 3D hand configurations. [32] detects 2D hand keypoints and directly fits a hand model to the 2D detection. [29] gathers a large-scale dataset through an automated data collection method similar to [32] and then uses the collected mesh as supervision. In this work, we limit biomechanical feasibility by introducing a set of constraints on the skin model instead of only impose constraints on the skeleton as [37]. In contrast to [8, 29], our method is designed to verify the feasibility of (noisy) 2D supervision and do not introduce any extra 2.5D or 3D data.

Self-supervised 3D Reconstruction. Recently, there are methods that propose to learn 3D geometry from monocular image only. For example, [47] proposes an unsupervised approach to learn 3D deformable objects from raw single-view images, but they assume the object is perfectly symmetric, which is not the case in the hand reconstruction. [17] removes out keypoints from supervision signals, but it uses ground truth 2D silhouette as supervision and only deals with categories with small intra-class shape differences, such as birds, shoes, and cars. [44] proposes a depth-based self-supervised 3D hand pose estimation method, but the depth image provides much more strong evidence and supervision than the RGB image. Recently, [12, 40, 41] propose self-supervised face reconstruction with the use of 3D morphable model of face (3DMM) [6] and 2D landmarks detection. Our approach is similar to them, but the hand is relatively non-flat and asymmetrical when compared with the 3D face, and the hand suffers from more severe self-occlusion. These characteristics make this self-supervised hand reconstruction task more challenging.

Texture Modeling in Hand Recovery. [13, 14] exploit shading and texture information to handle the self-occlusion in the hand tracking system. Recently, [35] uses principal component analysis (PCA) to build a parametric texture model of hand from a set of textured scans. In this work, we try to model texture from self-supervised training without introducing extra data, and further investigate whether the texture modeling helps with the shape modeling.

From the above analysis and comparison of related work, we believe that self-supervised 3D hand reconstruction is feasible and significant, but to the best of our knowledge, no such idea has been studied in this field. In this work, we fill this gap and propose the first self-supervised 3D hand reconstruction network, and prove the effectiveness of the proposed method through experiments.

3. Method

Our method enables end-to-end learning of a 3D hand reconstruction network in a self-supervised manner, as illustrated in Fig. 2. To this end, we use an autoencoder that receives an image of a hand as input and outputs hand pose, shape, texture, and camera viewpoint (Section 3.1 and 3.2). We generate multiple 2D representations in image space (Section 3.3) and design a series of loss functions and regularization terms for network training (Section 3.4). In the following, we describe the proposed method in detail.

3.1. Deep Hand Encoding

Given a image I containing a hand, the network first uses an EfficientNet-b0 backbone [39] to encode the image into a geometry semantic code vector x and a texture semantic code vector y. The geometry semantic code vector x parameterizes the hand pose $\theta \in \mathbb{R}^{30}$, shape $\beta \in \mathbb{R}^{10}$, scale $s \in \mathbb{R}^1$, rotation $R \in \mathbb{R}^3$ and translation $T \in \mathbb{R}^3$ in a unified manner: $x = (\theta, \beta, s, R, T)$. The texture semantic code vector y parameterizes the hand texture $C \in \mathbb{R}^{778\times3}$ and scene lighting $L \in \mathbb{R}^{11}$ in a unified manner: $y = (C, L)$.

3.2. Model-based Hand Decoding

Given the geometry semantic code vector x and the texture semantic code vector y, our model-based decoder generates a textured 3D hand model in the camera space. In the following, we describe the used hand model and decoding network in detail.

Pose and Shape Representation. The hand surface is represented by a manifold triangle mesh $M \equiv (V, F)$ with $n = 778$ vertices $V = \{v_i \in \mathbb{R}^3 | 1 \le i \le n\}$ and faces F. The faces F indicates the connection of the vertices in the hand surface, where we assume the face topology keeps fixed. Given the mesh topology, a set of $k = 21$ joints $J = \{j_i \in \mathbb{R}^3 | 1 \le i \le k\}$ can be directly formulated from the hand mesh. Here, the hand mesh and joints are recovered from the pose vector θ and the shape vector β via MANO which is a low-dimensional parametric model learned from more than two thousand 3D hand scans [36].

3D Hand in Camera Space. After representing 3D hand via MANO hand model from pose and shape parameters, the mesh and joints are located in the hand-relative coordinate systems. To represent the output joints and mesh in the camera coordinate system, we use the estimated scale, rotation and translation to conserve the original hand mesh M_0 and joints J_0 into the final representations: $M = sM_0R + T$ and $J = sJ_0R + T$.

Texture and Lighting Representation. We use per-vertex RGB value of $n = 778$ vertices to represent the texture of hand $C = \{c_i \in \mathbb{R}^3 | 1 \le i \le n\}$, where c_i yields the RGB values of vertex i. In our model, we use a simple ambient light and a directional light to simulate lighting conditions [25]. The lighting vector L parameterizes ambient light intensity $l^a \in \mathbb{R}^1$, ambient light color $l^a_c \in \mathbb{R}^3$, directional light intensity color $l^d \in \mathbb{R}^1$, directional light color $l^d_c \in \mathbb{R}^3$, and directional light direction $n^d \in \mathbb{R}^3$ in a unified representation: $L = (l^a, l^a_c, l^d, l^d_c, n^d)$.

3.3. Represent Hand in 2D

A set of estimated 3D joints within the camera scope can be projected into the image space by camera projection. Similarly, the output textured model can be formulated into a realistic 2D hand image through a neural renderer. In addition to the 2D keypoints projected from the model-based 3D joints, we can also estimate the 2D position of each keypoint in the input image. Here, we represent 2D hand in three modes and explore the complementarity among them.

Joints Projection. Given a set of 3D joints in camera coordinates J and the intrinsic parameters of the camera, we use camera projection Π to project 3D joints into a set of $k = 21$ 2D joints $J^{pro} = \{j^{pro}_i \in \mathbb{R}^2 | 1 \le i \le k\}$, where j^{pro}_i yields the position of the i-th joint in image UV coordinates: $J^{pro} = \Pi(J)$.

Image Formation. A 3D mesh renderer is used to conserve the triangle hand mesh into a 2D image, here we use an implementation[1] of [25]. Given the 3D mesh M, the texture of the mesh C and the lighting L, the neural renderer Δ can generate a silhouette of hand S^{re} and a color image I^{re}: $S^{re}, I^{re} = \Delta(M, C, L)$.

Extra 2D Joint Estimation. Projecting model-based 3D joints into 2D helps the projected 2D keypoints retain structural information, but at the same time gives up the independence of each key point. In view of this matter, we additionally use a 2D keypoint estimator to directly estimate a set of $k = 21$ independent 2D joints $J^{2d} = \{j^{2d}_i \in \mathbb{R}^2 | 1 \le i \le k\}$, where j^{2d}_i indicates the position of the i-th joint in image UV coordinates. In our 2D keypoint estimator, a stacked hourglass network [31] along with an integral pose regression [38] is used. Note that the 2D hand pose estimation module is optionally deployed in the training period and is not required during the inference.

3.4. Training Objective

Our overall training loss E consists of three parts including 3D branch loss E_{3d}, 2D branch loss E_{2d}, and 2D-3D consistency loss E_{con}:

$$E = w_{3d}E_{3d} + w_{2d}E_{2d} + w_{con}E_{con} \quad (1)$$

[1] https://github.com/daniilidis-group/neural_renderer

Note, E_{2d} and E_{con} are optional and only used when the 2D estimator is applied. The constant weights w_{3d}, w_{2d} and w_{con} balance the three terms. In the following, we describe these loss terms in detail.

3.4.1 Losses of the 3D Branch

To train the model-based 3D hand decoder, we enforce geometric alignment E_{geo}, photometric alignment E_{photo}, and statistical regularization E_{regu}:

$$E_{3d} = w_{geo}E_{geo} + w_{photo}E_{photo} + w_{regu}E_{regu} \tag{2}$$

Geometric Alignment. We propose a geometric alignment loss E_{geo} based on the detected 2D keypoints which are obtained at an offline stage through an implementation[2] of [9]. The detected 2D keypoints $L = \{(j_i^{de}, con_i)|1 \leq i \leq k\}$ allocate each keypoint with a 2D position $j_i^{de} \in \mathbb{R}^2$ and a 1D confidence $con_i \in [0, 1]$. The geometric alignment loss in the 2D image space consists of a joint location loss E_{loc} and a bone orientation loss E_{ori}. The joint location loss E_{loc} enforces the projected 2D keypoints J^{pro} to be close to its corresponding 2D detections J^{de}, and the bone orientation loss E_{ori} enforces the $m = 20$ bones of these two sets of keypoints to be aligned.

$$E_{loc} = \frac{1}{k}\sum_{i=1}^{k} con_i \mathcal{L}_{SmoothL1}(j_i^{de}, j_i^{pro}) \tag{3}$$

$$E_{ori} = \frac{1}{m}\sum_{i=1}^{m} con_i^{bone} \| \nu_i^{de} - \nu_i^{pro} \|_2^2 \tag{4}$$

Here, a SmoothL1 loss [23] is used in Eq. 3 to make the loss term to be more robust to local adjustment since the detection keypoints are not fit well with the MANO keypoints. In Eq. 4, ν_i^{de} and ν_i^{pro} are the normalized i-th bone vector of the detected 2D joints and the projected 2D joints, respectively, and con_i^{bone} is the product of the confidence of the two detected 2D joints of the i-th bone. The overall geometric alignment loss E_{geo} is the weighted sum of E_{loc} and E_{ori} with a weighting factor w_{ori}:

$$E_{geo} = E_{loc} + w_{ori}E_{ori} \tag{5}$$

Photometric Consistency. For the image formation, the ideal result is the rendered color image I^{re} matches the foreground hand of the input I. To this end, we employ a photometric consistency which has two parts, the *pixel loss* E_{pixel} is computed by averaging the least absolute deviation (L1) distance for all visible pixels to measure the pixel-wise difference, and the structural similarity (SSIM) loss E_{SSIM} is the structural similarity between the two images [46].

$$E_{pixel} = \frac{con_{sum}}{| S^{re} |} \sum_{(u,v)\in S^{re}} \| I_{u,v} - I_{u,v}^{re} \|_2 \tag{6}$$

[2] https://github.com/Hzzone/pytorch-openpose

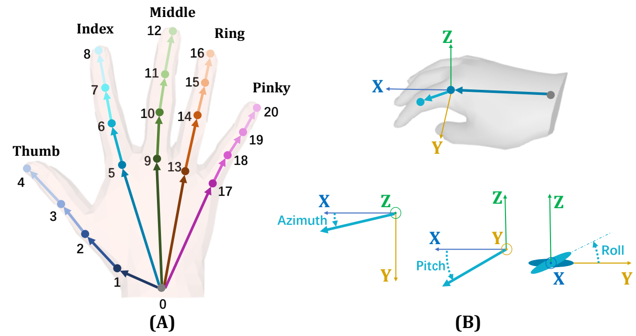

Figure 3: (A) Joint skeleton structure. (B) A sample of bone rotation angles. The five bones ($\overrightarrow{01}$,$\overrightarrow{05}$,$\overrightarrow{09}$,$\overrightarrow{013}$,$\overrightarrow{017}$) on the palm are fixed. Each finger has 3 bones, and the relative orientation of each bone from its root bone is represented by azimuth, pitch, and roll.

$$E_{SSIM} = 1 - SSIM(I \odot S^{re}, I^{re}) \tag{7}$$

Here, the rendered silhouette S^{re} is used to get the foreground part of the input image for loss computation. In Eq. 6, we use con_{sum}, which is the sum of the detection confidence of all keypoints, to distinguish different training samples. This is because we think that low-confidence samples correspond to ambiguous texture confidence, e.g., the detection confidence of an occluded hand is usually low. The photometric consistency loss E_{photo} is the weighted sum of E_{pixel} and E_{SSIM} by a weighting factor w_{SSIM}.

$$E_{photo} = E_{pixel} + w_{SSIM}E_{SSIM} \tag{8}$$

Statistical Regularization. During training, to make the results plausible, we introduce regularization terms, including shape regularization E_β, texture regularization E_C, scale regularization E_s, and 3D joints regularization E_J. The shape regularization term is defined as $E_\beta = \| \beta - \bar{\beta} \|$ to encourage the estimated hand model shape β to be close to the average shape $\bar{\beta} = \vec{0} \in \mathbb{R}^{10}$. The texture regularization E_C is used to penalize outlier RGB values. The scale regularization term E_s is used to ensure the output hand is of appropriate size, so as to help determine the depth of the output in this monocular 3D reconstruction task. As for the regularization constraints on skeleton E_J, we define feasible range for each rotation angle a_i (as shown in Fig. 3B) and penalize those who exceed the feasible threshold. We provide more details about E_C,E_s and E_J in the Appendix.

The statistical regularization E_{regu} is the weighted sum of E_β, E_C, E_s and E_J with weighting factors w_C, w_s and w_J:

$$E_{regu} = E_\beta + w_C E_C + w_s E_s + w_J E_J \tag{9}$$

3.4.2 2D-3D Consistency

Losses of the 2D Branch. For the 2D keypoint estimator, we use a joint location loss as Eq. 3 with replacing the projected 2D joint j_i^{pro} by estimated 2D joint j_i^{2d}:

$$E_{2d} = \frac{1}{k}\sum_{i=1}^{k} con_i \mathcal{L}_{SmoothL1}(j_i^{de}, j_i^{2d}) \tag{10}$$

2D-3D Consistency Loss. Since outputs of the 2D branch and the 3D branch are intended to represent the same hand in different spaces, they should be consistent when they are transferred to the same domain. Through this consistency, structural information contained in the 3D reconstruction network can be introduced into the 2D keypoint estimator, and meanwhile estimated 2D keypoints can provide joint-wise geometric cues for 3D hand reconstruction. To this end, we propose a novel 2D-3D consistency loss to link per projected 2D joint j_i^{pro} with its corresponding estimated 2D joint j_i^{2d}:

$$E_{con} = \frac{1}{k}\sum_{i=1}^{k} \mathcal{L}_{SmoothL1}(j_i^{pro}, j_i^{2d}) \tag{11}$$

4. Experiments

In this section, we first present datasets and evaluation metrics (Section 4.1), and implementation details (Section 4.2). Then, we show the performance of our method and conduct comprehensive analysis (Section 4.3 and 4.4).

4.1. Datasets and evaluation metrics

We evaluated our method on two challenging real datasets, both of which evaluate 3D joints and 3D meshes. The results are reported results through online submission systems[3,4].

FreiHAND. The FreiHAND dataset is a large-scale real-world dataset,which contains 32,560 training samples and 3,960 test samples. For each training sample, one real RGB image and extra three images with different synthetic backgrounds are provided. Part of the sample is a hand grabbing an object, but it does not provide any annotations for the foreground object, which poses additional challenges.

HO-3D. The HO-3D dataset collects color images of a hand interacting with an object. The dataset is made of 68 sequences, totaling 77,558 frames of 10 users manipulating one among 10 different objects. The training set contains 66,034 images and the test set contains 11,524 images. The objects in this dataset are larger than that in FreiHAND, thus resulting in larger occlusions to hands.

Evaluation Metrics. We evaluate 3D hand reconstruction by evaluating 3D joints and 3D meshes. For 3D joints, we report the **mean per joint position error** (MPJPE) in the Euclidean space for all joints on all test frames in *cm* and the **area under the curve** (AUC) of the PCK $\mathrm{AUC_J}$. Here, the PCK refers to the percentage of correct keypoints, is plotted using 100 equally spaced thresholds between 0*mm* to 50*mm*. For 3D meshes, we report the **mean per vertex position error** (MPVPE) in the Euclidean space for all joints on all test frames in *cm* and the AUC of the percentage of correct vertex $\mathrm{AUC_V}$. We also compare the F-score [28] which is the harmonic mean of recall and precision for a given distance threshold. We report distance threshold at 5*mm* and 15*mm* and report F-score of mesh vertices at 5*mm* and 15*mm* by $\mathrm{F_5}$ and $\mathrm{F_{15}}$. Following the previous works [18, 53], we compare aligned prediction results with Procrustes alignment, and all 3D results are evaluated by the online evaluation system. For 2D joints, we report the MPJPE in *pixel* and the curve plot of **fraction of joints within distance**.

4.2. Implementation

Pytorch [34] is used for implementation. For the 3D reconstruction network, the EfficientNet-b0 [39] is pre-trained on the ImageNet dataset. The 2D keypoint estimator along with the 2D-3D consistency loss is optionally used. If we train the whole network with the 2D keypoint estimator, a stage-wise training scheme is used. We train the 2D keypoint estimator and 3D reconstruction network by 90 epochs separately, where E_{3d} and E_{2d} are used, respectively. The initial learning rate is 10^{-3} and reduced by a factor of 2 after every 30 epochs. Then we finetune the whole network with E by 60 epochs with the learning rate initialized to 2.5×10^{-4} and reduced by a factor of 3 after every 20 epochs. We use Adam [27] to optimize the network weights with a batch size of 64. We train our model on two NVIDIA Tesla V100 GPUs, which takes around 36 hours for training on FreiHAND. We provide more details in the Appendix.

4.3. Comparison with State-of-the-art Methods

We give comparison on FreiHAND with four recent model-based fully-supervised methods [7, 20, 35, 53] and a state-of-the-art weakly-supervised method [37] in Table 2. Note that [30] is not included here since it designs an advanced "image-to-lixel" prediction instead of directly regress MANO parameters. Our approach focuses on providing a self-supervised framework with lightweight components, where the hand regression scheme is still affected by highly non-linear mapping. Therefore, we make a fairer comparison with popular model-based methods [7, 20, 35, 53] to demonstrate the performance of this self-supervised approach. Without using any annotation, our approach outperforms [20, 53] on all evaluation metrics and achieves comparable performance to [7, 35]. [37] only outputs 3D pose, and its pose performance is slightly better than our results on FreiHAND test set but with much

[3] https://competitions.codalab.org/competitions/21238
[4] https://competitions.codalab.org/competitions/22485

Supervision	Method	$AUC_J\uparrow$	MPJPE↓	$AUC_V\uparrow$	MPVPE↓	$F_5\uparrow$	$F_{15}\uparrow$
3D	[53](2019)	0.35	3.50	0.74	1.32	0.43	0.90
	[20](2019)	0.74	1.33	0.74	1.33	0.43	0.91
	[7](2019)	**0.78**	**1.10**	**0.78**	**1.09**	**0.52**	**0.93**
	[35](2020)	**0.78**	1.11	**0.78**	1.10	0.51	**0.93**
2D	[37](2020)*	**0.78**	1.13	-	-	-	-
-	Ours	0.77	1.18	0.77	1.19	0.48	0.92

Table 2: Comparison of main results on the FreiHAND test set. The performance of our self-supervised method is comparable to the recent fully-supervised and weakly-supervised methods. [37]* also uses synthetic training data with 3D supervision.

Supervision	Method	$AUC_J\uparrow$	MPJPE↓	$AUC_V\uparrow$	MPVPE↓	$F_5\uparrow$	$F_{15}\uparrow$
3D	[20](2019)	-	-	-	1.30	0.42	0.90
	[18](2020)	-	-	-	**1.06**	**0.51**	**0.94**
	[19](2020)	**0.773**	**1.11**	0.773	1.14	0.43	0.93
-	Ours	**0.773**	1.14	**0.777**	1.12	0.45	0.93

Table 3: Comparison of main results on the HO-3D test set. Without using any object information and hand annotation, our hand pose and shape estimation method performs comparably with recent fully-supervised methods.

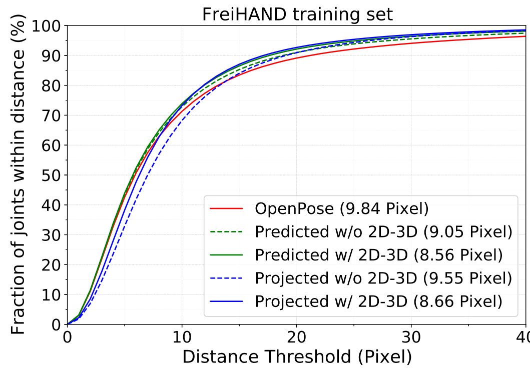

Figure 4: A comparison of 2D keypoint sets used or outputted at the training stage on FreiHAND. The fraction of joints within distance is plotted. The average 2D distances in *pixel* are shown in the legend. Refer to Section 4.4.2 for details.

more training data used including RHD dataset [52] (with 40,000+ synthetic images and 3D annotations) as well as 2D ground truth annotation of the FreiHAND.

In the hand-object interaction scenario, we compare with three recent fully-supervised methods on HO-3D in Table 3. Compared to the hand branch of [20], our self-supervised results show higher mesh reconstruction performance where we get a 14% reduction in MPVPE. Compared with [19], which is a fully-supervised joint hand-object pose estimation method, our approach obtains comparable joints and shape estimation results. [18] gets slightly better shape estimation results than ours, which may be due to it uses multi-frame joint hand-object pose refinement and mesh supervision.

In Fig. 5, we show 2D keypoint detection from OpenPose [9] and our hand reconstruction results of difficult samples. We also compare the reconstruction results with MANO-CNN, which directly estimates MANO parameters with a CNN [53], but we modify its backbone to be the same as ours. Our results are more accurate and additionally with texture.

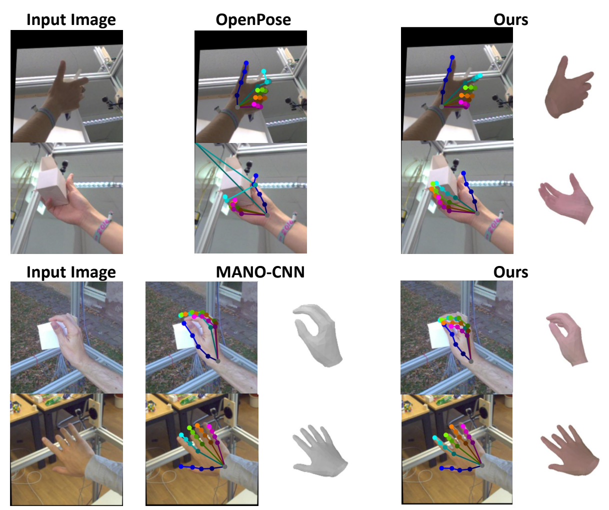

Figure 5: Qualitative comparison to OpenPose [9] and MANO-CNN on the FreiHAND test set. For OpenPose, we visualize detected 2D keypoints. For our method and MANO-CNN, we visualize projected 2D keypoints and 3D mesh.

4.4. Ablation Study

4.4.1 Effect of Each Component

As presented in Table 4, we give evaluation results on FreiHAND of settings with different components along with corresponding loss terms used in the network. The baseline only uses the 3D branch with E_{loc} and E_{regu}, then we add E_{ori} which helps the MPJPE and MPVPE decrease by 19.5%. After adding the 2D branch with E_{2d} and the 2D-3D consistency constrain E_{con}, the MPJPE and MPVPE further reduce by 4%. The E_{photo} slightly improves the pose and shape estimation results.

4.4.2 Comparison of Different 2D Keypoint Sets

In our approach, there are three sets of 2D keypoints, including detected keypoints J^{de}, estimated 2D keypoints J^{2d}, and output projected keypoints J^{pro}, where J^{de} is used as supervision terms while J^{2d} and J^{pro} are output items. In our setting, we use multiple 2D representations to boost the final 3D hand reconstruction, so we do not advocate the novelty of 2D hand estimation, but compare 2D accuracy in the training set to demonstrate the effect of learning from noisy supervision and the benefits of the proposed 2D-3D consistency.

Although we use OpenPose outputs as the keypoint supervision source (see ***OpenPose*** in Fig. 4), we get lower overall 2D MPJPE when we pre-train the 2D and 3D branches separately (see ***Predicted w/o 2D-3D*** and ***Pro-***

Losses				MPJPE↓	MPVPE↓	AUC_J↑	AUC_V↑	F_5↑	F_{15}↑
E_{loc}, E_{regu}	E_{ori}	E_{2d}, E_{con}	E_{photo}						
✓				1.54	1.58	0.696	0.687	0.387	0.852
✓	✓			1.24	1.26	0.754	0.750	0.457	0.903
✓	✓	✓		1.19	1.20	0.764	0.763	0.479	0.915
✓	✓	✓	✓	**1.18**	**1.19**	**0.766**	**0.765**	**0.483**	**0.917**

Table 4: Ablation studies for different losses used in our method on the FreiHAND testing set. Refer to Section 4.4.1 for details.

Dataset	Method	AUC_J↑	AUC_V↑	F_5↑	F_{15}↑
FreiHAND	WSL	0.730	0.725	0.42	0.89
	SSL	**0.766**	**0.765**	**0.48**	**0.92**
HO-3D	WSL	0.765	0.769	0.44	**0.93**
	SSL	**0.773**	**0.777**	**0.45**	**0.93**

Table 5: Comparison of self-supervised results and weakly-supervised results. Refer to Section 4.4.3 for details.

jected w/o 2D-3D in Fig. 4). After finetuning these two branches with 2D-3D consistency, we find both of them gain additional benefits. After the finetuning, the 2D branch (***Predicted w/ 2D-3D***) gains 5.4% reduction in 2D MPJPE and the 3D branch (***Projected w/ 2D-3D***) gains 9.3% reduction in 2D MPJPE. From the curves, we can see that 2D keypoint estimation (including OpenPose and our 2D branch) gets higher accuracy in small distance thresholds while the regression-based methods (***Projected w/o 2D-3D***) get higher accuracy with larger distance threshold. From the curves, the proposed 2D-3D consistency can improve the 3D branch in all distance thresholds, which verifies the rationality of our network design.

4.4.3 Comparison with GT 2D Supervision

We compare the weak-supervised (WSL) scheme using ground truth annotations with our self-supervised (SSL) approach to investigate the ability of our method to handle noisy supervision sources. Both settings use the same network structure and implementation, and WSL uses the ground truth 2D keypoint annotations whose keypoint confidences are set to be the same. As shown in Table 5, our SSL approach has better performance than WSL settings on both datasets. We think this is because the detection confidence information is embedded into the proposed loss functions, which helps the network discriminate different accuracy in the noisy samples. In addition, we find that the SSL method outperforms the WSL method in a smaller amplitude on HO-3D (by 1.0%) than that on FreiHand (by 4.9%). We think this is because the HO-3D contains more occluded hands, resulting in poor 2D detection results. Therefore, we conclude that noisy 2D keypoints can supervise shape learning for the hand reconstruction task, while the quality of the unlabeled image also has a certain impact.

5. Discussion

While our method results in accurate and vivid hand reconstruction in many challenging scenarios (e.g., hand-object interaction, self-occlusion), we also observe failure cases as shown in Fig. 6. The reconstruction accuracy is lower in the extreme pose, severe occlusion, and extreme viewpoint, partly due to poor supervision from single-view 2D keypoint detection. The texture modeling under complex skin reflections is inaccurate, which may be due to the fact that it is difficult for us to accurately simulate com-

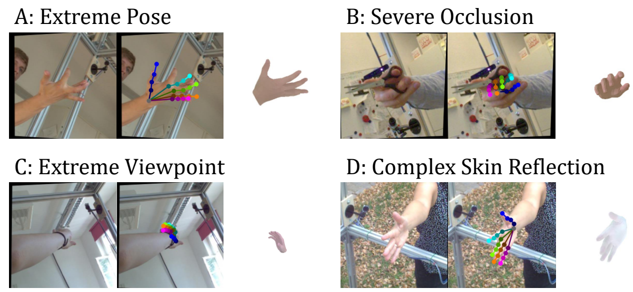

Figure 6: Failure cases. We show input image, projected joints, and 3D reconstruction. See Section 5 for details.

plex skin reflection using a simple illumination representation and a coarse hand mesh. As shown in the last line of Table 4, the texture modeling cannot bring a marked improvement to the shape reconstruction, which is also the case in [35]. This may be because the hand model [36] is not meticulous enough, and the skin reflection simulation is not accurate.

6. Conclusion

We have presented a self-supervised 3D hand reconstruction network that can be trained from a collection of unlabeled hand images. The network encodes the input image into a set of meaningful semantic parameters that represent hand pose, shape, texture, illumination, and the camera viewpoint, respectively. These parameters can be decoded into a textured 3D hand mesh as well a set of 3D joints, and in turn, 3D mesh and joints can be projected into 2D image space, which enables our network to be end-to-end learned. Our network performs well under noisy supervision sources from 2D hand keypoint detection while is able to obtain accurate 3D hand reconstruction from a single-view hand image. Experimental results show that our method achieves comparable performance with state-of-the-art fully-supervised methods. As for the future study, it is possible to extend the parametric hand mesh to other representations (e.g., signed distance function) for more detailed hand surface representation. We also believe that more accurate skin reflection modeling can help hand reconstruction with higher fidelity.

Acknowledgement: This work is supported by the National Key Research and Development Program of China 2018YFB2100603, the Fundamental Research Funds for the Central Universities No.2042020KF0016, and the National Science Foundation Grant CNS-1951952.

References

[1] Anil Armagan, Guillermo Garcia-Hernando, Seungryul Baek, Shreyas Hampali, Mahdi Rad, Zhaohui Zhang, Shipeng Xie, MingXiu Chen, Boshen Zhang, Fu Xiong, et al. Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3d hand pose estimation under hand-object interaction. In *European Conference on Computer Vision*, 2020.

[2] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *Conference on Computer Vision and Pattern Recognition*, 2003.

[3] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-supervised domain adaptation via gan and mesh model for estimating 3d hand poses interacting objects. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[5] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision*, 2012.

[6] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[7] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[8] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *European Conference on Computer Vision*, 2018.

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[10] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *International Conference on Computer Vision*, 2019.

[11] Yujin Chen, Zhigang Tu, Di Kang, Ruizhi Chen, Linchao Bao, Zhengyou Zhang, and Junsong Yuan. Joint hand-object 3d reconstruction from a single image with cross-branch feature fusion. *arXiv preprint arXiv:2006.15561*, 2020.

[12] Yajing Chen, Fanzi Wu, Zeyu Wang, Yibing Song, Yonggen Ling, and Linchao Bao. Self-supervised learning of detailed 3d face reconstruction. *IEEE Transactions on Image Processing*, 29:8696–8705, 2020.

[13] Martin de La Gorce, David J Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[14] Martin de La Gorce, Nikos Paragios, and David J Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[15] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[16] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[17] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *European Conference on Computer Vision*, 2020.

[18] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[19] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[20] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[21] Markus Höll, Markus Oberweger, Clemens Arth, and Vincent Lepetit. Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2018.

[22] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In *European Conference on Computer Vision*, 2020.

[23] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[24] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *European Conference on Computer Vision*, 2018.

[25] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[26] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2014.

[28] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017.

[29] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[30] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, 2020.

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, 2016.

[32] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *Winter Conference on Applications of Computer Vision*, 2018.

[33] Maria Parelli, Katerina Papadimitriou, Gerasimos Potamianos, Georgios Pavlakos, and Petros Maragos. Exploiting 3d hand pose estimation in deep learning-based sign language recognition from rgb videos. In *European Conference on Computer Vision*. Springer, 2020.

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Neural Information Processing Systems Workshops*, 2017.

[35] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. Parametric hand texture model for 3d hand reconstruction and personalization. In *European Conference on Computer Vision*. Springer, 2020.

[36] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 2017.

[37] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *European Conference on Computer Vision*, 2020.

[38] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *European Conference on Computer Vision*, 2018.

[39] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

[40] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[41] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *International Conference on Computer Vision Workshops*, 2017.

[42] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics*, 2016.

[43] Zhigang Tu, Hongyan Li, Dejun Zhang, Justin Dauwels, Baoxin Li, and Junsong Yuan. Action-stage emphasized spatiotemporal vlad for video action recognition. *IEEE Transactions on Image Processing*, 28(6):2799–2812, 2019.

[44] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3d hand pose estimation through training by fitting. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[45] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. RGB2Hands: Real-Time Tracking of 3D Hand Interactions from Monocular RGB Video. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 2020.

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[47] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[48] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. Seqhand: Rgb-sequence-based 3d hand pose and shape estimation. In *European Conference on Computer Vision*, 2020.

[49] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[50] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *International Conference on Computer Vision*, 2019.

[51] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[52] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *International Conference on Computer Vision*, 2017.

[53] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *International Conference on Computer Vision*, 2019.