Consistent 3D Hand Reconstruction in Video via Self-Supervised Learning

Zhigang Tu, *Member, IEEE,* Zhisheng Huang, Yujin Chen, Di Kang, Linchao Bao, *Member, IEEE,* Bisheng Yang, *Senior Member, IEEE,* and Junsong Yuan, *Fellow, IEEE*

Abstract—We present a method for reconstructing accurate and consistent 3D hands from a monocular video. We observe that the detected 2D hand keypoints and the image texture provide important cues about the geometry and texture of the 3D hand, which can reduce or even eliminate the requirement on 3D hand annotation. Accordingly, in this work, we propose $\mathrm{S^2HAND}$, a self-supervised 3D hand reconstruction model, that can jointly estimate pose, shape, texture, and the camera viewpoint from a single RGB input through the supervision of easily accessible 2D detected keypoints. We leverage the continuous hand motion information contained in the unlabeled video data and explore $\mathrm{S^2HAND(V)}$, which uses a set of weights shared $\mathrm{S^2HAND}$ to process each frame and exploits additional motion, texture, and shape consistency constrains to obtain more accurate hand poses, and more consistent shapes and textures. Experiments on benchmark datasets demonstrate that our self-supervised method produces comparable hand reconstruction performance compared with the recent full-supervised methods in single-frame as input setup, and notably improves the reconstruction accuracy and consistency when using the video training data.

Index Terms—hand pose estimation, 3D hand reconstruction, video analysis, self-supervision

1 Introduction

Hands play a central role in the interaction between humans and the environment, from physical contact and grasping to daily communications via hand gesture. Learning 3D hand reconstruction is the preliminary step for many computer vision applications such as augmented reality [1], sign language translation [2], [3], action recognition [4], [5], and human-computer interaction [6], [7], [8]. However, due to diverse hand configurations and interaction with the environment, 3D hand reconstruction remains a challenging problem, especially when the task relies on monocular data as input.

Compared with multi-view images [9], [10], [11], [12] and depth maps [13], [14], [15], [16], [17], the monocular hand images are more common in practice. In recent years, we have witnessed many efforts in recovering 3D shapes of human hands from single-view RGB images. For example, [18], [19], [20], [21], [22] were proposed to predict 3D hand pose from an RGB image. However, they only represent the 3D hand through sparse joints, and ignore the 3D shape information, which are required for some applications such as grasping objects with virtual hands [6]. To better capture the surface information of the hand, previous studies predict the triangle mesh either via regressing per-vertex coordinates [23], [24] or by deforming a parametric hand model [25], [26]. Outputting such high-dimensional representations from 2D input is challenging for neural networks to learn. As a result, the training process relies heavily on 3D hand 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 recent work try to explore parametric texture estimation in a learning-based hand recovery system [27], while most previous work of 3D hand reconstruction do not consider texture modeling [27].

One of our key observations is that the 2D cues in the

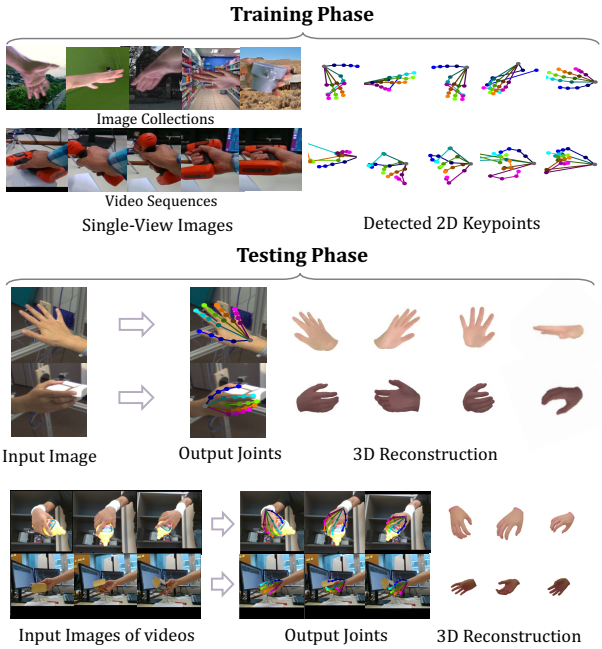


Fig. 1: Given a collection of unlabeled hand images or videos, we learn a 3D hand reconstruction network in a self-supervised manner. Top: the training uses unlabeled hand images from image collections or video sequences and their corresponding noisy detected 2D keypoints. Bottom: our model outputs accurate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

- *Z. Tu, Z. Huang, B. Yang are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (email: {tuzhigang, hzs5230, bshyang}@whu.edu.cn). Z. Tu and Z. Huang contributed equally and they are co-first authors.*
- *Y. Chen is with Technical University of Munich. Work done at Wuhan University. Correspondence to: Y. Chen (email: yujin.chen@tum.de)*
- *D. Kang and L. Bao are with Tencent AI Lab.*
- *J. Yuan is with the Computer Science and Engineering Department, University at Buffalo, Buffalo, NY 14228, USA. (email: jsyuan@buffalo.edu).*

hand image are quite informative to reconstruct the 3D hand model in the real world. The 2D hand keypoints contain rich structural information, and the 2D image contains abundant texture and shape information. Both are important for reducing the use of expensive 3D annotations but have not been fully investigated. Leveraging these cues, we could directly use 2D annotations and the input image to learn the geometry and texture representations without relying on 3D annotations [21]. However, it is still labor-consuming to annotate 2D hand keypoints per image. To completely save the manual annotation, we propose to extract 2D hand keypoints as well as geometric representations from the unlabeled hand image to help the shape reconstruction and use the texture information contained in the input image to help the texture modeling.

Additionally, video sequences contain rich hand motion and more comprehensive appearance information. Usually, a frame-wise fully-supervised hand reconstruction model does not take these information into serious consideration since 3D annotations already provide a strong supervision. As a result, it is more difficult for a frame-wise model to produce consistent results from video frames compared to sequence-wise models, since no temporal information is utilized. Thereby, we propose to penalize the inconsistency of the output hand reconstructions from consecutive observations of the same hand. In this way, motion prior in video is distilled in the frame-wise model to help reconstruct more accurate hand for every single frame. Notably, the constraints on the sequence output are also employed in a self-supervised manner.

Driven by the above observations, this work aims to train an accurate 3D hand reconstruction network using only the supervision signals obtained from the input images or video sequences while eliminating manual annotations of the training images. To this end, we use an off-the-shelf 2D keypoint detector [28] to generate some noisy 2D keypoints, and supervise the hand reconstruction by these noisy detected 2D keypoints and the input image. Although our reconstruction network relies on the predefined keypoint detector, we call it a self-supervised network, following the naming convention in the face reconstruction literature [29], [30] as only the self-annotation is provided to the training data. Further, we leverage the self-supervision signal embedded in the video sequence to help the network produce more accurate and temporally more coherent hand reconstructions. To learn in a self-supervised manner, there are several issues to be addressed. First, how to efficiently use joint-wise 2D keypoints to supervise the ill-posed monocular 3D hand reconstruction? Second, how to handle noise in the 2D detection output since our setting is without utilizing any ground truth annotation? Third, is it possible to make use of the continuous information contained in video sequences to encourage smoothness and consistency of reconstructed hands in a frame-wise model?

To address the first issue, a model-based autoencoder is learned to estimate 3D joints and shape, where the output 3D joints are projected into 2D image space and forced to align with the detected keypoints during training. However, if we only align keypoints in image space, invalid hand pose often occurs. This may be caused by an invalid 3D hand configuration which is still compatible with the projected 2D keypoints. Furthermore, 2D keypoints cannot reduce the scale ambiguity of the predicted 3D hand. Thus, we propose to learn 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.

To address the third issue, we decompose the hand motion into the joint rotations and ensure smooth rotations of hand joints between frames by conforming to a quaternion-based representation. Furthermore, a novel quaternion loss function is proposed to allow all possible rotation speeds. Besides motion consistency, hand appearance is another main concern. A texture and shape (T&S) consistency loss function is introduced to regularize the coherence of the output hand texture and shape.

In brief, we present a self-supervised 3D hand reconstruction model $\mathrm{S^2HAND}$ and its advanced $\mathrm{S^2HAND}(V)$. The models enable us to train neural networks that can predict 3D pose, shape, texture and camera viewpoint from images without any ground truth annotation of training images, except that we use the outputs from a 2D keypoint detector (see Fig. 1). Notably, $\mathrm{S^2HAND(V)}$ is able to extract informative supervision from unannotated videos to help learn a better frame-wise model. In order to achieve this, $\mathrm{S^2HAND(V)}$ inputs the sequential data to multiple weight-shared $\mathrm{S^2HAND}$ models and employs proposed constraints on the sequential output at the training stage.

The advantage of our proposed methods are summarized as follows:

- We present the first self-supervised 3D hand reconstruction models, which accurately output 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 constructed.
- We introduce a hand texture estimation module to learn vivid hand texture via 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.

This work is an extension of our conference paper [31]. The new contributions include:

- We extend our $\mathrm{S^2HAND}$ model to the $\mathrm{S^2HAND(V)}$ model, which further exploits the self-supervision signals embedded in video sequences. The improvement in accuracy and smoothness is 3.5% and 3.1%, respectively.
- We present a quaternion loss function, which is based on an explored motion-aware joints rotation repre-

sentation, to help learn smooth hand motion. Experiments demonstrate its significant advantage over the similar methods in both accuracy and smoothness.
- We propose a texture and shape consistency regularization term to encourage coherent shape and texture reconstruction.
- We illustrate that utilizing extra in-the-wild unlabeled training data can further boost the performance of our model.

2 Related Work

Hand Pose and Shape Estimation. Researchers have developed a lot of different methods in hand pose and shape estimation, such as regression-based method [22], [32], [33], [34], [35] and model-based method [36], [37], [38], [39]. Comparing to hand pose which is represented by 3D coordinates of hand joints alone, hand mesh contains more detailed shape information and recently has become the focus in the research community. Several methods utilize the hand mesh topology to directly output 3D mesh vertices. E.g. [40], [41], [42] use the spiral convolution to recover hand mesh and [34], [43], [44] use the graph convolution to output mesh vertices. Although these methods introduce as few priors as possible, they require large amounts of annotated data for training. In this self-supervised work, we make use of the priors contained in the MANO hand model [45], where MANO can map pose and shape parameters to a triangle mesh [26], [46], [47], [48], to reduce reliance on the labeled training data.

Because the parametric model contains abundant structure priors of human hands, recent works integrate hand model as a differentiable layer in neural networks [25], [26], [46], [48], [49], [50], [51], [52]. Among them, [49], [51], [52] output a set of intermediate estimations, like segmentation mask and 2D keypoints, and then map 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 utilize it only during training to facilitate 3D reconstruction. More generally, recent methods [25], [26], [46], [48], [50] directly adopt an autoencoder that couples an image feature encoding stage with a model-based decoding stage. Unlike [25], [26], 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. [19] uses a depth map to perform additional weak supervision to strengthen 2D supervision. [21] proposes the biomechanical constraints to help the network output feasible 3D hand configurations. [53] detects 2D hand keypoints and directly fits a hand model to the 2D detection. [24] gathers a large-scale dataset through an automated data collection method similar to [53] and then applies 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 imposing constraints on the skeleton as [21]. In contrast to [19], [24], our method is designed to verify the feasibility of (noisy) 2D supervision and avoids introducing any extra 2.5D or 3D data.

Self-supervised 3D Reconstruction. Recently, there are methods that propose to learn 3D geometry from the monocular image only. For example, [54] presents 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. [55] removes keypoints from the supervision signals, but it uses ground truth 2D silhouette as supervision and only tackles categories with small intra-class shape differences, such as birds, shoes, and cars. [56] exploits a self-supervised contrastive learning for hand pose estimation, but only the encoder is pretrained in the self-supervised manner. [57] designs a self-supervised module to overcome inconsistency between the 2D and 3D hand pose, but they only consider the sparse joint keypoints. [58] explores a depth-based self-supervised 3D hand pose estimation method, but the depth image provides much stronger evidence and supervision than the RGB image. Recently, [29], [30], [59] exploits a self-supervised face reconstruction method with the usage of 3D morphable model of face (3DMM) [60] and 2D landmarks detection. Our approach is similar to them, but the hand is 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. [61], [62] exploit shading and texture information to handle the self-occlusion problem in the hand tracking system. Recently, [27] 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.

Motion Learning from Sequence Data for 3D Hand Estimation. To leverage motion information contained in sequence data, several methods have been proposed in hand pose estimation. [25] uses the photometric consistency between neighboring frames of sparsely annotated RGB videos. [63] presents a graph-based method to exploit spatial and temporal relationship for sequence pose estimation. [64] utilizes the temporal information through bidirectional inferences. [36], [65], [66] design a temporal consistency loss for motion smoothness. However, these methods either are specialized for motion generation or only impose a weak regularization for motion smoothness.

There exists no approach to capture hand motion dynamics fundamentally, leading to limited benefits can be gained from modeling motion. In this work, we aim to exploit self-supervised information from hand motion dynamics. Unlike most of the previous approaches [67], [68], [69], [70] which adopt recurrent or graph-based network structure to learn hand motion in a sequence-to-sequence manner, we instead use a motion-related loss function to help our frame-wise model converges better and bridges the gap with fully-supervised methods.

From the above analysis and comparison, we believe that self-supervised 3D hand reconstruction is feasible and

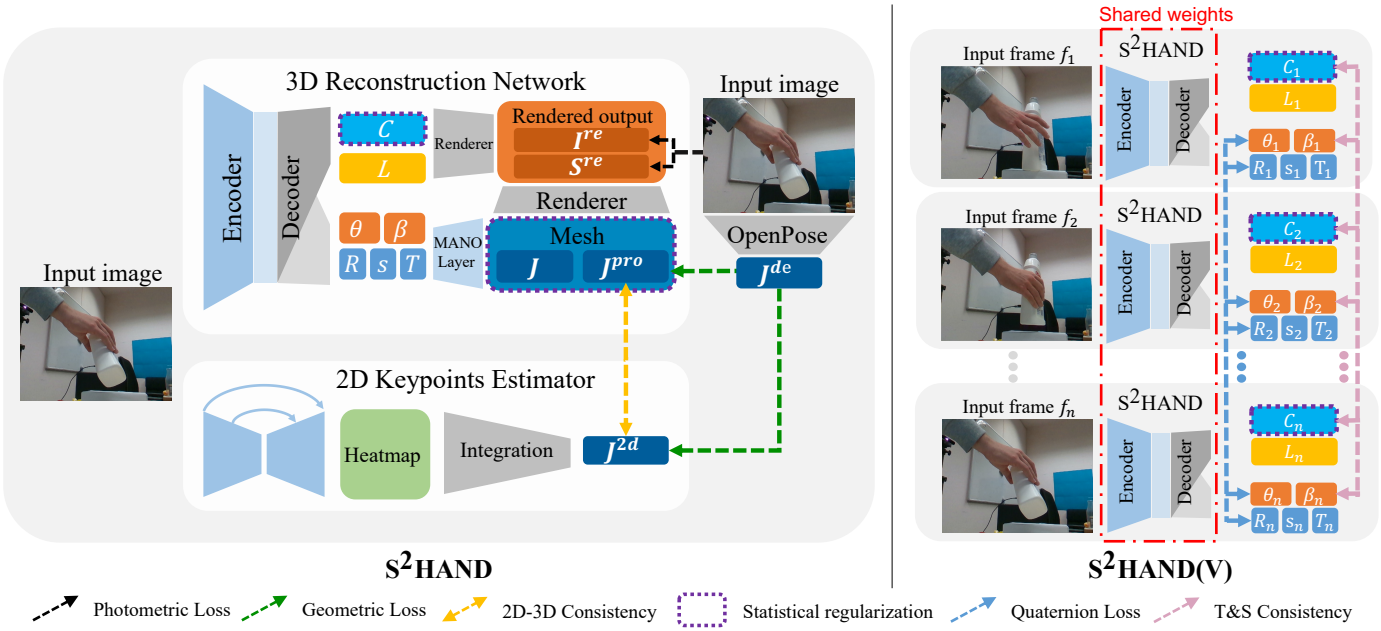


Fig. 2: Overview of the proposed models. The S^{2}HAND(V) on the right learns to reconstruct consistent 3D hands from video sequences without ground truth annotations based on S^{2}HAND. Given an input image, the S^{2}HAND model generates a 3D textured hand with its corresponding multiple 2D representations through a 3D reconstruction network and a 2D keypoints estimator. Effective loss functions and regularization terms are designed for self-supervised network training. Given a video sequence, the S^{2}HAND(V) model produces sequential outputs from several weight-shared S^{2}HAND models with temporal constraints. A quaternion loss and a T&S loss are presented to exploit continuous motion information to promote consistent hand reconstruction. During the inference, only the 3D reconstruction network is utilized and the S^{2}HAND(V) acts just like a specially trained S^{2}HAND due to weight sharing. The symbols used in this figure can be found in Section 3.2 and Section 3.3.

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 model, and prove its effectiveness through extensive experiments.

3 Methodology

3.1 Overview

Our method enables end-to-end learning of accurate and consistent 3D hand reconstruction from video sequences in a self-supervised manner through S^{2}HAND(V) (Section 3.3), which is based on S^{2}HAND (Section 3.2). The overview is illustrated in Fig. 2.

The S^{2}HAND model takes an image as input and generates a textured 3D hand represented by pose, shape and texture, along with corresponding lighting, camera viewpoint (Section 3.2.1 and 3.2.2) and multiple 2D representations in the image space (Section 3.2.3). Some efficient loss functions and regularization terms (Section 3.2.4) are explored to train the network without using ground truth annotations. The S^{2}HAND(V) model takes video sequences as input and produces consistent sequential outputs from multiple S^{2}HAND models where their weights are shared. A quaternion loss (Section 3.3.1) and a T&S consistency loss (Section 3.3.2) are designed to train the network with temporal constraints. We describe the proposed method in detail as below.

3.2 Self-supervised Hand Reconstruction from Image Collections

The S^{2}HAND model learns self-supervised 3D hand reconstruction from image collections via training a 3D hand reconstruction network with the help of a trainable 2D keypoints estimator (See Section 3.2.3).

3.2.1 Deep Hand Encoding

Given an image I that contains a hand, the 3D hand reconstruction network first extracts the feature maps with the EfficientNet-b0 backbone [71], and then transforms them 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: $y = (C, L)$.

3.2.2 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 will 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 $V = \{v_i \in \mathbb{R}^3 | 1 \leq i \leq n\}$ 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 [26] $J = \{j_i \in \mathbb{R}^3 | 1 \leq i \leq k\}$ (as shown in Fig. 3A) 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, where MANO is a low-dimensional parametric model [45].

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 in terms of: $M = sM_0R + T$ and $J = sJ_0R + T$.

Texture and Lighting Representation. We use per-face RGB value of 1538 faces to represent the texture of hand $C = \{c_i \in \mathbb{R}^3 | 1 \leq i \leq n\}$, 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 the lighting conditions [72]. 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 $l^d \in \mathbb{R}^1$, directional light color $l^d_c \in \mathbb{R}^3$, and directional light direction $n^d \in \mathbb{R}^3$ in a unified representation: $L = (l^a, l^a_c, l^d, l^d_c, n^d)$.

3.2.3 2D Hand Representations

A set of estimated 3D joints within the camera space 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 with 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 perspective camera projection Π to project 3D joints into a set of $k = 21$ 2D joints $J^{pro} = \{j^{pro}_i \in \mathbb{R}^2 | 1 \leq i \leq k\}$, 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 the implementation[1] of [72]. 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 can help the projected 2D keypoints retain the structural information, but at the same time gives up the knowledge of per-joint prior. To address this issue, we additionally use a 2D keypoint estimator to directly estimate a set of $k = 21$ independent 2D joints $J^{2d} = \{j^{2d}_i \in \mathbb{R}^2 | 1 \leq i \leq k\}$, 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 [73] along with an integral pose regression [74] is used. Note that the 2D hand pose estimation module is optionally deployed in the training period and is not required during the inference.

3.2.4 Training Objective

Our overall training loss E_{S^2HAND} consists of three parts, i.e. a 3D branch loss E_{3d}, a 2D branch loss E_{2d} and a 2D-3D consistency loss E_{cons}:

$$E_{S^2HAND} = w_{3d}E_{3d} + w_{2d}E_{2d} + w_{cons}E_{cons} \quad (1)$$

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

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

$$E_{3d} = w_{geo}E_{geo} + w_{photo}E_{photo} + w_{regu}E_{regu} \quad (2)$$

Geometric Alignment. We propose a geometric alignment loss E_{geo} based on the detected 2D keypoints which are obtained through an implementation[2] of [28]. The detected 2D keypoints $L = \{(j^{de}_i, con_i) | 1 \leq i \leq k\}$ allocate each keypoint with a 2D position $j^{de}_i \in \mathbb{R}^2$ 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 the keypoints in these two sets to be aligned:

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

$$E_{ori} = \frac{1}{m}\sum_{i=1}^{m} conf^{bone}_i \| \nu^{de}_i - \nu^{pro}_i \|^2_2 \quad (4)$$

Here, a SmoothL1 loss [75] is used in Eq. 3 to make the loss term to be more robust to local adjustment since the detected keypoints are not fit well with the MANO keypoints. In Eq. 4, ν^{de}_i and ν^{pro}_i are the normalized i-th bone vector of the detected 2D joints and the projected 2D joints, respectively, and $conf^{bone}_i$ 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}:

$$E_{geo} = E_{loc} + w_{ori}E_{ori} \quad (5)$$

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 estimated by evaluating the structural similarity between the two images [76]:

$$E_{pixel} = \frac{conf_{sum}}{|S^{re}|} \sum_{(u,v)\in S^{re}} \| I_{u,v} - I^{re}_{u,v} \|_1 \quad (6)$$

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

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 $conf_{sum}$, 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

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

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

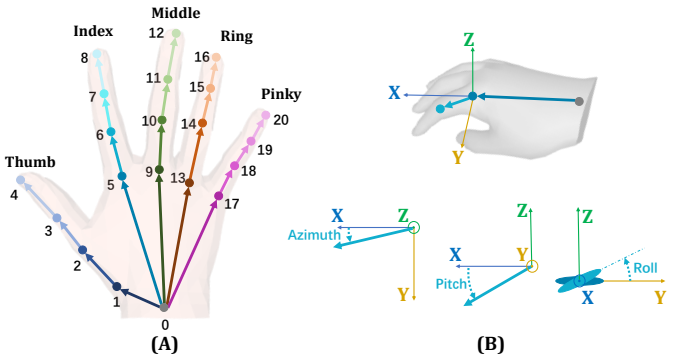


Fig. 3: (A)The joint skeleton structure. (B) A sample of bone rotation angles. The five bones ($\overrightarrow{01}$,$\overrightarrow{05}$,$\overrightarrow{09}$,$\overrightarrow{013}$,$\overrightarrow{017}$) on the palm are fixed. Each finger has 3 bones, and the relative orientation of each bone from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

photometric consistency loss E_{photo} is the weighted sum of E_{pixel} and E_{SSIM} with 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 some regularization terms, including the shape regularization E_β, the texture regularization E_C, the scale regularization E_s, and the 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 has appropriate size, so as to help determining the depth of the output in this monocular 3D reconstruction task. To enforce the regularizations 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. The remaining E_C, E_s and E_J terms follow [31].

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:

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

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

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

3.3 Consistent Self-supervised Hand Reconstruction from Video Sequences

The S^{2}HAND(V) model learns consistent self-supervised 3D hand reconstruction from video sequences via training weight-shared S^{2}HAND models with temporal constraints, including a quaternion loss and a T&S consistency loss.

3.3.1 *Quaternion-based Motion Regularization*

We reformulate hand motion in joint rotation perspective. We choose the unit quaternion [77], [78], [79], [80] as our joint rotation representation, which can represent spatial orientations and rotations of elements in a convenient and efficient way. The unit quaternion associated with a spatial rotation is constructed as:

$$\mathbf{q} = \left(\cos(\frac{\alpha}{2}), \sin(\frac{\alpha}{2})\vec{\mathbf{u}}\right) \tag{12}$$

where α is the rotation angle and $\vec{\mathbf{u}}$ denotes the rotation axis in $\mathbb{R}^3$. Notably, $\mathbf{q}$ can represent both rotation and orientation.

Smooth orientation transition $\mathbf{q}_t$ between initial $\mathbf{q}_0$ joint orientation and final $\mathbf{q}_1$ joint orientation is defined by a unique axis $\vec{\mathbf{v}}$ and corresponding rotation angle γ around the axis. The transition process can be expressed as follows:

$$\mathbf{q}_t = (\mathbf{q}_1 \mathbf{q}_0^{-1})^{\phi(t)} \mathbf{q}_0 = \left(\cos(\frac{\gamma}{2}), \sin(\frac{\gamma}{2})\vec{\mathbf{v}}\right)^{\phi(t)} \mathbf{q}_0 \tag{13}$$

where $\mathbf{q}_0^{-1}$ represents the inverse of $\mathbf{q}_0$, and the product operation here is the Hamilton product. The $\phi(t)$ denotes a monotonically non-decreasing function, ranging from 0 to 1 and controlls the orientation transition from $\mathbf{q}_0$ to $\mathbf{q}_1$. When $\phi(t)$ equals 0 or 1, $\mathbf{q}_t$ will equal to $\mathbf{q}_0$ and $\mathbf{q}_1$ respectively. In order to reduce the computational cost brought by the Hamilton product, we further rewrite Eq. 13 as a linear combination of the two quaternions $\mathbf{q}_0$ and $\mathbf{q}_1$:

$$\mathbf{q}_t = (\mathbf{q}_1 \mathbf{q}_0^{-1})^{\phi(t)} \mathbf{q}_0 = Norm\left[\mu(t)\mathbf{q}_0 + \varepsilon(t)\mathbf{q}_1\right] \tag{14}$$

where $Norm[\cdot]$ denotes normalization to ensure the result is a unit quaternion. $\mu(t)$ and $\varepsilon(t)$ are time-dependent coefficients which are determined by $\phi(t)$. One instance of Eq. 14 is Slerp [79], which is a widely used linear quaternion interpolation method with constant rotation speed, assuming $\phi(t) = t$:

$$\mathbf{q}_t = (\mathbf{q}_1 \mathbf{q}_0^{-1})^t \mathbf{q}_0 = \frac{\sin((1-t)\eta)}{\sin(\eta)} \mathbf{q}_0 + \frac{\sin(t\eta)}{\sin(\eta)} \mathbf{q}_1 \tag{15}$$

where η is the included angle between $\mathbf{q}_0$ and $\mathbf{q}_1$ as two vectors, which can be computed by:

$$\eta = \frac{\mathbf{q}_1 \cdot \mathbf{q}_0}{\|\mathbf{q}_1\| \, \|\mathbf{q}_0\|} \tag{16}$$

where $\cdot$ denotes inner product of two vectors and $\|\cdot\|$ is the magnitude of a vector.

Instead of generating interpolated poses as psuedo-labels with one specific $\phi(t)$ for supervision, we propose

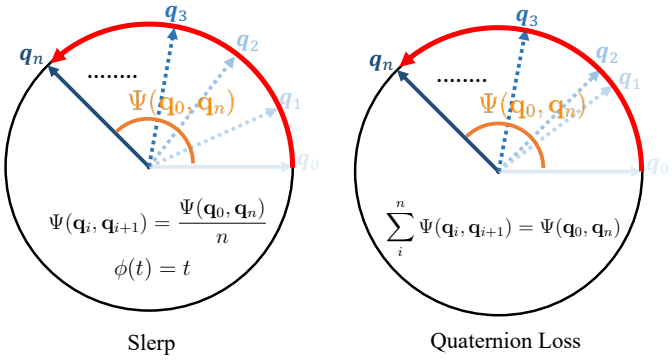


Fig. 4: Comparison between our quaternion loss and Slerp. The circle represents a 2D projective plane of 4D unit quaternion sphere. The red arch denotes the set of quaternion that satisfies Eq. 13, which ensures smooth orientation transition. The equation in each circle represents the corresponding prior. The remaining symbols can be found in Section 3.3. As can be seen, both Slerp and quaternion loss has the prior to make sure Eq. 13 is satisfied. However Slerp has an additional prior $\phi(t) = t$, while our Quaternion loss covers all possible $\phi(t)$, which allows smooth orientation transition at all possible speed.

a quaternion loss function to cover all possible joint rotation speeds as following:

$$E_{quat} = \left\| \sum_{i=1}^{n-1} \Psi(H_i, H_{i+1}) - \Psi(H_1, H_n) \right\| \quad (17)$$

where Ψ is the function to compute the rotation angle γ between two quaternions, and H_i denotes the output hand pose represented in quaternion of frame i. In practice, H_i is the concatenation of i-th pose vector θ_i and i-th rotation R_i to cover all 21 hand joints:

$$H_i = Quaternion(Concatenate[\theta_i, R_i]) \quad (18)$$

where $Quaternion$ denotes the transformation from representation of MANO outputs to quaternion representation and $Concatenate$ denotes the concatenation operation. The comparison between the proposed quaternion loss and Slerp is illustrated in Fig. 4.

To understand the proposed quaternion loss, two points are important. One is that the quaternion interpolation is essentially finding a rotation curve through a fixed rotation axis between two poses, as suggested by Eq. 13. The other is that the rotation angle γ in the quaternion space relates to included angle η in vector space, as indicated by Eq. 14. Specifically:

$$\cos(\frac{\gamma}{2}) = \cos(\eta) \quad (19)$$

which provides an efficient way to compute γ and is derived from:

$$\Re\left(\mathbf{q}_1 \mathbf{q}_0^{-1}\right) = \Re\left(\cos(\frac{\gamma}{2}), \sin(\frac{\gamma}{2})\vec{\mathbf{v}}\right) = \|\mathbf{q}_1\| \, \|\mathbf{q}_0\| \cos(\eta) \quad (20)$$

where $\Re$ represents the real part of a quaternion, $\mathbf{q}_0$, $\mathbf{q}_1$, γ and $\vec{\mathbf{v}}$ are the same as before. $\|\cdot\|$ is the magnitude of a vector. η denotes the included angle between $\mathbf{q}_0$ and $\mathbf{q}_1$ as two vectors. Eq. 20 can be deduced by comparing the inner product and the Hamilton product of the $\mathbf{q}_0$ and $\mathbf{q}_1$.

3.3.2 T&S Consistency Regularization

We introduce a regularization term on texture and shape to consider consistency of hand appearance in videos. Since texture is coupled with light, our T&S loss is formulated as:

$$E_{T\&S} = \sum_{i=1}^{n} \left\| C_i^L - \overline{C^L} \right\| + \sum_{i=1}^{n} \left\| \beta_i - \overline{\beta} \right\| \quad (21)$$

where C_i^L and β_i are the i-th lighted texture and shape of the sequential reconstruction output, $\overline{C_i^L}$ and $\overline{\beta_i}$ are the corresponding average of the sequential output. The lighted texture C_i^L is computed following [72]:

$$C_i^L = (l^a l_c^a + (n^d \cdot n_i) l^d l_c^d) C_i \quad (22)$$

where n_i is the normal direction of C_i in canonical zero pose [45] and the rest are defined in Section 3.2.1.

A low standard deviation of sequential hand appearance reconstruction from video sequences is promoted by this loss function.

3.3.3 Training Objective

Our overall training loss $E_{S^2HAND(V)}$ consists of three parts, including a S^2HAND loss E_{S^2HAND}, a quaternion loss E_{quat} and a $T\&S$ loss $E_{T\&S}$:

$$E_{S^2HAND(V)} = E_{S^2HAND} + w_{quat} E_{quat} + w_{ts} E_{T\&S} \quad (23)$$

where E_{S^2HAND} is the same as that in Section 3.2.4. For E_{quat} and $E_{T\&S}$, please refer to Section 3.3.1 and Section 3.3.2 respectively. The constant weights w_{quat} and w_{ts} are used to balance the three terms.

4 Experiments

4.1 Datasets

We evaluate the proposed methods on three datasets. Two of them (FreiHAND and HO-3D) are challenging realistic datasets, aiming for assessing 3D joints and 3D meshes with hand-object interaction. The results are reported through the online submission systems [3,4]. The remaining one (STB) is a hand-only video dataset. Besides, we adopt another dataset (YT 3D) to provide in-the-wild data.

The FreiHAND dataset [48] 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.

The HO-3D dataset [81] 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. We use this dataset in two cases. In the case of self-supervised hand reconstruction from image collections with S^2HAND, we do not use the sequence information provided by HO-3D and

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

mix all sequences as a image collection. In the case of self-supervised hand reconstruction from video sequences with $\mathrm{S}^2\mathrm{HAND(V)}$, we make use of the sequence information and compose the training batch accordingly. Details can be found in Section 4.3.

The STB dataset [82] is a hand-only dataset, which contains 12 sequences with 18000 frames in total. RGB images along with depth-images, 2D and 3D joint annotations are provided. We follow the splits in [82], using 10 sequences for training and 2 sequences for evaluation. We select this dataset to validate the proposed methods in the hand-only scenario.

The YT 3D dataset [40] contains 116 in-the-wild videos, which is comprised of 102 train videos, 7 validation videos and 7 test videos, along with 47125, 1525 and 1525 hand annotations. We only use this dataset as extra in-the-wild training data in Section 4.5.3 without any annotations. Since the 102 train videos are edited with lots of cutaway and only part of videos are accessible due to copyright issues, we preprocess this dataset by filtering out unavailable videos and discontinuous hand motion frames according to the detected 2D keypoints, yielding 34 train videos containing 21628 frames with detected 2D keypoints.

4.2 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. 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 [83] 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 [48], [81], we compare aligned prediction results with Procrustes alignment, and all 3D results are evaluated by the online evaluation system on FreiHAND and HO-3D. For 2D joints, we report the MPJPE in *pixel* and the curve plot of **fraction of joints within distance**. For smooth hand reconstruction, we report the **acceleration error** (ACC-ERR) and the **acceleration**(ACC) which are first proposed in [84]. ACC-ERR measures average difference between ground truth acceleration and the acceleration of the predicted 3D joints in mm/s^2 while ACC calculates mean acceleration of the predicted 3D joints in mm/s^2. Generally, lower ACC-ERR and ACC indicate smoother sequence predictions. For shape and texture consistency in sequence predictions, we report corresponding **standard deviations (S.D.)**, in which low deviation means coherent and consistent sequence predictions. Specifically, we compute texture S.D. and shape S.D., which are the average of per dimensional S.D. of the lighted textures and shape parameters in sequence predictions respectively.

4.3 Implementation Details

Pytorch [85] is used for implementation. For the 3D reconstruction network, the EfficientNet-b0 [71] 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.

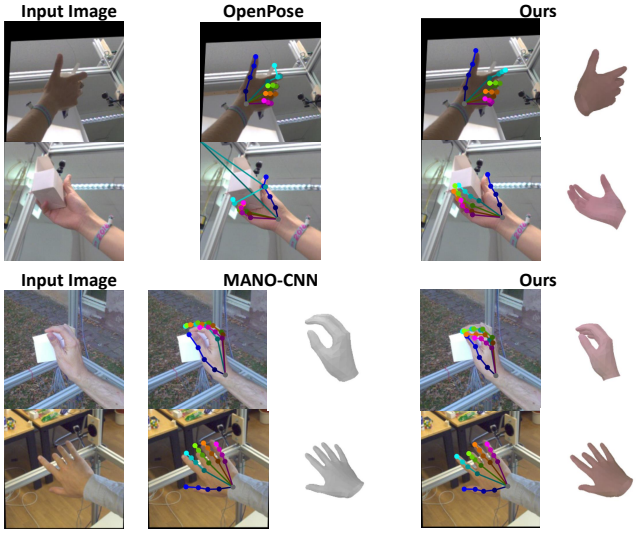


Fig. 5: Qualitative comparison to OpenPose [28] and MANO-CNN on the FreiHAND testing set. For OpenPose, we visualize the detected 2D keypoints. For our method and MANO-CNN, we visualize both the projected 2D keypoints and 3D mesh.

When training $\mathrm{S}^2\mathrm{HAND}$ with image collections, we finetune the whole network with E_{S^2HAND} 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 [86] 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. Otherwise, when training $\mathrm{S}^2\mathrm{HAND(V)}$ with $E_{S^2HAND(V)}$, the input remains 4D (B,C,H,W), but the sampling strategy is different. Specifically, for a training batch, we first randomly sample m sequences from the training sequences, then randomly sample n frames in each of the sampled sequences, finally combine these frames to compose a batch. This results in a batch size of mn. In our experiments, we set m equal to $64//n$, where 64 follows the batch size in training $\mathrm{S}^2\mathrm{HAND}$ and // represents floor division. We find n equals 3 (see Section 4.5.2 for the ablation study) gets the best performance, thus we use it as the default setting. Notice that all the sampling procedures take place during training. For the stability of this sampling strategy, please refer to Section 2 of the supplementary materials. The learning rate, the reducing schedule, and the Adam optimizer are set the same as before. For the weighting factors, we set $w_{3d} = 1$, $w_{2d} = 0.001$, $w_{cons} = 0.0002$, $w_{geo} = 0.001$, $w_{photo} = 0.005$, $w_{quat} = 0.05$, $w_{ts} = 0.01$, $w_{regu} = 0.01$, $w_{ori} = 100$, $w_{SSIM} = 0.2$, $w_C = 0.5$, $w_s = 10$ and $w_J = 10$. Here the weighting factors need to regard different fundamental units of the losses, thus the magnitudes do not strictly imply the importance. For how to select these weights and the weights sensitivity, please refer to the supplementary materials.

TABLE 1: Comparison of main results on the FreiHAND testing set. The performance of our self-supervised method S^{2}HAND is comparable to the recent fully-supervised and weakly-supervised methods. [21]* also uses the synthetic training data with 3D supervision. Note that FreiHAND is not presented with video sequences, which disables learning of S^{2}HAND(V).

Supervision	Method	AUC_J↑	MPJPE↓	AUC_V↑	MPVPE↓	F_5↑	F_{15}↑
3D	[48](2019)	0.35	3.50	0.74	1.32	0.43	0.90
	[26](2019)	0.74	1.33	0.74	1.33	0.43	0.91
	[46](2019)	**0.78**	**1.10**	**0.78**	**1.09**	**0.52**	**0.93**
	[27](2020)	**0.78**	1.11	**0.78**	1.10	0.51	**0.93**
2D	[21](2020)*	**0.78**	1.13	-	-	-	-
-	S^{2}HAND	0.77	1.18	0.77	1.19	0.48	0.92

TABLE 2: Comparison of main results on the HO-3D testing set. Without using any object information and hand annotation, our S^{2}HAND model performs comparable with the recent fully-supervised methods [25]. Further with the temporal constraints, our S^{2}HAND(V) even surpasses [25].

Supervision	Method	AUC_J↑	MPJPE↓	AUC_V↑	MPVPE↓	F_5↑	F_{15}↑
3D	[26](2019)	-	-	-	1.30	0.42	0.90
	[81](2020)	-	-	-	**1.06**	**0.51**	**0.94**
	[25](2020)	0.773	1.11	0.773	1.14	0.43	0.93
-	S^{2}HAND	0.773	1.14	0.777	1.12	0.45	0.93
	S^{2}HAND(V)	**0.780**	**1.10**	**0.781**	1.09	0.46	**0.94**

TABLE 3: Ablation studies on different losses used in our method on the FreiHAND testing set. Refer to Section 4.5.1 for details.

Losses					MPJPE↓	MPVPE↓	AUC_J ↑	AUC_V ↑	F_5 ↑	F_{15} ↑
E_{loc}	E_{regu}	E_{ori}	E_{2d}, E_{con}	E_{photo}						
✓					1.97	2.31	0.611	0.545	0.257	0.763
✓	✓				1.54	1.58	0.696	0.687	0.387	0.852
✓	✓	✓			1.24	1.26	0.754	0.750	0.457	0.903
✓	✓	✓		✓	1.22	1.24	0.759	0.754	0.468	0.909
✓	✓	✓	✓		1.19	1.20	0.764	0.763	0.479	0.915
✓	✓	✓	✓	✓	**1.18**	**1.19**	**0.766**	**0.765**	**0.483**	**0.917**

4.4 Comparison with State-of-the-art Methods

We give comparison on FreiHAND with four recent model-based fully-supervised methods [26], [27], [46], [48] and a state-of-the-art weakly-supervised method [21] in Table 1. Note that [87] is not included here since it designs an advanced "image-to-lixel" prediction instead of directly regressing MANO parameters. Our approach S^{2}HAND 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 [26], [27], [46], [48] to demonstrate the performance of this self-supervised approach. Without using any annotation, our approach S^{2}HAND outperforms [26], [48] on all evaluation metrics and achieves comparable performance to [27], [46]. [21] only outputs 3D pose, and its pose performance is slightly better than our results on FreiHAND test set but with much more training data used including RHD dataset [22] (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 2. Compared to the hand branch of [26], both of our self-supervised models S^{2}HAND and S^{2}HAND(V) show higher mesh reconstruction performance where we get a 14% and 16% reduction in MPVPE respectively. Compared with [25], which is a fully-supervised joint hand-object pose estimation method, our S^{2}HAND obtains comparable joints and shape estimation results, while our S^{2}HAND(V) even surpasses [25]. [81] gets slightly better shape estimation results than ours, probably due to the fact that they first estimate 2D keypoint positions using heatmaps and then fit MANO model to these keypoints. Since [21] utilizes a totally different version of HO-3D that is published with HANDS 2019 Challenge[5], we do not compare with it on HO-3D (not shown in Table 2) as we do on FreiHAND.

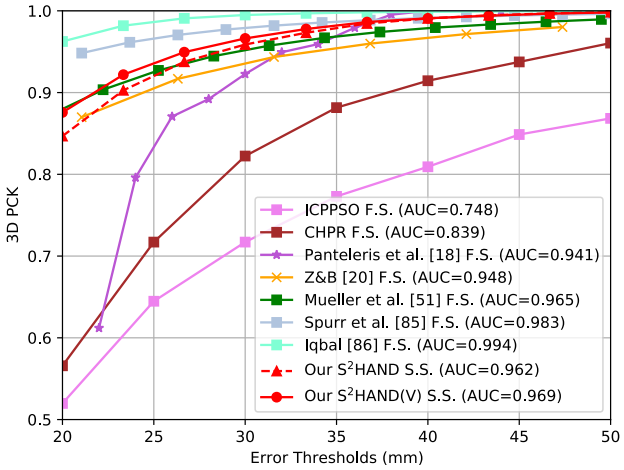


Fig. 6: Comparisons between our proposed self-supervised (S.S.) methods and other SOTA fully-supervised (F.S.) methods on the STB dataset (hand-only scenario).

In the hand-only scenario, we compare with the fully-supervised methods [20], [22], [53], [88], [89] on STB in Fig. 6. Some state-of-the-art fully-supervised methods (e.g. [50] with 0.995 AUC and [90] with 0.997 AUC) are not included in Fig. 6, because their PCK values are not provided and they have similar performance with [20]. The proposed self-supervised method S^{2}HAND outperforms some previous fully-supervised methods [22], [53], [88], and S^{2}HAND(V) further improves the performance of S^{2}HAND. On the other side, there is a certain gap between ours and the recent fully-supervised methods [20], [89] since they have direct

5. https://sites.google.com/view/hands2019/challenge

TABLE 4: Comparison of the accuracy of different motion-related constraints on the HO-3D testing set. Quaternion loss shows its effectiveness over the similar loss functions on modeling smoothness.

Method	AUC_J↑	MPJPE↓	AUC_V↑	MPVPE↓	F_5↑	F_{15}↑
S^2HAND	0.773	1.14	0.777	1.12	0.45	0.93
S^2HAND(V) (w/ Temporal Loss [65])	0.771	1.15	0.773	1.14	0.44	0.93
S^2HAND(V) (w/ Smooth Loss [36])	0.774	1.13	0.776	1.12	0.45	0.93
S^2HAND(V) (w/ Quaternion Loss)	0.779	**1.10**	**0.781**	**1.09**	**0.46**	**0.94**
S^2HAND(V) (w/ Quaternion Loss, T&S Loss)	**0.780**	**1.10**	**0.781**	**1.09**	**0.46**	**0.94**

TABLE 5: Comparison of the smoothoness performance of different motion-related constraints on the HO-3D dataset. Quaternion loss gives the smoothest predictions and is highly in line with ACC and ACC-ERR.

Method	Train set			Test set	
	ACC-ERR↓	ACC↓	Quaternion Loss↓	ACC↓	Quaternion Loss↓
S^2HAND	2.85	2.29	0.014	3.68	0.020
S^2HAND(V) (w/ Temporal Loss [65])	2.57	1.96	0.011	2.87	0.015
S^2HAND(V) (w/ Smooth Loss [36])	2.89	2.29	0.014	2.89	0.015
S^2HAND(V) (w/ Quaternion Loss)	**2.23**	**1.59**	**0.008**	**2.81**	**0.014**

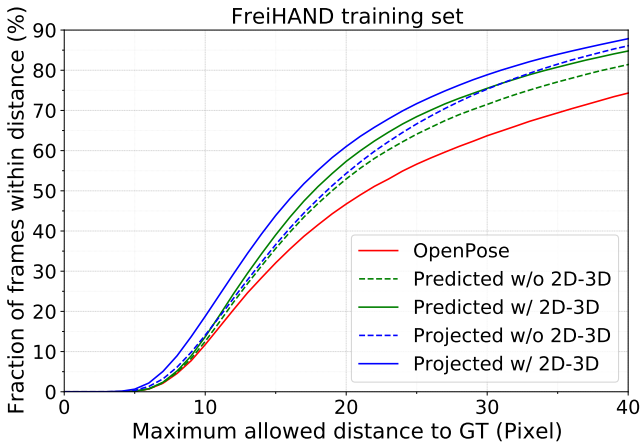


Fig. 7: A comparison of 2D keypoint sets used or outputted at the training stage on FreiHAND. The fraction of frames within the maximum joint distance is plotted. Refer to Section 4.5.1 for details.

3D supervision.

In Fig. 5, we show 2D keypoint detection from OpenPose [28] and our S^2HAND results of difficult samples. We also compare the reconstruction results with MANO-CNN, which directly estimates MANO parameters with a CNN [48], but we modify its backbone to be the same as ours. Our results are more accurate and additionally with texture.

4.5 Self-comparison

In Section 4.5.1 and Section 4.5.2, we conduct extensive self-comparisons to verify the effectiveness of each component of our models S^2HAND and S^2HAND(V). In Section 4.5.3, we explore the effect of absorbing extra in-the-wild video sequences with S^2HAND(V).

4.5.1 Hand Reconstruction from Image Collections

For self-supervised hand reconstruction from image collection with S^2HAND, we conduct ablation studies on FreiHAND, since it is a widely used challenging dataset for hand pose and shape estimation from single RGB images.

First, we give evaluation results on FreiHAND of settings with different components along with corresponding loss terms used in the network in Table 3. 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_{cons}, the MPJPE and MPVPE further reduce by 4%. The E_{photo} slightly improves the pose and shape estimation results.

TABLE 6: Comparison of self-supervised results and weakly-supervised results. Refer to Section 4.5.1 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**

Then, we make 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. 7), we get lower overall 2D MPJPE when we pre-train the 2D and 3D branches separately (see ***Predicted w/o 2D-3D*** and ***Projected w/o 2D-3D*** in Fig. 7). 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.

Last, 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 6, our SSL approach has better performance than WSL settings on both datasets. We think this is

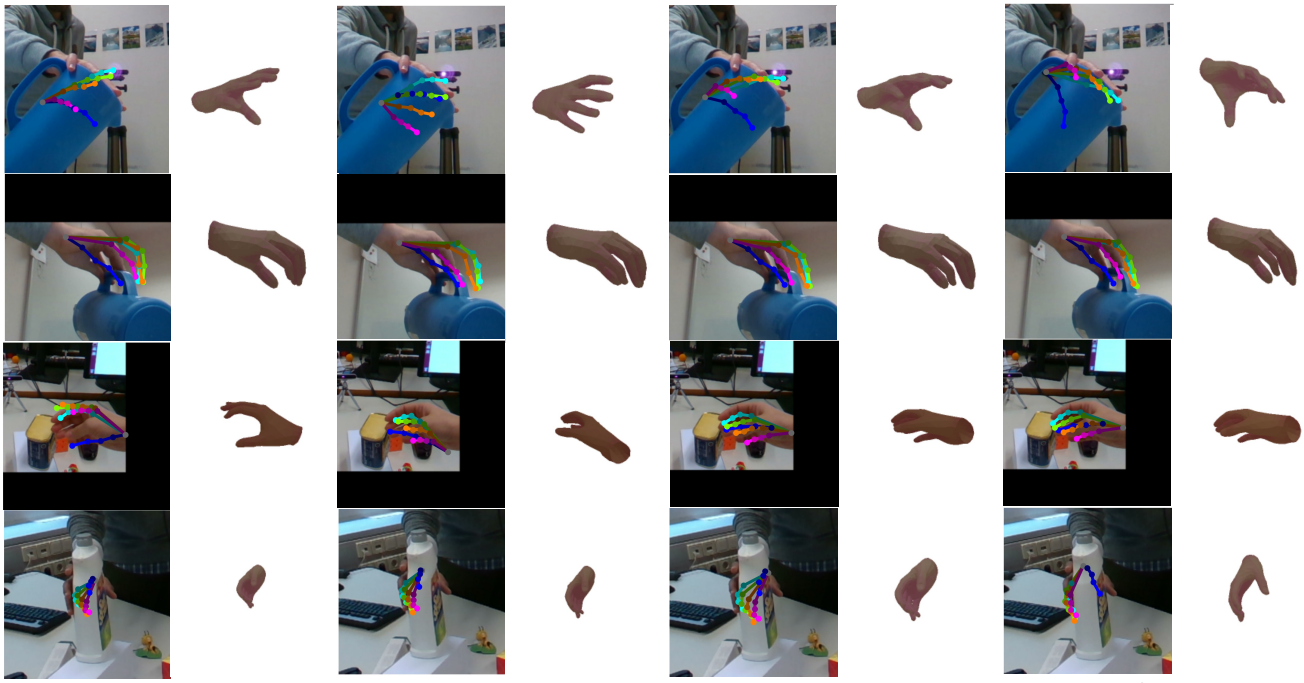


Fig. 8: Qualitative comparison of different motion-related constraints on the HO-3D testing set. Our S^2HAND(V) with the quaternion loss achieves the best qualitative results.

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.

4.5.2 Consistent Hand Reconstruction from Video Sequences

For consistent self-supervised hand reconstruction from video sequences with S^2HAND(V), we conduct ablation studies on HO-3D since it is a widely used challenging dataset that presents hand-object interaction images with sequence information.

We first study the accuracy performance of quaternion loss in comparison with other commonly used motion-related constraints modeling smoothness in sequence outputs. The compared constraints include temporal loss from [36] closing neighboring poses as much as possible and smooth loss from [65] limiting neighboring pose variation with a threshold. Quantitative and qualitative results are presented in Table 4 and Fig. 8. Compared to S^2HAND, S^2HAND(V) with quaternion loss improves single frame prediction by reducing 3.5% in MPVPE, while S^2HAND(V) with smooth loss from [36] only gets a reduction of less than 1% in MPVPE and S^2HAND(V) with temporal loss from [65] even degenerates the accuracy. We think this is because weak supervision is prone to optimize models in the wrong direction and suffers from being too sketchy and limited under self-supervised settings. The temporal loss from [65] is beneficial when the 3D annotation is available but may collapse the model by making the network insensitive to the high-frequency details in absence of strong supervision signals. The smooth loss from [36] introduces threshold, but in exchange enlarges the solution space and vanishes when the threshold is exceeded. In contrast, quaternion loss narrows the solution space based on hand structures and hand motion dynamics and provides more significant supervision signals, which proves its effectiveness over other similar constraints in accuracy.

TABLE 7: Comparison of different configurations of the quaternion loss on the HO-3D testing set.

Config	$AUC_J\uparrow$	MPJPE↓	$AUC_V\uparrow$	MPVPE↓
interv=1, n=3	0.778	1.11	0.780	1.10
interv=3, n=3	**0.780**	**1.10**	**0.782**	**1.09**
interv=6, n=3	0.775	1.13	0.776	1.12
interv=3, n=6	0.774	1.13	0.777	1.12

We next explore our quaternion loss with different configurations in terms of the actual frame interval (interv) of sampled frames and the number of frames (See n in Eq. 17). The results are shown in Table 7. A medium interval value ($\mathrm{interv} = 3$ in Table 7) achieves the best performance. We attribute this to two reasons. On one hand, large interval may witness changes of the joint rotation axis in sampled frames, which contradicts the fixed axis prior in quaternion interpolation (See Section 3.3.1). On the other hand, small interval only witnesses small rotation angles, which limits the effect of quaternion loss. Also, interval is related to the motion speed. In fact, a slow hand motion with larger interval would be equivalent to a fast hand motion with smaller interval. Though the quaternion loss is designed to handle hand motion speed variation, an optimum interval still exists based on the overall motion speed. For number of frames n, increasing it causes model accuracy to drop. We believe this is because optimizing multiple unconfident

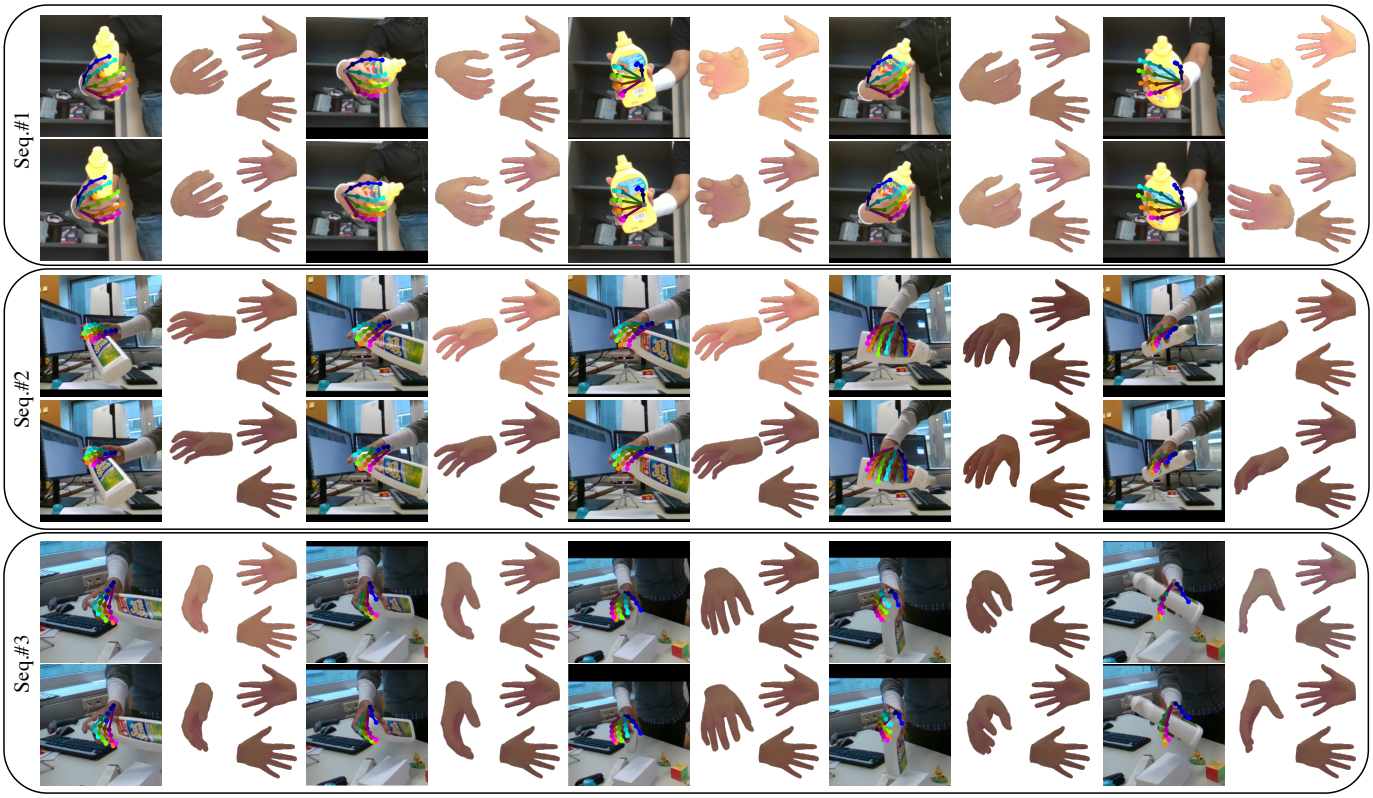


Fig. 9: Qualitative demonstration of the effectiveness of the T&S consistency loss. For each frame, we show output keypoints (left), output 3D reconstruction (middle), and both sides of the output textures (with lighting) in flat hands (top-/bottom-right). For each sequence, we show results without T&S loss on the top row and with T&S loss on the bottom row. T&S loss significantly improves the output appearance consistency in sequence predictions.

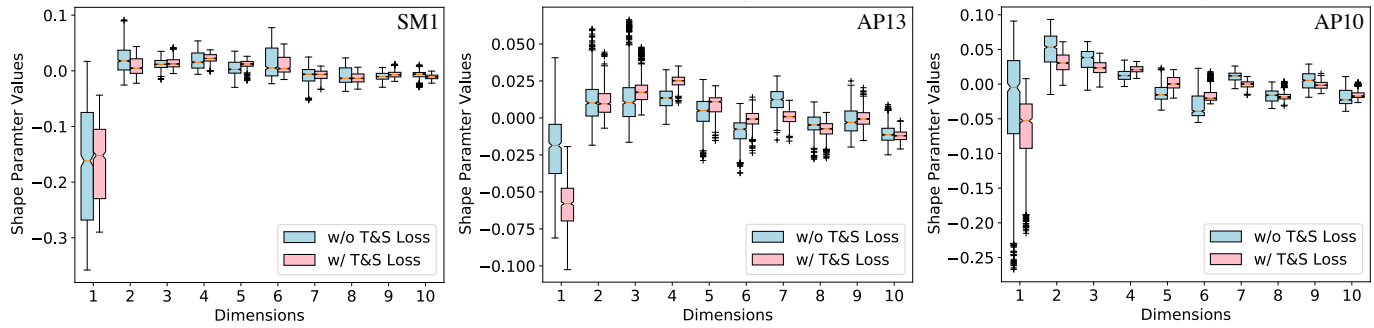


Fig. 10: Boxplots of shape parameters in sequence predictions on the HO-3D testing set. SM1, AP13, and AP10 are sequences from HO-3D testing set. T&S loss reduces S.D. across all 10 dimensions of the shape parameters.

predictions at the same time puts an extra burden on the gradient-descent-based optimizer and destabilizes the learning procedure. From above, we conclude that the best configuration of quaternion loss depends on the frame rates of input video sequences and the confidence of the output. Adjusting the configuration of quaternion loss dynamically according to the input and the output can be a promising direction for future work.

We then compare smoothness performance of quaternion loss with others by concatenating their single frame predictions to corresponding sequences predictions. The results are reported on both train set and test set in Table 5. Note that we only report ACC-ERR on train set since 3D ground truth is required to calculate ACC-ERR. As shown in Table 5, all smoothness related prior improve the smoothness of the sequence outputs, and our proposed quaternion loss achieves the best performance in all evaluation matrices. [65] comes second in terms of smoothness, but shows the worst performance on accuracy in Table 4. We think better smoothness does not imply higher accuracy, where a trade-off between smoothness and accuracy is shown in some cases. Our quaternion loss differently does best on both accuracy and smoothness, which again proves its superiority over similar methods. In addition, we report the average quaternion loss of the concatenated sequences predictions. We find that our quaternion loss is highly in line with ACC and ACC-ERR, which is encouraging since they are calculated from completely different angles. ACC-ERR and ACC regard no hand structure and hand motion characteristic, rely solely on mechanics in terms of acceleration, while quaternion loss does the opposite. Thus, we think our proposed quaternion loss not only proves its advantages over other loss functions in smoothness, but is capable of being the metric measuring smoothness of sequence predictions as well.

Finally, we inspect the effect of regularizing outputs of

Fig. 11: Qualitative results of $S^2HAND(V)$ with extra in-the-wild data. The first three rows show results from HO-3D and the last three rows show results from YT 3D.

TABLE 8: Results of absorbing extra in-the-wild data from YT 3D on the HO-3D testing set. * represents changing camera model from perspective to the orthogonal model, which enables to learn with in-the-wild data without camera information.

Method	$AUC_J\uparrow$	MPJPE↓	$AUC_V\uparrow$	MPVPE↓	$F_5\uparrow$	$F_{15}\uparrow$
S^2HAND	0.773	1.14	0.777	1.12	0.45	0.93
S^2HAND*	0.770	1.15	0.774	1.13	0.45	0.93
$S^2HAND(V)$* (w/ Quaternion Loss)	0.778	1.11	0.780	1.10	0.45	**0.94**
$S^2HAND(V)$* (w/ Quaternion Loss) + YT 3D data	**0.782**	**1.09**	**0.783**	**1.09**	**0.46**	**0.94**

TABLE 9: Results of hand appearance consistency of our methods on the HO-3D testing set. Texture S.D. is computed using lighted textures defined in Eq. 22.

Method	Texture S.D.	Shape S.D.
S^2HAND	0.033	0.013
$S^2HAND(V)$ (w/ Quaternion Loss)	0.042	0.016
$S^2HAND(V)$ (w/ Quaternion Loss, T&S Loss)	**0.016**	**0.012**

hand shape and texture in sequence predictions with the T&S consistency loss. The results are presented in Table 4, Table 9, Fig. 9 and Fig. 10. Though the proposed T&S consistency loss does not further improve pose accuracy as shown in Table 4, it significantly improves the hand appearance consistency in sequence predictions. Quantitatively, with the T&S loss, the shape S.D. and texture S.D. drops 62% and 25% respectively (see Table 9). Qualitatively, $S^2HAND(V)$ with the T&S loss gives more consistent reconstructions of different frames from the same sequence than without T&S as shown in Fig. 9. We also provide boxplots of shape parameters to show dimensional S.D. reduction in Fig. 10. Additionally, a visualization of per-face texture S.D. is provided in the supplementary material.

4.5.3 Hand Reconstruction with Extra In-the-wild Data

For hand reconstruction with extra in-the-wild data to fully exploit the advantage of our self-supervised method $S^2HAND(V)$, we use 34 train videos from YT 3D [40] and switch to orthogonal camera model with corresponding camera projection to enable learning with in-the-wild data without camera information.

The results are presented in Table 8 and Fig. 11. Changing the camera model makes the model performance drop a little, which may be caused by the ambiguity of camera focal length. Then, imposing proposed quaternion loss boosts the performance by 3.4%, which conforms to experiments results in Section 4.5.2. Finally, adding extra in-the-wild data further improves our model by 1.8%, resulting in 1.09cm in MPVPE, which is the best result we can get on the HO-3D test set. From above, we see that our proposed method is able to utilize extra in-the-wild data without any camera information and benefits from these training data. It is worth noticing that there is a certain domain gap between HO-3D and YT 3D since HO-3D regards hand-object interaction scenario while YT 3D is mostly comprised of sign language videos. However, the improvement on HO-3D has still been witnessed, which we think proves the significance of utilizing in the data and the advantage of the proposed method.

5 Conclusion

In this work, we have proposed self-supervised 3D hand reconstruction models S^2HAND and $S^2HAND(V)$ which can be trained from a collection of hand images and video sequences without manual annotations, respectively. The 3D hand reconstruction network in both models encodes the input image into a set of meaningful semantic parameters that represent hand pose, shape, texture, illumination, and the camera viewpoint. These parameters can be decoded into a textured 3D hand mesh as well as a set of 3D joints, and in turn, the 3D mesh and joints can be projected into the 2D image space, which enables our network to be end-to-end learned. We further exploit the self-supervision signals

embedded in hand motion videos by developing a novel quaternion loss and a texture and shape consistency loss to obtain more accurate and consistent hand reconstruction. Experimental results show that our models perform well under noisy supervision sources captured from 2D hand keypoint detection, and achieve comparable performance to the state-of-the-art fully-supervised method. Moreover, the experiments on in-the-wild video data show that our self-supervised model is effective to learn useful information from in-the-wild data to further improve its performance.

For the future study, we think the texture and shape representation could be investigated deeply. Notice that we only estimate a per-face color and adopt a low-resolution hand model with 778 vertices. Besides, we rely on the shape and texture regularization terms to learn hand appearance from limited raw data under complex environments and potential hand-object interaction. These defects lead to the reconstruction results having limited details. Recently, Corona et al. [91] utilize implicit representation to learn hand shape and texture of high quality. This points in a good direction for further exploration.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 62106177 and the National Science Fund for Distinguished Young Scholars of China under Grant 41725005. It was also supported by the Joint Fund of the Ministry of Education of China under Grant 8091B032156. The numerical calculation was supported by the supercomputing system in the Supercomputing Center of Wuhan University.

REFERENCES

[1] H. Lee, M. Billinghurst, and W. Woo, "Two-handed tangible interaction techniques for composing augmented blocks," Virtual Reality, vol. 15, no. 2, pp. 133–146, 2011.
[2] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in Conference on Computer Vision and Pattern Recognition, 2018.
[3] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in Conference on Computer Vision and Pattern Recognition, 2020.
[4] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in Conference on Computer Vision and Pattern Recognition, 2018.
[5] Z. Tu, J. Zhang, H. Li, Y. Chen, and J. Yuan, "Joint-bone fusion graph convolutional network for semi-supervised skeleton action recognition," IEEE Transactions on Multimedia, 2022.
[6] M. Höll, M. Oberweger, C. Arth, and V. Lepetit, "Efficient physics-based implementation for realistic hand-object interaction in virtual reality," in Conference on Virtual Reality and 3D User Interfaces, 2018.
[7] M. Parelli, K. Papadimitriou, G. Potamianos, G. Pavlakos, and P. Maragos, "Exploiting 3d hand pose estimation in deep learning-based sign language recognition from rgb videos," in European Conference on Computer Vision, 2020.
[8] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, and J. Yuan, "Action-stage emphasized spatiotemporal vlad for video action recognition," IEEE Transactions on Image Processing, vol. 28, no. 6, pp. 2799–2812, 2019.
[9] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3d hand pose estimation from single depth images using multi-view cnns," IEEE Transactions on Image Processing, vol. 27, no. 9, pp. 4422–4436, 2018.
[10] Z. Yu, J. S. Yoon, I. K. Lee, P. Venkatesh, J. Park, J. Yu, and H. S. Park, "Humbi: A large multiview dataset of human body expressions," in Conference on Computer Vision and Pattern Recognition, 2020.
[11] Z. Zhao, T. Wang, S. Xia, and Y. Wang, "Hand-3d-studio: A new multi-view system for 3d hand reconstruction," in IEEE International Conference on Acoustics, Speech, and Signal Processing, 2020.
[12] G. Poier, D. Schinagl, and H. Bischof, "Learning pose specific representations by predicting different views," in Conference on Computer Vision and Pattern Recognition, 2018.
[13] A. Armagan, G. Garcia-Hernando, S. Baek, S. Hampali, M. Rad, Z. Zhang, S. Xie, M. Chen, B. Zhang, F. 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.
[14] Y. Chen, Z. Tu, L. Ge, D. Zhang, R. Chen, and J. Yuan, "So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning," in International Conference on Computer Vision, 2019.
[15] L. Ge, H. Liang, J. Yuan, and D. 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] L. Huang, J. Tan, J. Liu, and J. Yuan, "Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation," in European Conference on Computer Vision, 2020.
[17] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Yong Chang, K. Mu Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge et al., "Depth-based 3d hand pose estimation: From current achievements to future goals," in Conference on Computer Vision and Pattern Recognition, 2018.
[18] V. Athitsos and S. Sclaroff, "Estimating 3d hand pose from a cluttered image," in Conference on Computer Vision and Pattern Recognition, 2003.
[19] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in European Conference on Computer Vision, 2018.
[20] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz, "Hand pose estimation via latent 2.5 d heatmap regression," in European Conference on Computer Vision, 2018.
[21] A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges, and J. Kautz, "Weakly supervised 3d hand pose estimation via biomechanical constraints," in European Conference on Computer Vision, 2020.
[22] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in International Conference on Computer Vision, 2017.
[23] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3d hand shape and pose estimation from a single rgb image," in Conference on Computer Vision and Pattern Recognition, 2019.
[24] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in Conference on Computer Vision and Pattern Recognition, 2020.
[25] Y. Hasson, B. Tekin, F. Bogo, I. Laptev, M. Pollefeys, and C. Schmid, "Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction," in Conference on Computer Vision and Pattern Recognition, 2020.
[26] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in Conference on Computer Vision and Pattern Recognition, 2019.
[27] N. Qian, J. Wang, F. Mueller, F. Bernard, V. Golyanik, and C. Theobalt, "Parametric hand texture model for 3d hand reconstruction and personalization," in European Conference on Computer Vision, 2020.
[28] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 01, pp. 172–186, 2019.
[29] Y. Chen, F. Wu, Z. Wang, Y. Song, Y. Ling, and L. Bao, "Self-supervised learning of detailed 3d face reconstruction," IEEE Transactions on Image Processing, vol. 29, pp. 8696–8705, 2020.
[30] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt, "Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," in International Conference on Computer Vision Workshops, 2017.

[31] Y. Chen, Z. Tu, D. Kang, L. Bao, Y. Zhang, X. Zhe, R. Chen, and J. Yuan, "Model-based 3d hand reconstruction via self-supervised learning," in Conference on Computer Vision and Pattern Recognition, 2021.

[32] Y. Cai, L. Ge, J. Cai, N. M. Thalmann, and J. Yuan, "3d hand pose estimation using synthetic data and weakly labeled rgb images," IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 11, pp. 3739–3753, 2020.

[33] L. Yang and A. Yao, "Disentangling latent hands for image synthesis and pose estimation," in Conference on Computer Vision and Pattern Recognition, 2019.

[34] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in Conference on Computer Vision and Pattern Recognition, 2021.

[35] P. Chen, Y. Chen, D. Yang, F. Wu, Q. Li, Q. Xia, and Y. Tan, "I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling," in International Conference on Computer Vision, 2021.

[36] S. Liu, H. Jiang, J. Xu, S. Liu, and X. Wang, "Semi-supervised 3d hand-object poses estimation with interactions in time," in Conference on Computer Vision and Pattern Recognition, 2021.

[37] Y. Zhou, M. Habermann, I. Habibie, A. Tewari, C. Theobalt, and F. Xu, "Monocular real-time full body capture with inter-part correlations," in Conference on Computer Vision and Pattern Recognition, 2021.

[38] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng, "Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration," in Conference on Computer Vision and Pattern Recognition, 2021.

[39] Z. Cao, I. Radosavovic, A. Kanazawa, and J. Malik, "Reconstructing hand-object interactions in the wild," in International Conference on Computer Vision, 2021.

[40] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in Conference on Computer Vision and Pattern Recognition, 2020.

[41] I. Lim, A. Dielen, M. Campen, and L. Kobbelt, "A simple approach to intrinsic correspondence learning on unstructured 3d meshes," in European Conference on Computer Vision, 2018.

[42] D. Kulon, H. Wang, R. A. Güler, M. M. Bronstein, and S. Zafeiriou, "Single image 3d hand reconstruction with mesh convolutions," in British Machine Vision Conference, 2019.

[43] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," Advances in neural information processing systems, vol. 29, p. 3844–3852, 2016.

[44] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3d hand shape and pose estimation from a single rgb image," in Conference on Computer Vision and Pattern Recognition, 2019.

[45] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia), vol. 36, no. 6, 2017.

[46] A. Boukhayma, R. d. Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," in Conference on Computer Vision and Pattern Recognition, 2019.

[47] Y. Chen, Z. Tu, D. Kang, R. Chen, L. Bao, Z. Zhang, and J. Yuan, "Joint hand-object 3d reconstruction from a single image with cross-branch feature fusion," IEEE Transactions on Image Processing, vol. 30, pp. 4008–4021, 2021.

[48] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in International Conference on Computer Vision, 2019.

[49] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in Conference on Computer Vision and Pattern Recognition, 2019.

[50] ——, "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.

[51] J. Wang, F. Mueller, F. Bernard, S. Sorli, O. Sotnychenko, N. Qian, M. A. Otaduy, D. Casas, and C. Theobalt, "Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video," ACM Transactions on Graphics, vol. 39, no. 6, pp. 1–16, 2020.

[52] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, and F. Xu, "Monocular real-time hand shape and motion capture using multi-modal data," in Conference on Computer Vision and Pattern Recognition, 2020.

[53] P. Panteleris, I. Oikonomidis, and A. 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.

[54] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3d objects from images in the wild," in Conference on Computer Vision and Pattern Recognition, 2020.

[55] S. Goel, A. Kanazawa, and J. Malik, "Shape and viewpoint without keypoints," in European Conference on Computer Vision, 2020.

[56] A. Spurr, A. Dahiya, X. Wang, X. Zhang, and O. Hilliges, "Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning," in International Conference on Computer Vision, 2021.

[57] S. Guo, E. Rigall, L. Qi, X. Dong, H. Li, and J. Dong, "Graph-based cnns with self-supervised module for 3d hand pose estimation from monocular rgb," IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 4, pp. 1514–1525, 2020.

[58] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Self-supervised 3d hand pose estimation through training by fitting," in Conference on Computer Vision and Pattern Recognition, 2019.

[59] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz," in Conference on Computer Vision and Pattern Recognition, 2018.

[60] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in ACM Transactions on Graphics (Proceedings of SIGGRAPH), 1999, p. 187–194.

[61] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3d hand pose estimation from monocular video," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 9, pp. 1793–1805, 2011.

[62] M. de La Gorce, N. Paragios, and D. J. Fleet, "Model-based hand tracking with texture, shading and self-occlusions," in Conference on Computer Vision and Pattern Recognition, 2008.

[63] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in International Conference on Computer Vision, 2019.

[64] L. Chen, S.-Y. Lin, Y. Xie, Y.-Y. Lin, and X. Xie, "Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos," in Winter Conference on Applications of Computer Vision, 2021.

[65] J. Yang, H. J. Chang, S. Lee, and N. Kwak, "Seqhand: Rgb-sequence-based 3d hand pose and shape estimation," in European Conference on Computer Vision, 2020.

[66] J. Zhang, Z. Tu, J. Yang, Y. Chen, and J. Yuan, "Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13 232–13 242.

[67] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in International Conference on Computer Vision, 2015.

[68] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in Conference on Computer Vision and Pattern Recognition, 2017.

[69] E. Aksan, M. Kaufmann, and O. Hilliges, "Structured prediction helps 3d human motion modelling," in International Conference on Computer Vision, 2019.

[70] A. Hernandez, J. Gall, and F. Moreno-Noguer, "Human motion prediction via spatio-temporal inpainting," in International Conference on Computer Vision, 2019.

[71] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in International Conference on Machine Learning, 2019.

[72] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in Conference on Computer Vision and Pattern Recognition, 2018.

[73] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in European Conference on Computer Vision, 2016.

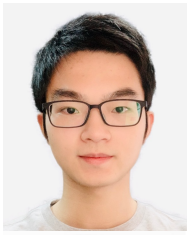
[74] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in European Conference on Computer Vision, 2018.

[75] P. J. Huber, "Robust estimation of a location parameter," in Breakthroughs in statistics, 1992.

[76] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.

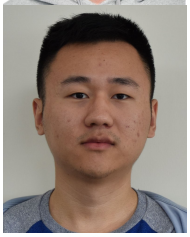
[77] D. Pavllo, C. Feichtenhofer, M. Auli, and D. Grangier, "Modeling human motion with quaternion-based neural networks," International Journal of Computer Vision, vol. 128, no. 4, pp. 855–872, 2020.
[78] X. Zhang, S. Qin, Y. Xu, and H. Xu, "Quaternion product units for deep learning on 3d rotation groups," in Conference on Computer Vision and Pattern Recognition, 2020.
[79] K. Shoemake, "Animating rotation with quaternion curves," in ACM Transactions on Graphics (Proceedings of SIGGRAPH), vol. 19, no. 3, 1985, p. 245–254.
[80] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in Conference on Computer Vision and Pattern Recognition, 2019.
[81] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3d annotation of hand and object poses," in Conference on Computer Vision and Pattern Recognition, 2020.
[82] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang, "A hand pose tracking benchmark from stereo matching," in International Conference on Image Processing, 2017.
[83] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," ACM Transactions on Graphics, vol. 36, no. 4, pp. 1–13, 2017.
[84] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3d human dynamics from video," in Conference on Computer Vision and Pattern Recognition, 2019.
[85] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in Neural Information Processing Systems Workshops, 2017.
[86] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in International Conference for Learning Representations, 2014.
[87] G. Moon and K. M. 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.
[88] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in Conference on Computer Vision and Pattern Recognition, 2018.
[89] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in Conference on Computer Vision and Pattern Recognition, 2018.
[90] T. Theodoridis, T. Chatzis, V. Solachidis, K. Dimitropoulos, and P. Daras, "Cross-modal variational alignment of latent spaces," in Conference on Computer Vision and Pattern Recognition Workshops, 2020.
[91] E. Corona, T. Hodan, M. Vo, F. Moreno-Noguer, C. Sweeney, R. Newcombe, and L. Ma, "Lisa: Learning implicit shape and appearance of hands," in Conference on Computer Vision and Pattern Recognition, 2022.

Zhigang Tu started his Master Degree in image processing at the School of Electronic Information, Wuhan University, China, 2008. In 2015, he received the Ph.D. degree in Computer Science from Utrecht University, Netherlands. From 2015 to 2016, he was a postdoctoral researcher at Arizona State University, US. Then from 2016 to 2018, he was a research fellow at the School of EEE, Nanyang Technological University, Singapore. He is currently a professor at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University. His research interests include computer vision, image processing, video analytics, and machine learning. Special for motion estimation, video super-resolution, object segmentation, action recognition and localization, and anomaly detection. He has co-/authored more than 50 articles on international SCI-indexed journals and conferences. He is an Associate Editor of the SCI-indexed journal TVC (IF=1.456), a Guest Editor of JVCIR (IF=2.479) and Combinatorial Chemistry and High Throughput Screening (IF=1.195). He is the first organizer of the ACCV2020 Workshop on MMHAU (Japan). He received the "Best Student Paper" Award in the 4^{th} Asian Conference on Artificial Intelligence Technology.



Zhisheng Huang received the B.Eng. degree in Surveying and Mapping Engineering Technology from Wuhan University in 2021. He is now a postgraduate student with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. His research interests mainly include computer vision, machine learning and human-computer interaction.

Yujin Chen received the B.Eng. and M.Sc. in Geo-Information from Wuhan University. He is currently a Ph.D. student with the Visual Computing Lab, Technical University of Munich, Germany. His research interests include computer vision and machine learning. His current focus is on 3D scene understanding, pose estimation, motion analysis, and representation learning.



Di Kang received the B.Eng. degree in Electronic Engineering and Information Science (EEIS) from the University of Science and Technology of China (USTC) in 2014, and the Ph.D. degree in computer science from City University of Hong Kong in 2019. In 2019, he jointed Tencent AI Lab, and is now serving as a senior researcher. His research interests include computer vision, and deep learning.

Linchao Bao received Ph.D. degree in Computer Science from City University of Hong Kong in 2015. Prior to that, he received M.S. degree in Pattern Recognition and Intelligent Systems from Huazhong University of Science and Technology in Wuhan, China. He was a research intern at Adobe Research from November 2013 to August 2014 and worked for DJI as an algorithm engineer from January 2015 to June 2016. His research interests include computer vision and graphics.

Bisheng Yang received the B.S. degree in engineering survey, the M.S. and Ph.D. degrees in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1996, 1999, and 2002, respectively. From 2002 to 2006, he was a Post-doctoral Research Fellow with the University of Zurich, Zurich, Switzerland. Since 2007, he has been a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, where he is currently the Director of the 3S and Network Communication Laboratory. He has hosted a project of the National High Technology Research and Development Program, a key project of the Ministry of Education, and four National Scientific Research Foundation Projects of China. His current research interests include 3-D geographic information systems, urban modeling, and digital city. He was a Guest Editor of the ISPRS Journal of Photogrammetry and Remote Sensing and Computers & Geosciences.

Junsong Yuan is Professor and Director of Visual Computing Lab at Department of Computer Science and Engineering (CSE), State University of New York at Buffalo, USA. Before joining SUNY Buffalo, he was Associate Professor (2015-2018) and Nanyang Assistant Professor (2009-2015) at Nanyang Technological University (NTU), Singapore. He obtained his Ph.D. from Northwestern University in 2009, M.Eng. from National University of Singapore in 2005, and B.Eng. from Huazhong University of Science Technology (HUST) in 2002. His research interests include computer vision, pattern recognition, video analytics, human action and gesture analysis, large-scale visual search and mining. He received Best Paper Award from IEEE Trans. on Multimedia, Nanyang Assistant Professorship from NTU, and Outstanding EECS Ph.D. Thesis award from Northwestern University. He served as Associate Editor of IEEE Trans. on Image Processing (T-IP), IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT), Machine Vision and Applications (MVA), and Senior Area Editor of Journal of Visual Communications and Image Representation (JVCI). He was Program Co-Chair of IEEE Conf. on Multimedia Expo (ICME'18), and served as Area Chair for CVPR, ICCV, and ACM MM. He is a Fellow of IEEE and IAPR.