Monocular 3D Hand Mesh Recovery via Dual Noise Estimation

Hanhui Li[1], Xiaojian Lin[1], Xuan Huang[1], Zejun Yang[2], Zhisheng Wang[2], Xiaodan Liang[1, 3*]

[1]Shenzhen Campus of Sun Yat-sen University, Shenzhen, China
[2]Tencent, Shenzhen, China
[3]DarkMatter AI Research, Guangzhou, China
lihh77@sysu.edu.cn, xdliang328@gmail.com

Abstract

Current parametric models have made notable progress in 3D hand pose and shape estimation. However, due to the fixed hand topology and complex hand poses, current models are hard to generate meshes that are aligned with the image well. To tackle this issue, we introduce a dual noise estimation method in this paper. Given a single-view image as input, we first adopt a baseline parametric regressor to obtain the coarse hand meshes. We assume the mesh vertices and their image-plane projections are noisy, and can be associated in a unified probabilistic model. We then learn the distributions of noise to refine mesh vertices and their projections. The refined vertices are further utilized to refine camera parameters in a closed-form manner. Consequently, our method obtains well-aligned and high-quality 3D hand meshes. Extensive experiments on the large-scale Interhand2.6M dataset demonstrate that the proposed method not only improves the performance of its baseline by more than 10% but also achieves state-of-the-art performance. Project page: https://github.com/hanhuili/DNE4Hand.

Introduction

Recent advances in parametric human models (Pavlakos et al. 2019) have facilitated human-centric applications, such as artificial intelligence generated content, human avatars, and virtual talking heads. With parametric models like (Romero, Tzionas, and Black 2017), reconstructing 3D hand meshes from images becomes plausible and convenient. This attracts considerable attention and extensive research has been conducted to improve the accuracy and speed of the parametric model fitting process (Zhang et al. 2021; Yu et al. 2023a; Meng et al. 2022; Moon 2023; Li et al. 2022b; Chen et al. 2022a).

Nevertheless, reconstructing well-aligned hand meshes from single-view images is still challenging because of the following two reasons: (i) Challenging factors like depth ambiguity, self/inter-hand occlusions, and complicated hand motions hinder estimation accuracy. (ii) Even worse, the predefined hand topology in parametric models further restricts hand mesh deformations, and consequently the parametric models are not flexible enough to represent various hands.

*Xiaodan Liang is the corresponding author.

Non-parametric hand models (Lin, Wang, and Liu 2021a,b; Lin et al. 2022; Jiang et al. 2023) seem to be a possible solution for the above issues. With the powerful representation learning ability (Tian et al. 2023), current non-parametric methods can predict mesh vertices directly. This provides great flexibility and in practice methods of this type usually yield better performance, compared with parametric models. However, without leveraging the hand structural prior, these methods are prone to producing severe artifacts and broken meshes.

It is natural to consider combining parametric models and non-parametric models to leverage the structural advantages of the former and the flexibility of the latter. Methods belonging to this paradigm (Tang, Wang, and Fu 2021; Li et al. 2022a; Ren et al. 2023; Yu et al. 2023b; Moon 2023) have been proposed recently. However, as far as we are concerned, most current methods seek a deterministic manner, such as predicting the deviations of vertices and parameters (Tang, Wang, and Fu 2021). This makes them hard to explore the solution space thoroughly. Note that recovering 3D hand meshes from monocular images is an ill-posed problem, which means multiple meshes can be associated with the same 2D observation. Therefore, a deterministic model may be ineffective to tackle this task.

To tackle the above issues, we propose to tackle monocular hand mesh recovery in a probabilistic framework. Particularly, given a monocular input image, we adopt an off-the-shelf parametric model as the baseline to obtain the coarse hand meshes. We then refine the coarse hand meshes by jointly estimating the noise of vertices and their image-plane projections, since they are highly related. We design a progressive framework to do so, in which image-aligned features are leveraged to estimate the parameters governing the noise distributions. With the estimated distributions, we adopt the reparameterization trick to generate multiple samples and estimate their confidence. In this way, we can leverage the sample with the highest confidence to optimize the hand vertices and their projections. Moreover, given the refined vertices and their 2D coordinates, we also propose a closed-form solution to refine camera parameters. Consequently, our method can generate hand meshes that are aligned with images well. Our experiments on the Interhand2.6M dataset show that the proposed method can boost the quality of the coarse meshes significantly, and achieve

the state-of-the-art performance.

Our contributions can be summarized as follows:

• To the best of our knowledge, this paper proposes the first probabilistic 2D and 3D noise estimation framework for the monocular hand mesh recovery task.

• An effective network architecture is introduced to realize the dual noise estimation process. This network leverages image-aligned features and multiple samples to enhance the coarse meshes generated by baseline parametric models.

• The proposed method is validated on the large-scale Interhand2.6M dataset and outperforms conventional methods.

Related Work

Parametric Hand Models

Parametric models for 3D hand meshes have gained considerable attention because it provides the convenient structural/geometric prior of hands. Extensive methods have been proposed for fitting parametric models, such as attention modules (Zhang et al. 2021; Yu et al. 2023a), inverse kinematic solvers (Shetty et al. 2023; Li et al. 2023), hand disentanglement (Meng et al. 2022; Moon 2023), and 2D-3D projection (Li et al. 2022b). The comprehensive review of parametric models can refer to (Tian et al. 2023). Parametric models can be roughly divided into regression based methods and optimization based methods. Regression based methods estimate the parameters directly while optimization based methods usually involve an online optimization process. Parametric models are restricted by their pre-defined hand templates and are inflexible to model hands of various poses and geometry.

Non-parametric Hand Models

Early non-parametric models aim at predicting hand joints from depth maps and point clouds (Cheng et al. 2022; Deng et al. 2022). With the recent advances in network architectures, non-parametric models that predict 3D hand vertices become popular. For instance, transformers and graph neural networks (Lin, Wang, and Liu 2021a,b; Lin et al. 2022; Jiang et al. 2023) have been proposed for mesh reconstruction. Since non-parametric models do not rely on the fixed hand topology, they are more flexible and easier to be aligned with images. However, without the structural prior of hands, non-parametric models also suffer from distorted and spiky reconstruction results.

Hybrid Hand Models

It is reasonable to construct hybrid models and leverage the advantages of both parametric and non-parametric models. Several pioneering studies have been conducted to achieve this goal. For example, IntagHand (Li et al. 2022a) utilizes the topology of MANO to construct the graph representation of vertices. It also defines graph attention modules to model vertex dependencies. Ren et al. 2023 incorporate the MANO model into a point cloud network for pose estimation. Yu et al. 2023b propose to estimate joints first via non-parametric models and then infer MANO parameters based on joints. Moon 2023 introduce a network to predict relative translation between two MANO hands.

The proposed method differs from traditional methods because of its probabilistic unified modeling of vertices and their 2D coordinates. With the proposed method, we can achieve the mutual and progressive refinement between vertices and 2D coordinates.

Implicit Hand Models

Except for explicit representations, recent studies also try to explore implicit functions (e.g., signed distance function, Park et al. 2019) to represent 3D hands. A notable advantage of implicit functions is that they are continuous and disentangled from spatial resolutions. This advantage indicates that implicit functions can generalize to arbitrary hands. Several implicit hand models have been proposed, such as LISA (Corona et al. 2022), AlignSDF (Chen et al. 2022b), Im2Hands (Lee et al. 2023), HandNeRF (Guo et al. 2023), and Hand Avatar (Chen, Wang, and Shum 2023). However, compared with explicit models, the computational cost of implicit models is more expensive.

Methodology

Overall Architecture

The architecture of our method is shown in Figure 1. It consists of a coarse mesh fitting stage and a refinement stage. Particularly, given a single-view image as input, we adopt ResNet-50 (He et al. 2016) as the image encoder to extract the feature maps $\mathbf{F}$ of size $H \times W \times C$. To better leverage the geometric and semantic information in the image, we adopt a 2D convolutional block to predict five auxiliary maps, including the depth map $\mathbf{a}_1 \in \mathbb{R}^{H\times W}$, the normal map $\mathbf{a}_2 \in \mathbb{R}^{H\times W\times 3}$, the joint heat map $\mathbf{a}_3 \in \mathbb{R}^{H\times W\times 42}$ (each hand has 21 joints), the DensePose map (Güler, Neverova, and Kokkinos 2018) $\mathbf{a}_4 \in \mathbb{R}^{H\times W\times 3}$, and the part semantic map $\mathbf{a}_5 \in \mathbb{R}^{H\times W\times 34}$ (16 parts for each hand, plus one background class). We merge $\mathbf{F}$ and these five auxiliary maps via channel-wise concatenation followed by another 2D convolutional block. For conciseness, we still denote the merged feature maps as $\mathbf{F}$.

$\mathbf{F}$ are then fed into a baseline fitting model to predict the parameters of MANO, including the pose coefficient $\boldsymbol{\theta} \in \mathbb{R}^{16\times 6}$ (Zhou et al. 2019), the shape coefficient $\boldsymbol{\beta} \in \mathbb{R}^{10}$, and the intrinsic camera parameters $\mathbf{c} \in \mathbb{R}^{2\times 2}$ for each hand. Inspired by Yu et al. 2023a, we utilize 2D convolutions to predict parameter maps and accumulate the parameters via spatial softmax. With the fitted parameters, we generate the initial coarse hand meshes.

To refine the coarse meshes and better align them with images, we introduce the dual noise estimation (DNE) module. The core of DNE is to conduct mesh refinement and alignment jointly in a denoising process. To this end, the DNE module first refines the image-plane projections of coarse vertices. Then the DNE module obtains the image-aligned features via interpolation and uses them to estimate 3D vertex deviations. Furthermore, based on the correspondences between 3D vertices and their 2D projections, the DNE module also adopts a closed-form solution to refine the camera parameters. The detailed architecture of the DNE

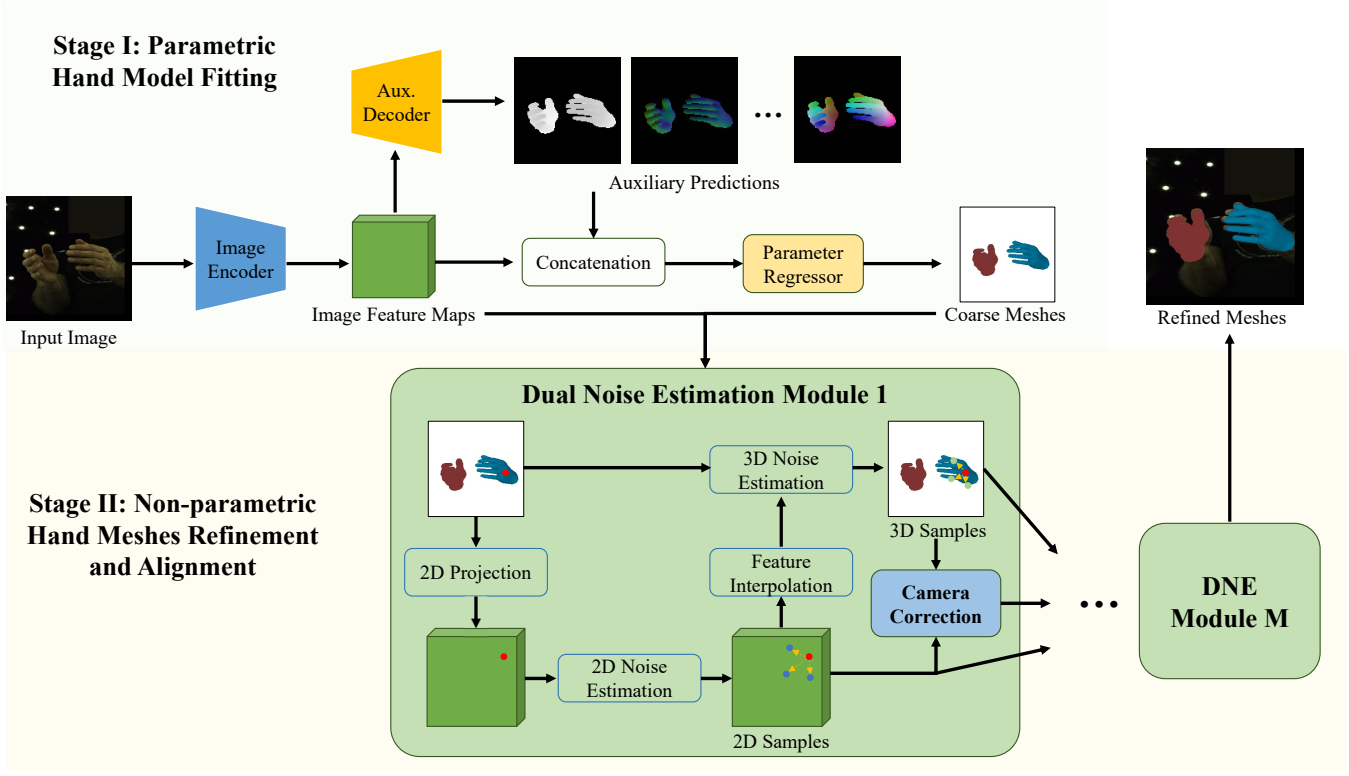

Figure 1: The overall framework of the proposed method. It is a two-stage that first generate coarse meshes and then refine them via multiple dual noise estimation modules.

module is presented in the next section. The above refinement process is conducted progressively via multiple DNE modules and in practice we find that more DNE modules yield more significant performance gains.

Dual Noise Estimation

Formulation. Given an arbitrary vertex $\mathbf{v} \in \mathbb{R}^3$ and its corresponding 2D coordinate $\mathbf{u} \in \mathbb{R}^2$, our proposed DNE module can be formulated as follows:

$$\Pi(\mathbf{v} + \boldsymbol{\varepsilon}_{3d}, \mathbf{c}) = \mathbf{u} + \boldsymbol{\varepsilon}_{2d}, \tag{1}$$

where Π denotes the 2D projection of $\mathbf{v}$ given the intrinsic camera parameters $\mathbf{c}$. Note that $\mathbf{u}$ is not necessarily obtained by Π. As we demonstrate later, $\mathbf{u}$ can be regressed from the image feature maps directly. $\boldsymbol{\varepsilon}_{3d}$ and $\boldsymbol{\varepsilon}_{2d}$ are the 3D and 2D noise terms that need to be estimated. To ensure our network is differentiable, we assume both $\boldsymbol{\varepsilon}_{3d}$ and $\boldsymbol{\varepsilon}_{2d}$ follow a certain distribution, on which we can apply the reparameterization trick (Kingma and Welling 2013). In this paper, we adopt the Gaussian distribution to model $\boldsymbol{\varepsilon}_{3d}$ and $\boldsymbol{\varepsilon}_{2d}$, namely,

$$\begin{aligned} \boldsymbol{\varepsilon}_{3d} &\sim \mathcal{N}(\boldsymbol{\mu}_{3d}, \gamma|\boldsymbol{\mu}_{3d}| + \delta) \\ \boldsymbol{\varepsilon}_{2d} &\sim \mathcal{N}(\boldsymbol{\mu}_{2d}, \gamma|\boldsymbol{\mu}_{2d}| + \delta) \end{aligned} \tag{2}$$

where $\gamma, \delta > 0$ are hyperparameters that control the scale and margin of noise, respectively. Based on Eq. (2), we can sample multiple $\boldsymbol{\varepsilon}_{3d}$ and $\boldsymbol{\varepsilon}_{2d}$ to better explore the solution space during training, and set $\boldsymbol{\varepsilon}_{3d} = \boldsymbol{\mu}_{3d}$ and $\boldsymbol{\varepsilon}_{2d} = \boldsymbol{\mu}_{2d}$ for inference. This makes the proposed method differ from traditional methods that only estimate 2D/3D deviations. Our task now turns to estimating appropriate $\boldsymbol{\mu}_{3d}$ and $\boldsymbol{\mu}_{2d}$.

$\boldsymbol{\mu}_{2d}$ Estimation. Image-aligned features that are obtained via feature interpolation are leveraged to estimate $\boldsymbol{\mu}_{2d}$. Particularly, we consider two types of 2D coordinates in the interpolation process, including (i) the 2D projections of vertices (i.e., $\Pi(\mathbf{v}, \mathbf{c})$) and (ii) those that are regressed directly from the image feature maps. The intuition behind such a combination is that features obtained by the first type can maintain the hand structure and be robust to outliers, while those of the second type are more flexible.

To regress 2D coordinates from the image feature maps $\mathbf{F}$, we reshape $\mathbf{F}$ to $(HW) \times C$ and adopt two consecutive multilayer perceptrons (MLPs) to transform $\mathbf{F}$ to $N \times C$ first and then $N \times 3$, where $N = 778$ is the number of vertices of one MANO hand. Let $\mathbf{f}_p$ and $\mathbf{f}_r$ denote the C-dimensional interpolated feature vector with the projected and regressed coordinates, respectively. We consider the following transformation $\phi : \mathbb{R}^{2C} \rightarrow \mathbb{R}^2$ to obtain the per-vertex mean of 2D noise:

$$\boldsymbol{\mu}_{2d} = \phi(\mathbf{f}_p, \mathbf{f}_r). \tag{3}$$

In our network, ϕ is implemented efficiently via feature concatenation followed by an MLP.

$\boldsymbol{\mu}_{3d}$ Estimation. We also extract image-aligned features

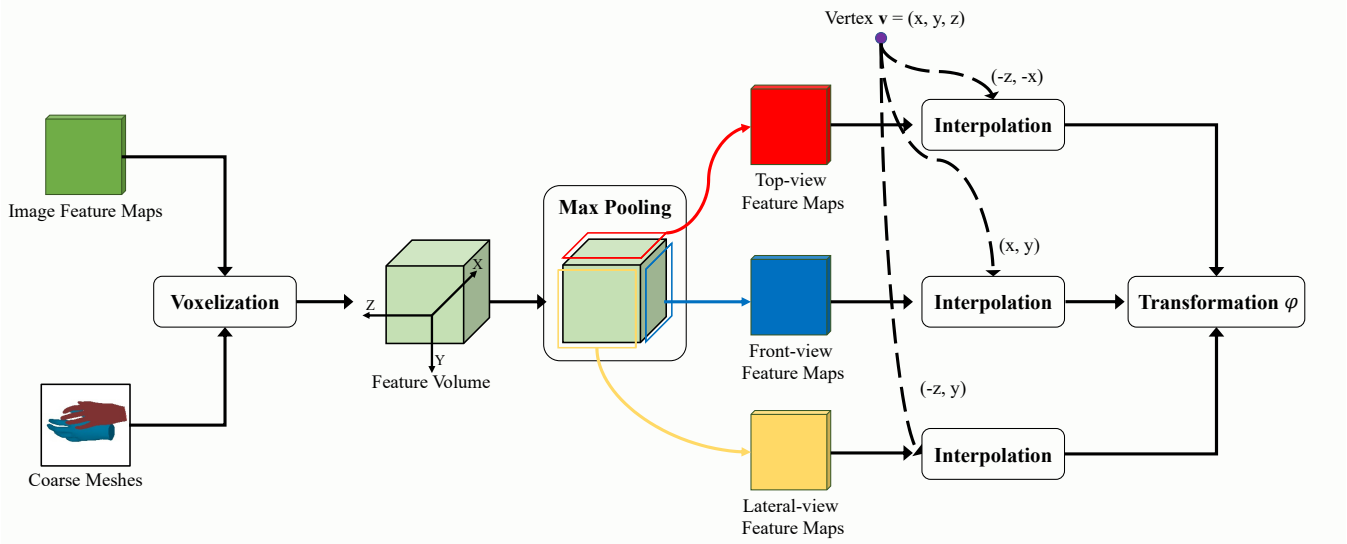

Figure 2: Architecture of the module for per-vertex mean of 3D noise estimation. It leverages three-view feature map disentanglement to alleviate depth ambiguity and occlusions.

from $\mathbf{F}$ to estimate $\boldsymbol{\mu}_{3d}$. The updated 2D coordinate $\mathbf{u}+\boldsymbol{\varepsilon}_{2d}$ is used for feature interpolation. Considering that image-aligned features might be insufficient in tackling depth ambiguity and occlusions, here we propose a simple yet effective method to alleviate this problem. As shown in Figure 2, we first create a voxel grid from the hand meshes (with normalized 3D coordinates), so that features of spatially closed vertices can be aggregated into the same voxel. We then conduct max pooling along each of the three axes of the grid, to obtain the three-view (front, lateral, and top) projections of voxel features.

The above multi-view feature projections help to achieve finer feature disentanglement compared with the single-view representation. Similar to Eq. (3), the per-vertex mean of 3D noise can be estimated via a transformation $\varphi : \mathbb{R}^{3C} \rightarrow \mathbb{R}^3$ as follows:

$$\boldsymbol{\mu}_{3d} = \varphi(\mathbf{f}_{front}, \mathbf{f}_{lateral}, \mathbf{f}_{top}). \tag{4}$$

We also adopt an MLP to implement φ. Note that except for the MANO model, we do not impose any other constraint on the topology of vertices. This allows us to seamlessly adopt architectures that are more complicated than MLPs (e.g., Graph attentions, Li et al. 2022a) to realize ϕ and φ.

Camera Correction. Last but not least, the updated $\mathbf{v}$ and $\mathbf{u}$ are used to refine the intrinsic camera parameters. We adopt the orthographic camera model and hence $\Pi(\mathbf{v}, \mathbf{c})$ can be defined as follows:

$$\Pi(\mathbf{v}, \mathbf{c}) = \mathbf{s}\mathbf{v}(x, y) + \mathbf{t}, \tag{5}$$

where $\mathbf{v}(x, y)$ denotes the x and y coordinates of $\mathbf{v}$. $\mathbf{s} = (s_x, s_y)$ and $\mathbf{t} = (t_x, t_y)$ are the scaling factors and the principle point translations of the camera from the normalized device coordinate space to the image space[1]. Hence $\mathbf{c}$ can be represented as $\mathbf{c} = \begin{bmatrix} s_x, & t_x \\ s_y, & t_y \end{bmatrix}$ and the projection process in can be written as $u_x = s_x v_x + t_x$, $u_y = s_y v_y + t_y$. Substituting Eq. (5) into Eq. (1), we estimate the intrinsic camera parameters by minimizing the following equation:

$$\Sigma_{n=1}^{N} ||\mathbf{u}_n - \mathbf{s}\mathbf{v}_n(x, y) + \mathbf{t}||_2^2 + \xi(||\mathbf{s}||_2^2 + ||\mathbf{t}||_2^2), \tag{6}$$

where $\xi > 0$ is a hyperparameter for regularization. Here we omit $\boldsymbol{\varepsilon}_{3d}$ and $\boldsymbol{\varepsilon}_{2d}$ and use $\mathbf{v}$ and $\mathbf{u}$ to denote the updated vertex and its 2D coordinate. Eq. (6) can be solved via ridge regression (Bishop and Nasrabadi 2006), which has the following closed-form solution:

$$\mathbf{c}' = (\mathbf{V}^\top \mathbf{V} + \xi \mathbf{I})^{-1} \mathbf{V}^\top \mathbf{U}, \tag{7}$$

where $\mathbf{V}$, $\mathbf{U} \in \mathbb{R}^{N \times 2}$ are the matrix representation of all vertices and their 2D coordinates, $\mathbf{I}$ is a 2×2 identity matrix.

Optimization

The proposed network is fully differentiable and is trained via minimizing the following loss function:

$$L = L_{aux} + L_{MANO} + L_v, \tag{8}$$

where L_{aux} denotes the loss for the five auxiliary tasks. L_{aux} is formulated as follows:

$$L_{aux} = \Sigma_{i=1}^{4} \lambda_i ||\mathbf{a}_i - \mathbf{a}_i^g||_1 + \lambda_5 CE(\mathbf{a}_5, \mathbf{a}_5^g), \tag{9}$$

where $||\cdot||_1$ is the l_1 loss and CE is the cross-entropy loss. Variables marked with the superscript g are ground truths. L_{MANO} is the loss term defined on the shape and pose parameters of MANO:

$$L_{MANO} = \lambda_\beta ||\boldsymbol{\beta} - \boldsymbol{\beta}^g||_1 + \lambda_\theta ||\boldsymbol{\theta} - \boldsymbol{\theta}^g||_1. \tag{10}$$

L_v targets at mesh vertices and their 2D projections and is defined as follows:

$$\begin{aligned} L_v = {} & \lambda_{3D} \Sigma_{m=1}^{M} \Sigma_{r=1}^{R} ||\mathbf{v}_{m,r} - \mathbf{v}^g||_1 + \\ & \lambda_{2D} \Sigma_{m=1}^{M} \Sigma_{r=1}^{R} ||\Pi(\mathbf{v}_{m,r}, \mathbf{c}_m) - \Pi(\mathbf{v}_m, \mathbf{c}^g)||_1, \end{aligned} \tag{11}$$

[1]https://pytorch3d.org/docs/cameras

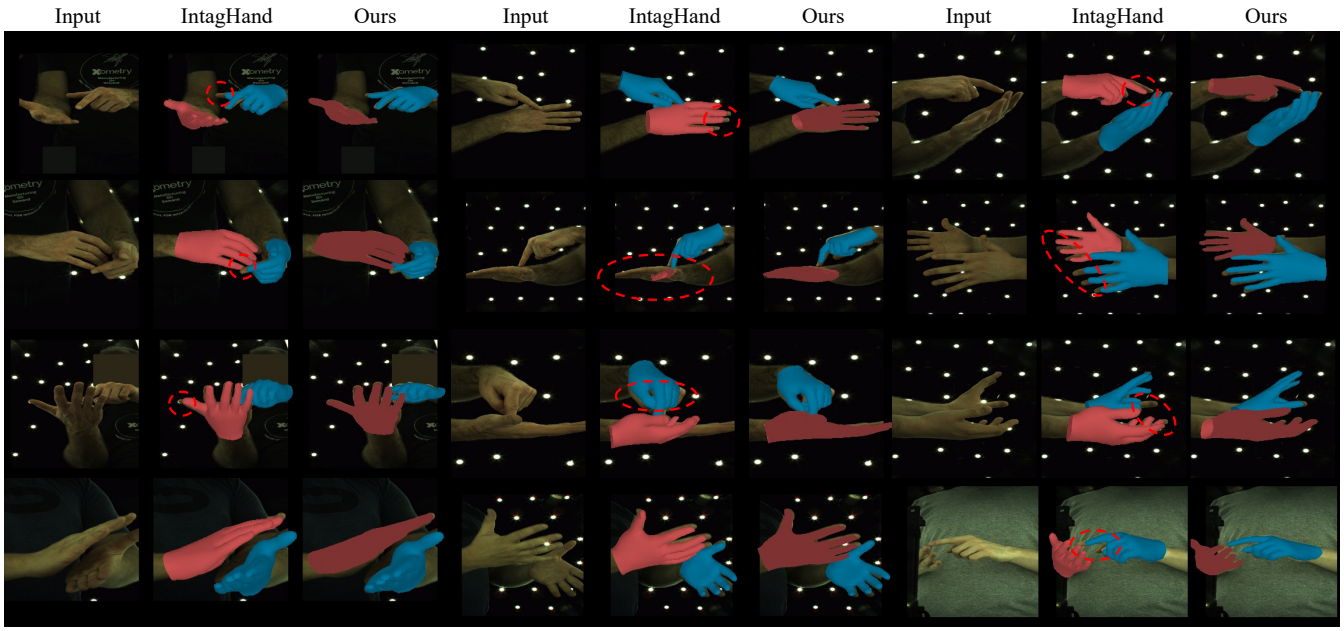

Figure 3: Visual comparison between the proposed method and the state-of-the-art non-parametric method (IntagHand). We highlight a few misaligned parts that are alleviated by the proposed method. Best viewed on screen.

where M is the number of DNE modules and R is the number of random samples in each DNE module. $\lambda_1, ..., \lambda_5, \lambda_\beta, \lambda_\theta, \lambda_{2D}, \lambda_{3D}$ are user-specified loss weights.

Experiment

Dataset and Evaluation Metrics

Dataset Our experiments are conducted on the large-scale Interhand2.6M dataset (Moon et al. 2020), which consists of about 1.3M training images and 0.8M test images. We use all single-hand (SH) and interacting-hand (IH) images in the training set for training. All images are cropped and resized to 256×256 based on the bounding boxes provided by the dataset. We adopt the widely-used mean per-joint position error (MPJPE) and mean per-vertex position error (MPVPE) as the evaluation metrics.

Implementation Details

Our networks are trained with 4 GeForce RTX 4090 graphics cards. We adopt the Adam optimizer (Kingma and Ba 2014) with a batch size of 120 and 30 training epochs. Each epoch takes about five hours. The initial learning rate is 10^{-3} and is scaled by 0.5 after each epoch until it reaches 10^{-6}. Detailed settings of hyperparameters are given in the supplemental material on our project page.

We adopt several data augmentation methods to improve the generalization ability of the proposed network following (Li et al. 2022a), including random image shifting in $[-10, 10]$ pixels, random rotations in $[-90°, 90°]$, random resizing with scaling factor in $[0.9, 1.1]$, horizontal flipping, and adding Gaussian noise $\sim \mathcal{N}(0, 0.3)$ to images.

Method	SH	IH	All
Moon et al. 2020	12.16	16.02	14.22
Zhang et al. 2021	-	13.48	-
Hampali et al. 2022	10.99	14.34	12.78
Meng et al. 2022	8.51	13.12	10.97
Li et al. 2022a	-	10.13	-
Yu et al. 2023b	-	9.68	-
Lee et al. 2023	-	9.68	-
Jiang et al. 2023	8.10	10.96	9.63
Ours-MLP	7.84	10.53	8.78
Our-GraphAttn	7.23	9.55	8.40

Table 1: MPJPE (mm ↓) on Interhand2.6M.

Comparison with State-of-the-arts

Table 1 reports the MPJPE performance of the proposed method against cutting-edge methods. As we have emphasized above, our dual noise estimation process does not have any other constraint besides the MANO topology, and hence it can be combined with different methods. To validate this, we implement two network variants, i.e., the one with the proposed multi-view MLPs (denoted as Ours-MLP) and the other one with graph attentions (Li et al. 2022a, denoted as Ours-GraphAttn). From Table 1 we can see that these two variants achieve state-of-the-art performance by reducing the MPJPE on the Interhand2.6M dataset from 9.63 to 8.78 (Ours-MLP) and 8.40 (Ours-GraphAttn). This indicates that the proposed dual noise estimation is a considerable strategy for current hand mesh recovery methods.

Figure 3 demonstrates the visual examples of interacting hand meshes reconstructed by our method (Ours-MLP) and

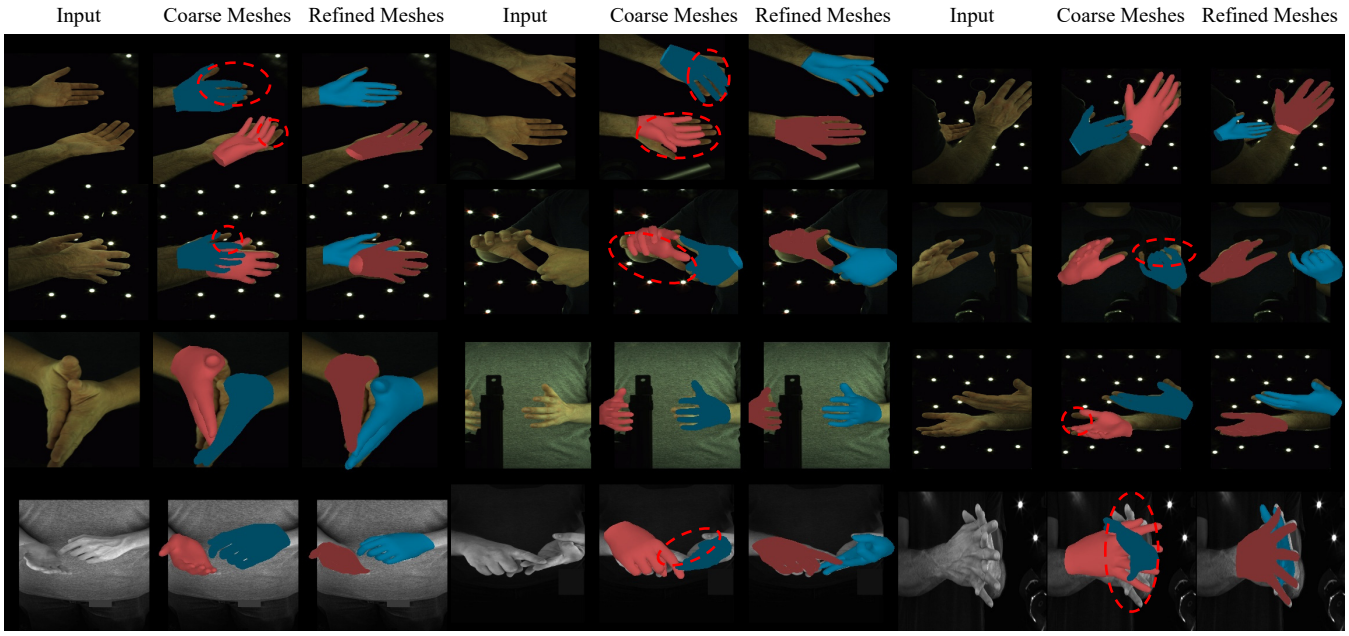

Figure 4: Visual comparison between the coarse meshes obtained by fitting the MANO model and those refined by the proposed method. Best viewed on screen.

the state-of-the-art IntagHand method (Li et al. 2022a). It is reasonable that IntagHand can generate well-aligned results because it is a non-parametric model. However, without the geometric prior of hands, it still exhibits artifacts near the fingertips or inaccurate 2D projections (e.g., the middle example in Figure 3). On the contrary, the proposed method adopts the coarse-to-fine paradigm, in which the coarse but relatively reliable hand meshes are exploited and refined progressively. Hence the reconstruction results of the proposed method are more stable and accurate.

Ablation Study

To provide a comprehensive analysis of the proposed method, we conduct several ablation experiments in this section. Considering that network training on the whole training set of Interhand2.6M is computationally expensive, we only use 10% of training data in our ablation studies, and the whole test set is used for evaluation. Other experimental settings remain unchanged in our ablation studies.

Effects of 3D noise estimation. We first evaluate the effectiveness of 3D noise estimation. This experiment considers three variants of the proposed method: the parametric fitting baseline, the baseline augmented with 3D noise estimation (denoted as $+\varepsilon_{3d}$), and the full DNE module. The experimental results are reported in Table 2. $+\varepsilon_{3d}$ brings notable performance gains to the baseline, as it reduces the MPVPE of the baseline on the sing-hand subset/interacting-hand subset/whole test set from 11.45/14.51/12.53 to 10.02/12.89/11.02. This suggests that the proposed multi-view MLP based 3D noise estimation module is effective. The full DNE module further improves the performance of the baseline and outperforms the variant with 3D noise estimation only on all test subsets. We owe this to the proposed 2D noise estimation and camera correction methods, as they help to obtain features that are more aligned with images.

Method	MPVPE-SH	MPVPE-IH	MPVPE-All
Baseline	11.45	14.51	12.53
$+\varepsilon_{3d}$	10.02	12.89	11.02
Full DNE	9.89	12.68	10.87

Table 2: Ablation study on 3D noise estimation.

Effects on 2D noise estimation We are also interested in the effects of each proposed component on 2D predictions, as generating well-aligned results is one of the major goals of this paper. Besides the three variants used in the previous experiment, we consider another variant that estimates 2D noise only (denoted as $+\varepsilon_{2d}$). In this ablation study, we use 2D MPVPE as the metric, and the results of these four variants are summarized in Table 3. We observe that both $+\varepsilon_{2d}$ and $+\varepsilon_{3d}$ outperform the baseline and the performance margin of the latter is more obvious. This is reasonable, as the conceptual field of a vertex in the 3D noise estimation process (max pooling on three-view feature maps) is larger than that in the 2D case (only features interpolated with the projected and the regressed coordinates). Consequently, the 3D noise estimation module can leverage more information for refinement. Utilizing 2D and 3D noise estimation jointly achieves the best performance. The full DGE module reduces the MPVPE of the baseline by more than $30\%/25\%/27\%$ on the three test sets, respectively. These results are sufficient to validate that the proposed method

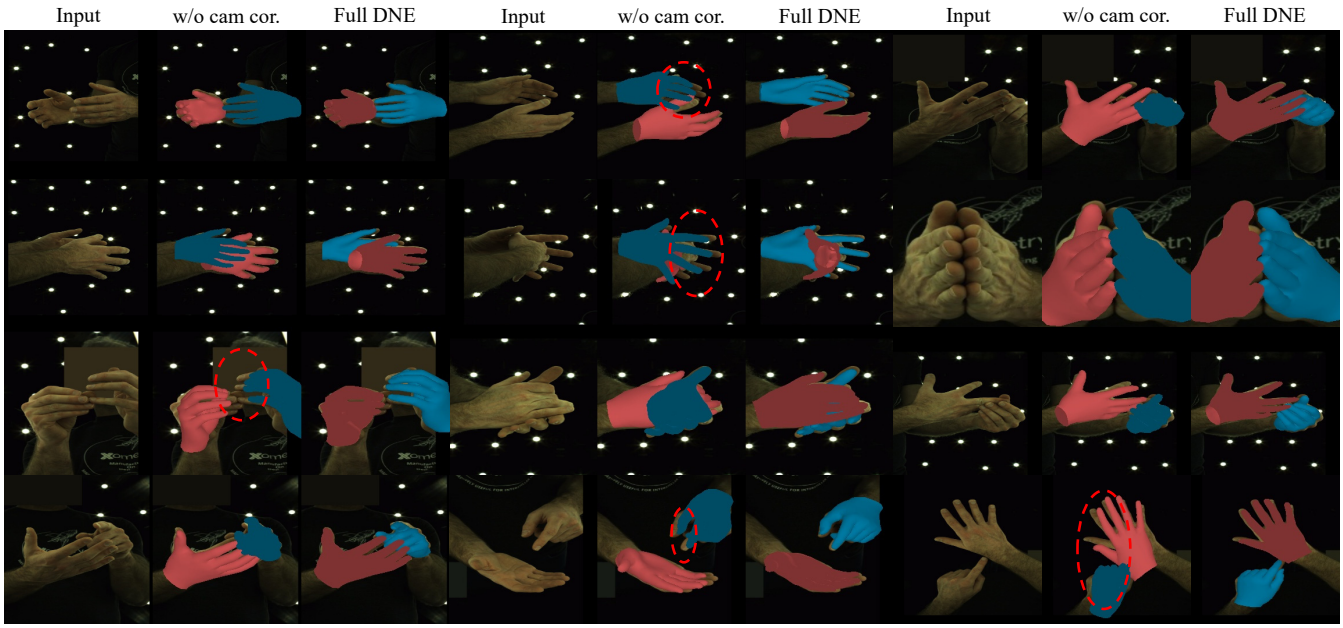

Figure 5: Visual comparison between the DNE module with and without the camera correction. Best viewed on screen.

Method	MPVPE-SH	MPVPE-IH	MPVPE-All
Baseline	13.92	12.18	13.31
$+\varepsilon_{2d}$	12.93	11.54	12.45
$+\varepsilon_{3d}$	10.20	9.96	10.12
Full DNE	9.64	9.06	9.44

Table 3: Ablation study of the proposed modules on 2D predictions. The MPVPE is calculated with 2D coordinates.

Metric	Baseline	M = 1	M = 3
MPVPE-SH	11.45	10.66	9.89
MPVPE-IH	14.15	13.52	12.68
MPVPE-All	12.53	11.67	10.87
MPJPE-SH	11.15	10.33	9.69
MPJPE-IH	14.10	13.16	12.41
MPJPE-All	12.19	11.33	10.64

Table 4: Ablation study on the number of DNE modules.

boosts the performance of the baseline in image-plane alignment successfully.

Effects on the number of DNE modules. At last, as we have mentioned above, the mesh refinement process can be conducted progressively via multiple DNE modules. To verify this, we compare the performance of using a single DNE module ($M = 1$) and that of using three DNE modules ($M = 3$). The experimental results are reported in Table 4. These results validate that higher performance gains can be obtained with more DNE modules.

Visual comparison of mesh refinement. The visual comparison between the meshes before and after refinements is shown in Figure 4. We can see that the meshes refined by the proposed method are more accurate. This again validates that leveraging the advantages of parametric models and non-parametric models is considerable.

Visual comparison of camera correction. We also compare the reconstruction results of the proposed with and without the camera correction in Figure 5. From this figure, we can see that leveraging camera correction helps to generate better results.

Conclusion

In this paper, we propose a novel method leveraging dual noise estimation to recover 3D hand meshes from single-view images. Our method models the noise of mesh vertices and their projections on the image plane in a unified probabilistic model. We implement the proposed framework via an end-to-end trainable network with two effective estimation branches. Furthermore, our framework can also refine the intrinsic camera parameters efficiently via ridge regression. Consequently, our method can generate hand meshes that are well-aligned with images. Our experiments and ablation studies on the Interhand2.6M dataset demonstrate the effectiveness of our method.

Our current method is not designed especially for single-hand or interacting-hand images. In the future, we plan to incorporate a cross-hand noise model to further enhance the proposed method. We will also consider other association strategies for vertices and their 2D coordinates, such as differential neural rendering.

Acknowledgments

This work was supported in part by National Key R&D Program of China under Grant No. 2020AAA0109700, Guangdong Outstanding Youth Fund (Grant No.

2021B1515020061), National Natural Science Foundation of China (NSFC) under Grant No. 61976233, No. 92270122, No. 62372482 and No. 61936002, Mobility Grant Award under Grant No. M-0461, Shenzhen Science and Technology Program (Grant No. RCYX20200714114642083), Shenzhen Science and Technology Program (Grant No. GJHZ20220913142600001), Nansha Key R&D Program under Grant No.2022ZD014 and Sun Yat-sen University under Grant No. 22lgqb38 and 76160-12220011.

References

Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Chen, X.; Liu, Y.; Dong, Y.; Zhang, X.; Ma, C.; Xiong, Y.; Zhang, Y.; and Guo, X. 2022a. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 20544–20554.

Chen, X.; Wang, B.; and Shum, H.-Y. 2023. Hand avatar: Free-pose hand animation and rendering from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8683–8693.

Chen, Z.; Hasson, Y.; Schmid, C.; and Laptev, I. 2022b. Alignsdf: Pose-aligned signed distance fields for hand-object reconstruction. In *Proceedings of the European Conference on Computer Vision*, 231–248.

Cheng, J.; Wan, Y.; Zuo, D.; Ma, C.; Gu, J.; Tan, P.; Wang, H.; Deng, X.; and Zhang, Y. 2022. Efficient virtual view selection for 3D hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 419–426.

Corona, E.; Hodan, T.; Vo, M.; Moreno-Noguer, F.; Sweeney, C.; Newcombe, R.; and Ma, L. 2022. Lisa: Learning implicit shape and appearance of hands. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 20533–20543.

Deng, X.; Zuo, D.; Zhang, Y.; Cui, Z.; Cheng, J.; Tan, P.; Chang, L.; Pollefeys, M.; Fanello, S.; and Wang, H. 2022. Recurrent 3D hand pose estimation using cascaded pose-guided 3D alignments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 932–945.

Güler, R. A.; Neverova, N.; and Kokkinos, I. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7297–7306.

Guo, Z.; Zhou, W.; Wang, M.; Li, L.; and Li, H. 2023. HandNeRF: Neural Radiance Fields for Animatable Interacting Hands. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21078–21087.

Hampali, S.; Sarkar, S. D.; Rad, M.; and Lepetit, V. 2022. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11090–11100.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Jiang, C.; Xiao, Y.; Wu, C.; Zhang, M.; Zheng, J.; Cao, Z.; and Zhou, J. T. 2023. A2J-Transformer: Anchor-to-Joint Transformer Network for 3D Interacting Hand Pose Estimation from a Single RGB Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8846–8855.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv:1312.6114*.

Lee, J.; Sung, M.; Choi, H.; and Kim, T.-K. 2023. Im2Hands: Learning Attentive Implicit Representation of Interacting Two-Hand Shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21169–21178.

Li, J.; Bian, S.; Xu, C.; Chen, Z.; Yang, L.; and Lu, C. 2023. HybrIK-X: Hybrid Analytical-Neural Inverse Kinematics for Whole-body Mesh Recovery. *arXiv preprint arXiv:2304.05690*.

Li, M.; An, L.; Zhang, H.; Wu, L.; Chen, F.; Yu, T.; and Liu, Y. 2022a. Interacting attention graph for single image two-hand reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2761–2770.

Li, Z.; Liu, J.; Zhang, Z.; Xu, S.; and Yan, Y. 2022b. Cliff: Carrying location information in full frames into human pose and shape estimation. In *Proceedings of the European Conference on Computer Vision*, 590–606.

Lin, K.; Lin, C.-C.; Liang, L.; Liu, Z.; and Wang, L. 2022. MPT: Mesh Pre-Training with Transformers for Human Pose and Mesh Reconstruction. *arXiv:2211.13357*.

Lin, K.; Wang, L.; and Liu, Z. 2021a. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1954–1963.

Lin, K.; Wang, L.; and Liu, Z. 2021b. Mesh graphormer. In *Proceedings of the IEEE International Conference on Computer Vision*, 12939–12948.

Meng, H.; Jin, S.; Liu, W.; Qian, C.; Lin, M.; Ouyang, W.; and Luo, P. 2022. 3d interacting hand pose estimation by hand de-occlusion and removal. In *Proceedings of the European Conference on Computer Vision*, 380–397.

Moon, G. 2023. Bringing Inputs to Shared Domains for 3D Interacting Hands Recovery in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 17028–17037.

Moon, G.; Yu, S.-I.; Wen, H.; Shiratori, T.; and Lee, K. M. 2020. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Proceedings of the European Conference on Computer Vision*, 548–564.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 165–174.

Pavlakos, G.; Choutas, V.; Ghorbani, N.; Bolkart, T.; Osman, A. A.; Tzionas, D.; and Black, M. J. 2019. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10975–10985.

Ren, P.; Chen, Y.; Hao, J.; Sun, H.; Qi, Q.; Wang, J.; and Liao, J. 2023. Two Heads Are Better than One: Image-Point Cloud Network for Depth-Based 3D Hand Pose Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2163–2171.

Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6): 1–17.

Shetty, K.; Birkhold, A.; Jaganathan, S.; Strobel, N.; Kowarschik, M.; Maier, A.; and Egger, B. 2023. PLIKS: A Pseudo-Linear Inverse Kinematic Solver for 3D Human Body Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 574–584.

Tang, X.; Wang, T.; and Fu, C.-W. 2021. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, 11698–11707.

Tian, Y.; Zhang, H.; Liu, Y.; and Wang, L. 2023. Recovering 3d human mesh from monocular images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yu, Z.; Huang, S.; Fang, C.; Breckon, T. P.; and Wang, J. 2023a. ACR: Attention Collaboration-based Regressor for Arbitrary Two-Hand Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12955–12964.

Yu, Z.; Li, C.; Yang, L.; Zheng, X.; Mi, M. B.; Lee, G. H.; and Yao, A. 2023b. Overcoming the Trade-off Between Accuracy and Plausibility in 3D Hand Shape Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 544–553.

Zhang, B.; Wang, Y.; Deng, X.; Zhang, Y.; Tan, P.; Ma, C.; and Wang, H. 2021. Interacting two-hand 3d pose and shape reconstruction from single color image. In *Proceedings of the IEEE International Conference on Computer Vision*, 11354–11363.

Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; and Li, H. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5745–5753.