H2ONet: Hand-Occlusion-and-Orientation-aware Network for Real-time 3D Hand Mesh Reconstruction

Hao Xu[1,2] Tianyu Wang[1] Xiao Tang[1] Chi-Wing Fu[1,2,3]
[1]Department of Computer Science and Engineering [2]Institute of Medical Intelligence and XR
[3]Shun Hing Institute of Advanced Engineering
The Chinese University of Hong Kong
{xuhao,wangty,cwfu}@cse.cuhk.edu.hk, xtang@link.cuhk.edu.hk

Abstract

Real-time 3D hand mesh reconstruction is challenging, especially when the hand is holding some object. Beyond the previous methods, we design H2ONet to fully exploit non-occluded information from multiple frames to boost the reconstruction quality. First, we decouple hand mesh reconstruction into two branches, one to exploit finger-level non-occluded information and the other to exploit global hand orientation, with lightweight structures to promote real-time inference. Second, we propose finger-level occlusion-aware feature fusion, leveraging predicted finger-level occlusion information as guidance to fuse finger-level information across time frames. Further, we design hand-level occlusion-aware feature fusion to fetch non-occluded information from nearby time frames. We conduct experiments on the Dex-YCB and HO3D-v2 datasets with challenging hand-object occlusion cases, manifesting that H2ONet is able to run in real-time and achieves state-of-the-art performance on both the hand mesh and pose precision. The code will be released on GitHub.

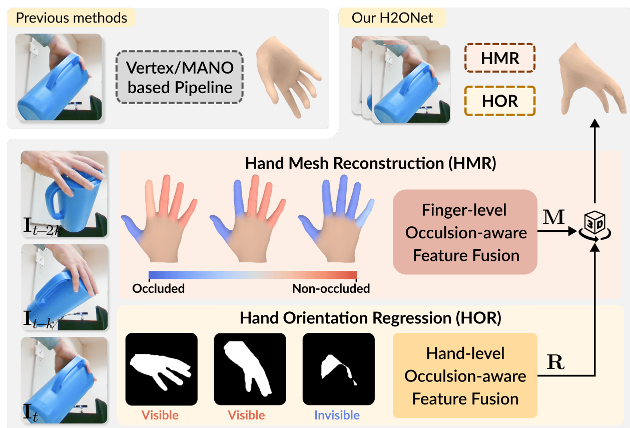

Figure 1. Structural comparison between our H2ONet and previous methods. We decouple 3D hand mesh reconstruction into two branches, one to reconstruct the hand mesh at canonical pose $\mathbf{M}$ and the other to regress the global hand orientation $\mathbf{R}$, such that we can fuse finger- and hand-level occlusion-aware features from multiple frames to better exploit the non-occluded information.

1. Introduction

Estimating 3D hand meshes from RGB images is a fundamental task useful for many applications, *e.g.*, augmented reality [17, 52], behavior understanding [26, 44], *etc.* To support these applications, user experience is very important, so the reconstruction should be accurate and robust, as well as fast, *i.e.*, real-time. Despite the promising results achieved by the recent works, it is still very challenging to simultaneously meet all the requirements, particularly when the hand is severely occluded, *e.g.*, holding some object.

Several recent methods are proposed for 3D hand mesh reconstruction from a single RGB image [13, 15, 31, 38–42, 45]. To alleviate the negative effect of occlusion, some try to extract occlusion-robust features by adopting the spatial attention mechanism applied in 3D hand pose estimation [14, 65, 68]. When the amount of occlusion is small, focusing more on non-occluded regions can help improve the network performance. However, the performance would largely drop when the occluded regions dominate, implying that relying solely on the prior information of hand shape and pose is insufficient. Besides, the attention mechanism brings extra computation and memory overhead. Though recent methods [8, 52] adopt lightweight frameworks for real-time inference, the influence of occlusion is ignored.

On the other hand, some recent works [18,57] start to explore multi-frame RGB images as input for 3D hand mesh reconstruction. SeqHAND [57] integrates LSTM as a feature extractor to memorize the hand motion over consecutive frames. Liu *et al.* [35] constrain the smoothness of hand shape and pose by designing inter-frame losses. Yet, they do not have specific designs to explicitly deal with the occlusion. Hasson *et al*. [18] leverages an optical-flow-guided strategy to promote photometric consistency. Nevertheless, the extra information is limited, as they use only two adjacent frames. Also, though multi-frame inputs provide more information, it is non-trivial to effectively extract and fuse multi-frame features to improve the reconstruction quality.

In this paper, we present **H2ONet**, a **H**and-**O**cclusion-and-**O**rientation-aware **N**etwork, aiming at exploiting non-occluding information from multi-frame images to reconstruct the 3D hand mesh. Our goal is to meet the requirements of (i) effectively utilizing the inter-frame information and (ii) explicitly alleviating the interference of occlusion.

First, as the hand orientation information and hand shape information are mixed in feature space, it is hard to directly fuse features from multiple frames. To better exploit useful information, we decouple hand mesh reconstruction into two tasks: one for hand mesh reconstruction at the canonical pose and the other for hand orientation regression, as shown in Fig. 1. The key advantages are that we can better fuse multi-frame features without considering hand orientation differences, and it enables us to apply strategies to alleviate the ill-posed issue in estimating hand orientation.

Second, to handle self and object occlusions on the hand, we propose to exploit non-occluding information spatially across fingers and temporally across frames. For the former, we design finger-level occlusion-aware feature fusion that leverages predicted finger-level occlusion probabilities to guide the adaptive fusion of per-finger features from multiple frames. For the latter, we design hand-level occlusion-aware feature fusion that catches auxiliary global information over frames guided by the hand-level occlusions.

In summary, our main contributions are:

- We design the hand-occlusion-and-orientation-aware network named H2ONet with a two-branch architecture to efficiently and effectively exploit non-occluding information from multiple frames.
- We formulate finger-level occlusion-aware feature fusion and hand-level occlusion-aware feature fusion modules. The former aggregates non-occluded finger-level information from multiple frames to promote hand shape reconstruction, whereas the latter alleviates the ill-posed issue when estimating the global hand orientation in case the hand is temporarily occluded.
- Through qualitative and quantitative comparisons on two datasets with severe hand occlusions, we show that H2ONet achieves state-of-the-art performance.

2. Related Work

Single-frame 3D hand mesh reconstruction. According to the type of input, single-frame methods can be divided into two categories: depth-based and RGB-based.

Early depth-based methods [29, 51, 53] fit a deformable hand mesh to the depth image via an iterative optimization. Recent deep-learning-based methods [36, 43, 56] utilize CNN, as a powerful feature extractor or parameter regressor, to improve performance.

For the RGB-based methods, most works [1–4, 10, 20, 25, 35, 59, 61–64, 67, 69] directly estimate the MANO parameters [46]. Other representations include voxel grid [24, 39, 42, 58], implicit function [37], UV map [7], and vertices [9, 15, 31, 33, 34]. Nevertheless, most existing works do not explicitly take occlusion into consideration and yield unsatisfactory results when the hand is occluded by itself or by other objects. Overall, the most related and recent work is HandOccNet [45]. It handles the occlusion in hand-object datasets by formulating self- and cross-attention modules. Yet, when the occlusions dominate the image space, estimating hand pose and shape can become severely ill-posed.

In this work, we propose to fully exploit non-occluding information, first by explicitly estimating finger-level occlusions and further by fusing hand-level information over multiple frames, to address the occlusion issue.

Multi-frame 3D hand mesh reconstruction. Some recent works start to exploit multi-frame inputs for 3D hand mesh reconstruction. SeqHAND [57] adopts convolution-LSTM [49]. Chen *et al*. [6] design a self-supervised method that utilizes bi-directional hand consistency over sequential frames. Overall, existing works focus mainly on improving temporal consistency and ignore the occlusion issue when fusing information from multiple frames. Besides, they can hardly maintain a fast inference, as they either use lots of frames as input or are based on optimization. In this work, we predict finger-level occlusion probabilities to guide the information fusion adaptively over multiple time frames.

3D hand-object mesh reconstruction. Joint reconstruction of hands and objects has been receiving increasing attention [4, 18, 19, 60]. Hasson *et al*. [18] assume known object models and leverages photometric consistency between adjacent frames to improve hand-object reconstructions. Later, they design an optimization method [19] to incorporate contact losses to encourage contact surfaces and penalize penetrations between hand and object from videos. Karunratanakul *et al*. [27] propose an implicit representation for hand in the form of sign distance fields. Recent work [54] also adopts a collaborative learning framework to implicitly model mutual occlusions via associative loss.

Occlusion-aware 3D human pose estimation. There are three common approaches to handling occlusions in human pose estimation. First, we can simulate the occlusions in the data augmentation, *e.g*., by covering parts of the image with black regions [48], by copying patches of the background to paste on non-occluded human regions [28], and by randomly setting some values of the estimated 2D heatmaps

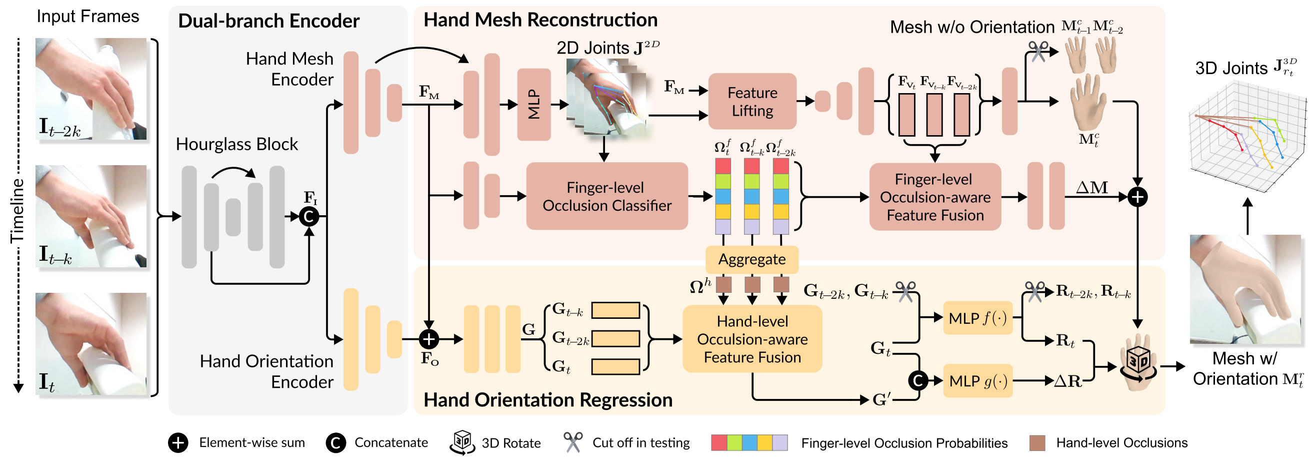

Figure 2. The H2ONet architecture: (i) the dual-branch encoder extracts general and task-specific features; (ii) the hand mesh reconstruction module focuses on constructing hand meshes at canonical poses by predicting finger-level occlusions and taking these information as guidance to fuse the multi-frame information; and (iii) the hand orientation regression module predicts the global hand orientation using the predicted hand-level visibility (equivalently, the occlusion) across multiple frames for support the estimation of the hand orientation.

to zero [11]. Though these techniques can bring some improvement, it is still hard to resolve real occlusions due to their domain gap from the synthetic occlusions.

Another approach is to more effectively utilize the spatial information, *e.g.*, [14, 65, 68] use the attention mechanism [55] to enhance the features of the non-occluded human regions and spread the information to the missing parts. Yet, the attention mechanism increases the computational overhead, making it hard to give real-time performance.

Last, some works compensate for the occluded information by utilizing temporal information from videos. Cheng *et al.* [12] lift the estimated occlusion-labeled 2D pose sequence to 3D through a temporal CNN. Yet, they mainly consider the bare human situation, without severe occlusions, as in the hand-object datasets.

3. Method

3.1. Overview

Fig. 2 shows the pipeline of H2ONet for reconstructing 3D hand meshes, particularly when the hand is holding and manipulating some object. Three RGB frames are first fed into the dual-branch encoder to extract task-specific information for the two subsequent modules to process.

- The hand mesh reconstruction module first predicts 2D joint coordinates and finger-level occlusions, then utilizes them to fetch and fuse features and reconstruct hand meshes by a decoder (Sec. 3.2).
- Concurrently, the hand orientation regression module aggregates global features with the hand-level occlusion predictions as the guidance and regresses the global hand orientation (Sec. 3.3). Details of the loss functions are presented in Sec. 3.4.

Dual-branch Encoder. We design the dual-branch encoder to extract features with two encoder heads that share general features extracted from the former network. Specifically, given input frames $\mathbf{I} = \{\mathbf{I}_{t-ik} : i \in \{0, 1, 2\}\}$, where $\mathbf{I}_t$ denotes the current frame and $\mathbf{I}_{t-k}, \mathbf{I}_{t-2k}$ denote two previous frames (k is frame gap), an hourglass block is first adopted to extract multi-scale refined feature $\mathbf{F_I}$. To save the computation, we replace the commonly-used ResNet [21] blocks with the computationally-efficient blocks proposed by SENet [23] and MobileNet [22], following [8]. After that, we feed $\mathbf{F_I}$ into (i) the hand mesh encoder to produce feature $\mathbf{F_M}$ for reconstructing the 3D hand mesh; and (ii) the hand orientation encoder to produce feature $\mathbf{F_O}$ for regressing the global hand orientation.

3.2. Hand Mesh Reconstruction

When the hand is manipulating an object, different fingers may switch between occluding and non-occluding states as the hand moves and rotates. Reconstructing the occluded fingers is ill-posed, so the predictions on them are less reliable than those of the non-occluded ones. Based on the assumption that the transformation among nearby frames is mostly rigid, we propose to exploit the finger-level occlusion probabilities as a guide to help fuse and utilize the finger-level knowledge adaptively from the non-occluded fingers across the input frames.

Preparing finger-level occlusion labels. Our method explicitly predicts finger-level occlusions, but the hand-object datasets provide ground truths only on hand segmentation. So, we design a method to automatically prepare occlusion labels using the provided ground truths for model training.

Fig. 3(a) illustrates the procedure. Given a ground-truth hand mesh, we first locate vertices associated with different hand parts, *i.e.*, the five fingers and the palm, which are distinguished by different colors. Then, we render the hand mesh in colors (Fig. 3(a-ii)) and the 2D vertices in colors (Fig. 3(a-v)) in the image view. By masking the rendered hand mesh image using the hand segmentation ground truth (Fig. 3(a-iii)), we can remove the hand regions that are occluded by the objects (Fig. 3(a-iv)). Next, for each finger, we can check each of its vertices (Fig. 3(a-v)) against the masked image (Fig. 3(a-iv)) to see if the vertex is occluded in the image view either caused by hand itself or object. If the number of occluded vertices is above a threshold (set as 50), the associated finger is regarded as "occluded." For the global hand occlusion, the hand is regarded as "occluded" if the proportion of occluded hand pixels is above 50%.

Finger-level occlusions prediction. To predict the occlusion of each finger, we should examine the features of the pixels associated with each finger and filter out irrelevant background features. However, hand segmentation requires per-pixel prediction, which reduces computational efficiency. So, we directly utilize the predicted 2D joint coordinates and fetch the joint features as the input to the finger-level occlusion classifier, as shown in Fig. 3(b).

Another concern is that some 2D joints may lie too close to each other, so using a shared network to predict occlusions only from the feature of individual fingers may lead to confusion. Therefore, we formulate a base MLP to extract the global feature from all the finger features and adopt five lightweight head MLPs to further predict the occlusion probability of each finger independently.

Occlusion-aware mesh reconstruction. Given feature $\mathbf{F_M}$ from the dual-branch encoder, we adopt the 2D joint coordinates regression and 2D-to-3D feature lifting of [8]. As Fig. 2 shows, $\mathbf{F_M}$ is first expanded by several convolutional layers and fed into an MLP to regress the normalized 2D joint positions $\mathbf{J}^{2D}$. Then, we fetch 2D joints feature $\mathbf{F_J}$ from $\mathbf{F_M}$ indexed by $\mathbf{J}^{2D}$, and project it to the low-resolution 3D vertices feature $\mathbf{F_V}$ by the learnable projection matrix $\mathbf{P}$, *i.e.*,

$$\mathbf{F_V} = \mathbf{P} \cdot \mathbf{F_J}, \tag{1}$$

where $\mathbf{F_V} \in \mathbb{R}^{n \times c}$, $\mathbf{P} \in \mathbb{R}^{n \times m}$, $\mathbf{F_J} \in \mathbb{R}^{m \times c}$; $m = 21$ is the number of joints; c is the feature channel size; and n is the number of vertices in the down-sampled hand mesh.

Furthermore, the lifted vertices feature $\mathbf{F_V}$ is fed into a four-layer SpiralConv-based [32] decoder to upsample hierarchically by a factor of 2 for each layer and finally regress the original-resolution mesh vertices. Since the feature before the last decoder layer should contain the least orientation information and already has the same number of vertices as the output hand mesh, we adopt it as the input to fuse the information from all frames. For clarity, we use $\mathbf{F}_{\mathbf{V}_{t-ik}}$, $i \in \{0, 1, 2\}$ to denote the decoded features of the $(t-ik)$-th frame. Specifically, we split a branch before the last layer of the decoder and assign $\mathbf{F}_{\mathbf{V}_{t-ik}}$ to different fingers based on the predefined vertex indices. Since non-occluded fingers are more informative, vertex features belonging to each finger f in the $(t-ik)$-th frame is first weighted by $\mathbf{W}^f_{t-ik}$, which is computed by operating softmax on the corresponding predicted finger-level occlusion possibilities $\mathbf{\Omega}^f_{t-ik}$ along the frame dimension, *i.e.*,

$$\mathbf{W}^f_{t-ik} = \frac{e^{\omega^f_{t-ik}}}{\sum_{j=0}^{N-1} e^{\omega^f_{t-jk}}}, \quad \mathbf{\Omega}^f_{t-ik} = \{\omega^d_{t-ik} : d \in \{0, ..., D-1\}\}, \tag{2}$$

where $N = 3$ is the number of input frames; $D = 5$ is the number of fingers; and ω^d_{t-ik} means the d-th finger occlusion probability of the $(t-ik)$-th frame. Then, for all fingers, the hybrid feature $\mathbf{F}'$ is obtained by

$$\mathbf{F}' = \sum_{i=0}^{N-1} \mathbf{W}_{t-ik} \cdot \mathbf{F}_{\mathbf{V}_{t-ik}}. \tag{3}$$

Last, we further fuse $\mathbf{F}'$ adaptively via MLP $q(\cdot)$ and generate per-vertex offset $\Delta\mathbf{M}$ of the current frame, *i.e.*,

$$\begin{aligned} \mathbf{M}^c_{t-ik} &= p\left(\mathbf{F}_{\mathbf{V}_{t-ik}}\right), i \in \{0, 1, 2\}. \\ \Delta\mathbf{M} &= q\left(\mathbf{F}'\right), \quad \mathbf{M}^c_t = \Delta\mathbf{M} + \mathbf{M}^c_t, \end{aligned} \tag{4}$$

where $p(\cdot)$ denotes the SpiralConv-based decoder and $\mathbf{M}^c_{t-ik}$ denotes the predicted mesh vertices at the canonical pose from the input frame $\mathbf{I}_{(t-ik)}$.

3.3. Hand Orientation Regression

Considering that the current hand can be severely occluded at times, although the object pose can provide certain cues, it is not always reliable. Therefore, we propose to

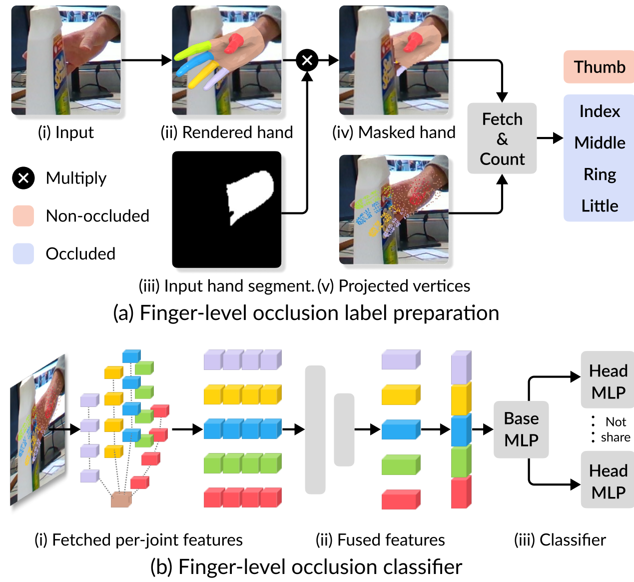

Figure 3. (a) The procedure of preparing finger-level occlusion labels. (b) The architecture of the finger-level occlusion classifier.

utilize the information from the previous frames to complement the information from the current frame. The key idea is to exploit the nearest non-occluded frame to help correct the estimated orientation of the current hand, if it is mostly occluded in the view. To this end, for all frames, we first encode and flatten the extracted feature $\mathbf{F_O}$ to obtain global features $\mathbf{G}$ for all frames. Then, we adopt the hand-level occlusions to exploit information from other frames.

Hand-level occlusion prediction. We tried to predict hand-level occlusion $\mathbf{\Omega}^h$ in two different ways. First, we added another branch to the finger-level occlusion classifier, whose input already provides the global information of the entire hand. Though the classification accuracy of this approach can reach over 90% on all datasets, we found in the experiments that the orientation regression precision of the misclassified cases will be heavily affected.

The other approach is to directly utilize the predicted finger-level occlusion probabilities and regard the hand as occluded only if all the fingers are occluded. Formally, for all frames, we have their global hand-level occlusions

$$\mathbf{\Omega}^h_{t-ik} = \prod_{d=0}^{D-1} \text{argmax}[\omega^d_{t-ik}],\ i \in \{0,1,2\}, \tag{5}$$

where $\mathbf{\Omega}^h_{t-ik}$ is the predicted hand-level occlusion of the $(t-ik)$-th frame and ω^d_{t-ik} is the predicted d-th finger occlusion probability; see Sec. 3.2. This implies a more strict condition that can alleviate the effect of misclassification.

Occlusion-aware orientation regression. The hand-level occlusion-aware feature fusion mainly helps to alleviate the ill-posed issue when the hand is highly (or fully) occluded. We design a frame-by-frame enquiry strategy to fuse the multi-frame features guided by their associated hand-level occlusion predictions $\mathbf{\Omega}^h$.

In detail, we check $\mathbf{\Omega}^h$ frame by frame in a reverse chronological order (from $\mathbf{I}_t$ to $\mathbf{I}_{t-2k}$), and fetch the feature $\mathbf{G}_{t-ik}$ as the output feature $\mathbf{G}'$ if the $(t-ik)$-th frame is non-occluded. Formally, we split the computation of $\mathbf{G}'$ into two steps. First, for the $(t-ik)$-th frame, we compute

$$\mathbf{G}'_{t-ik} = \begin{cases} \mathbf{\Omega}^h_t \cdot \mathbf{G}_t + \left(\prod_{j=0}^{N-1}\left(1-\mathbf{\Omega}^h_{t-jk}\right)\right)\cdot \mathbf{G}_t, & \text{if } i=0 \\ \left(\prod_{j=0}^{i-1}\left(1-\mathbf{\Omega}^h_{t-jk}\right)\right)\cdot \mathbf{\Omega}^h_{t-ik}\cdot \mathbf{G}_{t-ik}, & \text{otherwise.} \end{cases} \tag{6}$$

Here, $\mathbf{\Omega}^h_{t-ik}=0$ indicates that the $(t-ik)$-th frame is predicted as "occluded." Then, we obtain $\mathbf{G}'$ by $\sum_{i=0}^{N-1}\mathbf{G}'_{t-ik}$. Note that we set $\mathbf{G}'=\mathbf{G}_t$ if all frames are occluded.

To further obtain the hand orientation, we feed the original features $\mathbf{G}$ to an MLP $f(\cdot)$ to regress the rotation 6D [66] parameters for all frames, which are further transferred to form the 3D rotation matrix $\mathbf{R}$. Meanwhile, since refining the current hand orientation not only needs the information from the auxiliary frames but also the state of the current hand itself, we feed both the complement feature $\mathbf{G}'$ and the feature of the current frame $\mathbf{G}_t$ into another MLP $g(\cdot)$ to regress the rotation offset $\Delta\mathbf{R}$ and then update $\mathbf{R}_t$:

$$\begin{aligned} \mathbf{R}_{t-ik} &= f\left(\mathbf{G}_{t-ik}\right), i=0,...,N-1. \\ \Delta\mathbf{R} &= g\left(\text{cat}\left[\mathbf{G}',\mathbf{G}_t\right]\right), \quad \mathbf{R}_t = \Delta\mathbf{R}\,\mathbf{R}_t, \end{aligned} \tag{7}$$

where cat $[.\,,.]$ denotes the concatenate operation.

Finally, we apply $\mathbf{R}_{t-ik}$ on $\mathbf{M}^c_{t-ik}$ to obtain the rotated hand mesh $\mathbf{M}^r_{t-ik}$ in the camera coordinate system. Also, we calculate the 3D joint coordinates $\mathbf{J}^{3D}_{c_{t-ik}}$ at the canonical pose and $\mathbf{J}^{3D}_{r_{t-ik}}$ in the camera coordinate system by multiplying $\mathbf{M}^c_{t-ik}$ and $\mathbf{M}^r_{t-ik}$, respectively, with a vertex-to-joint regression matrix, which is predefined by MANO and set to be learnable to better fit the hands to the specific dataset. Note that the 3D joint coordinates and vertices of the previous frames are only used to compute losses. We cut off their network structures at test time for efficiency.

3.4. Loss Functions

For 3D hand mesh reconstruction, we adopt several common 2D and 3D loss functions. First, the L1 loss is applied to constrain the distance between the predicted hand mesh and the ground truth at the canonical pose. The same supervision is also applied to the 3D joint coordinates. Concurrently, the 2D joint coordinates are supervised by the normalized ground truths. Formally, we have the 3D mesh loss $\mathcal{L}^c_{\mathcal{M}}$, 3D joint loss $\mathcal{L}^c_{3D}$, and 2D joint loss $\mathcal{L}_{2D}$ as

$$\begin{gathered} \mathcal{L}^c_{\mathcal{M}} = \sum_{i=0}^{N-1}||\mathbf{M}^c_{t-ik} - \hat{\mathbf{M}}^c_{t-ik}||_1,\ \mathcal{L}^c_{\mathcal{J}_{3D}} = \sum_{i=0}^{N-1}||\mathbf{J}^{3D}_{c_{t-ik}} - \hat{\mathbf{J}}^{3D}_{c_{t-ik}}||_1, \\ \text{and } \mathcal{L}_{\mathcal{J}_{2D}} = \sum_{i=0}^{N-1}||\mathbf{J}^{2D}_{t-ik} - \hat{\mathbf{J}}^{2D}_{t-ik}||_1, \end{gathered} \tag{8}$$

where c denotes the canonical pose, meaning that the global rotation is not involved, and the hat superscript indicates the ground truth. Also, the same losses are applied to the 3D mesh and joints after rotating by the predicted orientation, so we have $\mathcal{L}^r_{\mathcal{M}}$ and $\mathcal{L}^r_{\mathcal{J}_{3D}}$. For clarity, we directly use $\mathcal{L}_{\mathcal{M}} = \mathcal{L}^c_{\mathcal{M}} + \mathcal{L}^r_{\mathcal{M}}$ and $\mathcal{L}_{\mathcal{J}_{3D}} = \mathcal{L}^c_{\mathcal{J}_{3D}} + \mathcal{L}^r_{\mathcal{J}_{3D}}$ to represent the losses of 3D mesh and joint before and after applying the rotation, respectively.

Besides, the normal and edge-length constraints are adopted to penalize the outlier vertices:

$$\begin{aligned} \mathcal{L}^c_{\mathcal{N}} &= \sum_{i=0}^{N-1}\sum_{\mathbf{f}\in\mathbf{M}_{c_{t-i}}}\sum_{\mathbf{e}\in\mathbf{f}}||\langle\mathbf{e},\ \hat{\mathbf{n}}\rangle||_1, \\ \text{and } \mathcal{L}^c_{\mathcal{E}} &= \sum_{i=0}^{N-1}\sum_{\mathbf{f}\in\mathbf{M}_{c_{t-i}}}\sum_{\mathbf{e}\in\mathbf{f}}|||\mathbf{e}| - |\hat{\mathbf{e}}|||_1, \end{aligned} \tag{9}$$

where $\mathbf{f}$ denotes the triangle faces from the predicted hand mesh, $\mathbf{e}$, and $\mathbf{n}$ denote the edges and normal vector of the

Methods	PA-J-PE ↓	PA-J-AUC ↑	PA-V-PE ↓	PA-V-AUC ↑	PA-F@5 ↑	PA-F@15 ↑	J-PE ↓	J-AUC ↑	V-PE ↓	V-AUC ↑	F@5 ↑	F@15 ↑
METRO [33]	7.0	-	-	-	-	-	15.2	-	-	-	-	-
Spurr *et al.* [50]	6.8	86.4	-	-	-	-	17.3	69.8	-	-	-	-
Liu *et al.* [35]	6.6	-	-	-	-	-	15.3	-	-	-	-	-
HandOccNet [45]	5.8	88.4	5.5	89.0	78.0	99.0	14.0	**74.8**	13.1	**76.6**	51.5	**92.4**
MobRecon [8]	6.4	87.3	5.6	88.9	78.5	98.8	14.2	73.7	13.1	76.1	50.8	92.1
Our H2ONet†	5.7	88.9	5.5	89.1	80.1	99.0	14.0	74.6	13.0	76.2	51.3	92.1
Our H2ONet	**5.3**	**89.4**	**5.2**	**89.6**	**80.5**	**99.3**	**13.7**	**74.8**	**12.7**	**76.6**	**52.1**	92.3

Table 1. Results on the DexYCB dataset. -: the results are unavailable. †: our method only uses single-frame input. The best and second-best results are marked in **bold** and underlined for better comparison. Our method achieves the best performance on almost all metrics.

	Methods	J-PE ↓	J-AUC ↑	V-PE ↓	V-AUC ↑	F@5 ↑	F@15 ↑
SF	Pose2Mesh [13]	12.5	-	12.7	-	44.1	90.9
	I2L-MeshNet [39]	11.2	-	13.9	-	40.9	93.2
	ObMan [20]	11.1	-	11.0	77.8	46.0	93.0
	HO3D *et al.* [16]	10.7	78.8	10.6	79.0	50.6	94.2
	METRO [33]	10.4	-	11.1	-	48.4	94.6
	Liu *et al.* [35]	10.2	79.7	9.8	80.4	52.9	95.0
	I2UV-HandNet [7]	9.9	80.4	10.1	79.9	50.0	94.3
	Tse *et al.* [54]	-	-	10.9	-	48.5	94.3
	HandOccNet [45]	9.1	81.9	9.0	81.9	56.1	96.2
	MobRecon* [8]	9.2	-	9.4	-	53.8	95.7
	MobRecon [8]	9.4	81.3	9.5	81.0	53.3	95.5
	Our H2ONet†	9.0	82.0	9.0	81.9	55.4	96.0
MF	Hasson *et al.* [18]	11.4	77.3	11.4	77.3	42.8	93.2
	Hasson *et al.* [19]	-	-	14.7	-	39.0	88.0
	Liu *et al.* ‡ [35]	9.8	-	9.4	81.2	53.0	95.7
	Our H2ONet	**8.5**	**82.9**	**8.6**	**82.8**	**57.0**	**96.6**

Table 2. Results on the HO3D-v2 dataset (after PA). SF and MF denote single-frame input and multi-frame input, respectively. *: the model is trained with complement data. -: the results are unavailable from previous papers. ‡: the model uses multi-frame supervision. †: our method with only a single frame as input. Our method achieves the best performance on all metrics.

Methods	J-PE ↓	J-AUC ↑	V-PE ↓	V-AUC ↑	F@5 ↑	F@15 ↑
Liu et al. [35]	30.0	49.0	28.9	50.3	23.2	68.5
HandOccNet [45]	24.9	53.9	24.2	55.1	26.0	72.9
MobRecon [8]	25.2	53.7	24.4	55.0	26.4	72.0
Our H2ONet*	**23.0**	**56.6**	**22.4**	**57.7**	**26.7**	**73.6**

Table 3. Results on the HO3D-v2 dataset (before PA). *: the model is trained with complement data. Our method achieves the best performance on all metrics.

triangle face, respectively. Note that they are computed only for the hand mesh at the canonical pose.

Moreover, for the finger-level occlusion prediction, the commonly-used soft-max cross-entropy loss is employed:

$$\mathcal{L}_{\mathcal{O}} = -\sum_{i=0}^{N-1}\sum_{d=0}^{D-1} \hat{\omega}_{t-ik}^{d} \log \omega_{t-ik}^{d}, \tag{10}$$

where ω_{t-ik}^{d} means the predicted occlusion probability of d-th finger in the $(t-ik)$-th frame.

For the hand orientation regression, the L2 loss is used:

$$\mathcal{L}_{\mathcal{R}} = \sum_{i=0}^{N-1} ||\mathbf{R}_{t-ik}^{T}\hat{\mathbf{R}}_{t-ik} - \mathbb{1}||_2, \tag{11}$$

where $\mathbb{1}$ is an identity matrix.

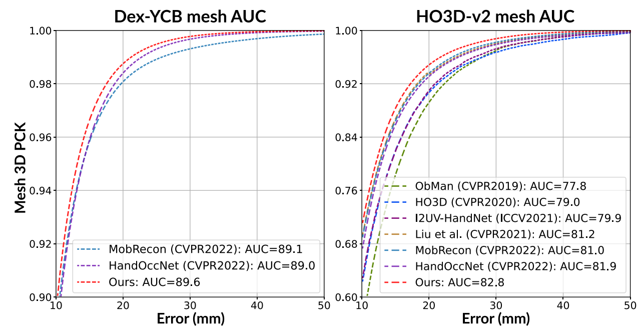

Figure 4. The mesh AUC comparison under different thresholds. Our method performs consistently better than others.

The overall loss is defined by $\mathcal{L}_{total} = \mathcal{L}_{\mathcal{M}} + \mathcal{L}_{\mathcal{J}_{3D}} + \mathcal{L}_{\mathcal{J}_{2D}} + \lambda\mathcal{L}_{\mathcal{N}}^{c} + \mathcal{L}_{\mathcal{E}}^{c} + \mathcal{L}_{\mathcal{O}} + \mathcal{L}_{\mathcal{R}}$ and λ is set to 0.1.

4. Experiments

4.1. Experimental Settings

Datasets. We employ two benchmark datasets in our experiments. The first one is Dex-YCB [5], a recent large-scale 3D hand-object dataset. It provides 1,000 sequences (over 582,000 frames) of 10 subjects grasping 20 different objects from 8 independent views. We use the default "S0" train/test split with 406,888/78,768 samples for training/testing. Evaluation on this large dataset can explore the effectiveness and robustness of different methods. The second one is HO3D-v2 [16], a widely-used 3D hand-object dataset, providing 55/13 sequences of 66,034/11,524 samples for training/test, respectively. As ground truths in its test set are not publicly accessible, evaluation can only be done by submitting results to the official server.

Evaluation metrics. We adopt the evaluation metrics from the HO3D-v2 official online competition. J-PE/V-PE denotes the joint/vertex position error, measuring the average Euclidean distance in millimeters between the predicted and ground-truth 3D hand joint/vertex coordinates. J-AUC/V-AUC indicates the area under the curve of the percentage of correct keypoints (PCK) in different error thresholds for joint/vertex. F-scores measure the harmonic mean of the recall and precision between the predicted and ground-truth hand mesh vertices; we adopt F@5mm and F@15mm, following the previous works. Note that PA de-

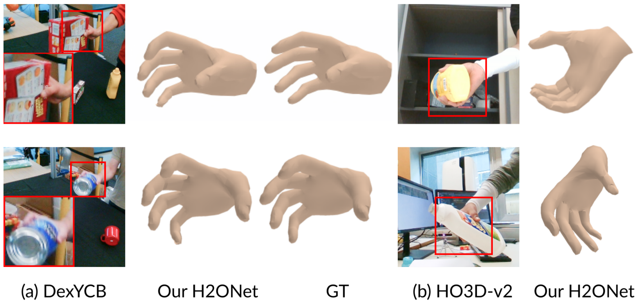

Figure 5. Example challenging cases in the two datasets, in which the occlusions dominate the images. H2ONet can still estimate the hand orientations and reconstruct plausible hand meshes.

notes metrics computed after Procrustes Alignment, meaning that the global rotation and scale differences are ignored.

Implementation details. The feature encoder network is pre-trained on ImageNet [47]. Adam optimizer [30] is applied to train the network with a batch size of 32 on an NVidia RTX 3090. To stabilize the training, we adopt a two-stage strategy. The hand mesh reconstruction branch is first optimized before jointly training with other parts. Input images are resized to 128×128 and augmented by random scaling, rotating, and color jittering. For the multi-frame selection, we set the frame gap k to 5 for Dex-YCB and 30 for HO3D-v2, due to their different FPS. All other details will be available in our code.

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

Evaluation on Dex-YCB. To evaluate the hand mesh reconstruction quality, we first compare our method with the state-of-the-art methods quantitatively on the Dex-YCB test set. We report the performance before and after the PA to better show H2ONet's effectiveness; see Table 1 and the top plot of Fig. 4. Note that Dex-YCB is a very recent dataset, so most previous works have not evaluated on it.

For a detailed comparison, we create a single-frame version of our method, in which we remove the multi-frame information fusions in both the mesh reconstruction and rotation estimation branches. From the results, we can see that the single-frame version of our H2ONet already attains a comparable performance with the state-of-the-art method HandOccNet, showing the effectiveness of our idea of decoupling the mesh reconstruction and global rotation estimation. Furthermore, the full multi-frame version of H2ONet obtains the best results on almost all metrics, manifesting its robustness against occlusion, while delivering real-time speed; see Sec. 4.3 for the details. Some qualitative results are shown in Fig. 5(a) and Fig. 6(a). More are shown in the supp. material. The state-of-the-art methods MobRecon and HandOccNet may fail to estimate the global rotation or recover accurate shapes due to severe occlusions, while our H2ONet can still produce satisfying results, revealing the effectiveness of our occlusion-aware designs.

Methods	Pose2Mesh [13]	I2L-MeshNet [39]	ObMan [20]	Liu et al. [35]	HandOccNet [45]	MobRecon [8]	Ours†	Ours
FPS	22	33	20	32	30	**59**	<u>43</u>	35

Table 4. Running time comparison with other methods. Ours†: only a single frame as input. Ours: multi-frame input while being real-time and achieving top performance (see Tabs. 1 to 3)

	Models	PA-J-PE ↓	PA-V-PE ↓	J-PE ↓	V-PE ↓
i)	B	6.36	5.59	14.20	13.11
ii)	B+D	5.65	5.45	14.02	13.03
iii)	B+D+MF	5.57	5.41	14.01	13.02
iv)	B+D+MF+FS	8.54	8.33	16.96	16.38
v)	B+D+MF+FO	5.31	5.21	13.96	12.98
vi)	B+D+MF+FO+HO	**5.30**	**5.19**	**13.68**	**12.70**

Table 5. Ablation study on major components.

Evaluation on HO3D-v2. To further evaluate the generalizability, we conduct the same experiments on the HO3D-v2 dataset. As the ground truths of its test set are not publicly available, we obtain the results of the competitors from the previous papers or the official evaluation server. The experimental results after and before PA are shown in Tables 2 and 3, respectively. We also provide the mesh AUC comparison under different thresholds; see the bottom plot of Fig 4. Beyond the existing works, our method explicitly considers finger-level occlusions and shows better performance. During the experiments, we observe that our hand orientation regression module often overfits the training set of HO3D-v2, due to its limited 3D rotation distribution. So, we pre-train the model on Dex-YCB for a few epochs to alleviate this issue; please see the supp. material for the rotation distribution comparison between the two datasets.

Also, from the results, it is clear to see that our method achieves better precision on all metrics for all different thresholds consistently, which demonstrates its effectiveness and robustness. To have an intuitive comparison, we show visual results for inputs with serious occlusions in Fig. 5(b) and the comparison with other methods in Fig. 6(b). Yet, our method can still produce plausible shapes while estimating more accurate global orientations.

4.3. Efficiency

Table 4 reports the inference times. All methods are tested on an NVidia RTX 2080Ti. Specifically, we set the batch size to one, exclude the data-loading time, and average the computing time over the entire HO3D-v2 test set (11,524 frames). Though our single-frame version is slightly slower than MobRecon, we achieve better performance. Particularly, our full model consistently attains the best performance, while being real-time, even if it has to process three times more frames.

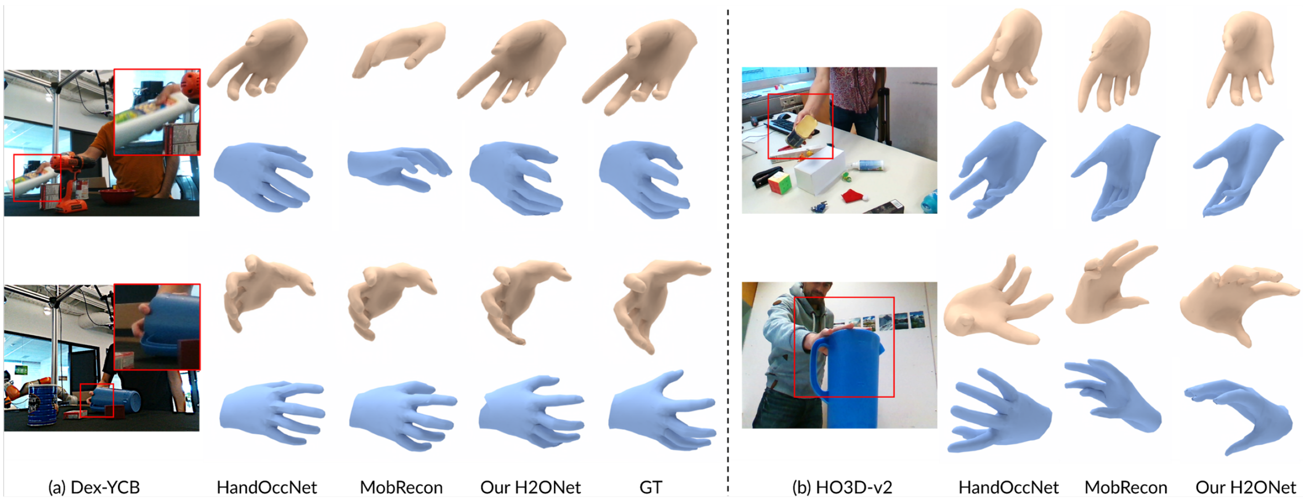

Figure 6. Qualitative comparison of our method and state-of-the-art 3D hand mesh reconstruction methods [8, 45] on different datasets. The first and second rows in each example denote the normal view and another view, respectively, for better comparison.

4.4. Ablation Studies

We perform ablation studies on Dex-YCB to study the effectiveness of H2ONet and its major components. As Table 5 shows, we denote Baseline (**B**) as our model after removing the following major components: Dual branch structure (**D**), Multiple Frame inputs (**MF**), Finger-level Occlusion-aware feature fusion (**FO**) in the hand reconstruction branch, and Hand-level Occlusion-aware feature fusion (**HO**) in the hand orientation regression branch. Besides, **FS** denotes that we directly select the output of the most non-occluded frame as the result of the current frame according to the ground-truth occlusion labels.

Dual-branch encoder. The dual-branch encoder structure plays an important role in our framework. Comparing the first two rows in Table 5, we can see that decoupling the hand mesh reconstruction and the global orientation regression can boost the performance with a relatively large gap, especially for the metrics after PA, demonstrating the effectiveness of our dual-branch design.

Finger-level occlusion-aware feature fusion. We first compare the performance with and without multi-frame inputs. Row iii) denotes the model that fuses multi-frame features by direct concatenation. Comparing Rows iii) and ii), directly fusing the features only brings a tiny improvement, implying that it is ineffective to let the network learn which parts of the features are useful without any guidance. Besides, comparing Rows iv) and iii), directly using the output of the most non-occluded frame results in worse performance since current non-occluded fingers may be occluded in the most non-occluded frame. However, comparing Rows v) and iii), adopting the finger-level occlusion predictions improves the performance, revealing the efficiency of adaptively combining the non-occluded information in multiple frames.

Hand-level occlusion-aware feature fusion. Comparing Rows vi) and v) shows an additional larger improvement brought by introducing our frame-by-frame enquiry feature fusion strategy, which demonstrates the effectiveness of our hand-level occlusion-aware fusion module. Note that this module has little influence on the metrics after PA, since it does not modify the hand shape.

5. Conclusion

We presented H2ONet, a new 3D hand mesh reconstruction method that effectively utilizes non-occluding information over fingers and multiple frames to address the occlusion issue. To better fuse multi-frame features, we decouple the pipeline into two branches to reconstruct the hand mesh at the canonical pose and regress the hand orientation. Besides, we design finger-level and hand-level occlusion-aware feature fusions to better exploit information from non-occluded regions across multi-frames. Experimental results confirm the state-of-the-art performance of H2ONet on two hand-object benchmarks.

Limitations. First, the rigid body assumption of the hand may limit the application scenarios in the real world. Second, it requires diverse rotation distribution of the dataset to train the hand orientation regression branch. Third, the mesh vertices offset may cause artifacts in some cases.

Acknowledgments. This work was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Reference Numbers: T45-401/22-N) and project MMT-p2-21 of the Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong. Hao Xu thanks for the care and support from Yutong Zhang and his family.

References

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering. In *CVPR*, pages 1067–1076, 2019. 2

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-supervised domain adaptation via GAN and mesh model for estimating 3D hand poses interacting objects. In *CVPR*, pages 6121–6131, 2020. 2

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip H.S. Torr. 3D hand shape and pose from images in the wild. In *CVPR*, pages 10843–10852, 2019. 2

[4] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *ICCV*, pages 12417–12426, 2021. 2

[5] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, pages 9044–9053, 2021. 6

[6] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. Temporal-aware self-supervised learning for 3D hand pose and mesh estimation in videos. In *WACV*, pages 1050–1059, 2021. 2

[7] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2UV-HandNet: Image-to-UV prediction network for accurate and high-fidelity 3D hand mesh modeling. In *ICCV*, pages 12929–12938, 2021. 2, 6

[8] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. MobRecon: Mobile-friendly hand mesh reconstruction from monocular image. In *CVPR*, pages 20544–20554, 2022. 1, 3, 4, 6, 7, 8

[9] Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2D-1D registration. In *CVPR*, pages 13274–13283, 2021. 2

[10] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3D hand reconstruction via self-supervised learning. In *CVPR*, pages 10451–10460, 2021. 2

[11] Yu Cheng, Bo Yang, Bo Wang, and Robby T. Tan. 3D human pose estimation using spatio-temporal networks with explicit occlusion training. In *AAAI*, pages 10631–10638, 2020. 3

[12] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T. Tan. Occlusion-aware networks for 3D human pose estimation in video. In *ICCV*, pages 723–732, 2019. 3

[13] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose. In *ECCV*, pages 769–787, 2020. 1, 6, 7

[14] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *CVPR*, pages 1831–1840, 2017. 1, 3

[15] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, pages 10833–10842, 2019. 1, 2

[16] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and object poses. In *CVPR*, pages 3196–3206, 2020. 6

[17] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM TOG*, 39(4):87–1, 2020. 1

[18] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, pages 571–580, 2020. 1, 2, 6

[19] Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from RGB videos. In *3DV*, pages 659–668, 2021. 2, 6

[20] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, pages 11807–11816, 2019. 2, 6, 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[22] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[23] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-Excitation networks. In *CVPR*, pages 7132–7141, 2018. 3

[24] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, pages 118–134, 2018. 2

[25] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *ICCV*, pages 11107–11116, 2021. 2

[26] Hanbyul Joo, Tomas Simon, Mina Cikara, and Yaser Sheikh. Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction. In *CVPR*, pages 10873–10883, 2019. 1

[27] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *3DV*, pages 333–344, 2020. 2

[28] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *ECCV*, pages 713–728, 2018. 2

[29] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *CVPR*, pages 2540–2548, 2015. 2

[30] P. Diederik Kingma and Lei Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

[31] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, pages 4990–5000, 2020. 1, 2

[32] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3D meshes. In *ECCVW*, 2018. 4

[33] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, pages 1954–1963, 2021. 2, 6

[34] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, pages 12939–12948, 2021. 2

[35] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3D hand-object poses estimation with interactions in time. In *CVPR*, pages 14687–14697, 2021. 1, 2, 6, 7

[36] Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. HandVoxNet: Deep voxel-based network for 3D hand shape and pose estimation from a single depth map. In *CVPR*, pages 7113–7122, 2020. 2

[37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. 2

[38] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. NeuralAnnot: Neural annotator for 3D human mesh training sets. In *CVPRW*, 2022. 1

[39] Gyeongsik Moon and Kyoung Mu Lee. I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In *ECCV*, pages 752–768, 2020. 1, 2, 6, 7

[40] Gyeongsik Moon and Kyoung Mu Lee. Pose2Pose: 3D positional pose-guided 3D rotational pose prediction for expressive 3D human pose and mesh estimation. *arXiv preprint arXiv:2011.11534*, 2020. 1

[41] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. DeepHandMesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. In *ECCV*, pages 440–455, 2020. 1

[42] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *ECCV*, pages 548–564, 2020. 1, 2

[43] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM TOG*, 38(4):1–13, 2019. 2

[44] Evonne Ng, Shiry Ginosar, Trevor Darrell, and Hanbyul Joo. Body2Hands: Learning to infer 3D hands from conversational gesture body dynamics. In *CVPR*, pages 11865–11874, 2021. 1

[45] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. HandOccNet: Occlusion-robust 3D hand mesh estimation network. In *CVPR*, pages 1496–1505, 2022. 1, 2, 6, 7, 8

[46] Javier Romero, Dimitris Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG*, 36(6):1–17, 2017. 2

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet: large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 7

[48] István Sárándi, Timm Linder, Kai O. Arras, and Bastian Leibe. How robust is 3D human pose estimation to occlusion? *arXiv preprint arXiv:1808.09316*, 2018. 2

[49] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, volume 28, 2015. 2

[50] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints. In *ECCV*, pages 211–228, 2020. 6

[51] David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *CVPR*, pages 5610–5619, 2016. 2

[52] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3D hand-mesh reconstruction. In *ICCV*, pages 11698–11707, 2021. 1

[53] Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *CVPR*, pages 644–651, 2014. 2

[54] Tze Ho Elden Tse, Kwang In Kim, Ales Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *CVPR*, pages 1664–1674, 2022. 2, 6

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017. 3

[56] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dual Grid Net: Hand mesh vertex regression from single depth maps. In *ECCV*, pages 442–459, 2020. 2

[57] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. Seqhand: RGB-sequence-based 3D hand pose and shape estimation. In *ECCV*, pages 122–139, 2020. 1, 2

[58] Linlin Yang, Shicheng Chen, and Angela Yao. SemiHand: Semi-supervised hand pose estimation with consistency. In *ICCV*, pages 11364–11373, 2021. 2

[59] Lixin Yang, Jiasen Li, Wenqiang Xu, Yiqun Diao, and Cewu Lu. BiHand: Recovering hand mesh with multi-stage bisected hourglass networks. In *BMVC*, 2020. 2

[60] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a contact potential field to model the hand-object interaction. In *ICCV*, pages 11097–11106, 2021. 2

[61] Baowen Zhang, Yangang Wang, Xiaoming Deng, Yinda Zhang, Ping Tan, Cuixia Ma, and Hongan Wang. Interacting two-hand 3D pose and shape reconstruction from single color image. In *ICCV*, pages 11354–11363, 2021. 2

[62] Xiong Zhang, Hongsheng Huang, Jianchao Tan, Hongmin Xu, Cheng Yang, Guozhu Peng, Lei Wang, and Ji Liu. Hand image understanding via deep multi-task learning. In *ICCV*, pages 11281–11292, 2021. 2

[63] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular RGB image. In *ICCV*, pages 2354–2364, 2019. 2

[64] Zimeng Zhao, Xi Zhao, and Yangang Wang. TravelNet: Self-supervised physically plausible hand motion learning from monocular color images. In *ICCV*, pages 11666–11676, 2021. 2

[65] Lu Zhou, Yingying Chen, Yunze Gao, Jinqiao Wang, and Hanqing Lu. Occlusion-aware siamese network for human pose estimation. In *ECCV*, pages 396–412, 2020. 1, 3

[66] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, pages 5745–5753, 2019. 5

[67] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR*, pages 5346–5355, 2020. 2

[68] Meilu Zhu, Daming Shi, Mingjie Zheng, and Muhammad Sadiq. Robust facial landmark detection via occlusion-adaptive deep networks. In *CVPR*, pages 3486–3496, 2019. 1, 3

[69] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In *ICCV*, pages 813–822, 2019. 2