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

Hao Xu\textsuperscript{1,2} Tianyu Wang\textsuperscript{1} Xiao Tang\textsuperscript{1} Chi-Wing Fu\textsuperscript{1,2,3}
\textsuperscript{1}Department of Computer Science and Engineering \textsuperscript{2}Institute of Medical Intelligence and XR \textsuperscript{3}Shun Hing Institute of Advanced Engineering
The Chinese University of Hong Kong
\{xuhao,wangty,cwf\}@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.

1. Introduction

Estimating 3D hand meshes from RGB images is a fundamental task useful for many applications, e.g., augmented reality \cite{17, 52}, behavior understanding \cite{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 \cite{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 \cite{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 \cite{8, 52} adopt lightweight frameworks for real-time inference, the influence of occlusion is ignored.

On the other hand, some recent works \cite{18, 57} start to explore multi-frame RGB images as input for 3D hand mesh reconstruction. SeqHAND \cite{57} integrates LSTM as a feature extractor to memorize the hand motion over consecutive frames. Liu et al. \cite{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 Hand-Occlusion-and-Orientation-aware Network, 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.

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.
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 \( I = \{I_{t-k} : i \in \{0, 1, 2\}\} \), where \( I_t \) denotes the current frame and \( I_{t-k}, I_{t-2k} \) denote two previous frames (\( k \) is frame gap), an hourglass block is first adopted to extract multi-scale refined feature \( F_t \). 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 \( F_t \) into (i) the hand mesh encoder to produce feature \( F_M \) for reconstructing the 3D hand mesh; and (ii) the hand orientation encoder to produce feature \( F_O \) 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
number of vertices in the down-sampled hand mesh. The number of joints;
2D joint positions $J$

SpiralConv-based [32] decoder to upsample hier-
archically by a factor of 2 for each layer and finally regress
the original-resolution mesh vertices. Since the feature be-
comes almost 0 for the last decoder layer should contain the least orienta-
tion of each finger, we should examine the features of
Finger-level occlusions prediction. To predict the oc-
cclusion of each finger, we should examine the features of
the pixels associated with each finger and filter out irre-
levance of each finger, we should examine the features of

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

$$ F_V = P \cdot F_J. \quad (1) $$

where $F_V \in \mathbb{R}^{n \times m}$, $P \in \mathbb{R}^{n \times m}$, $F_J \in \mathbb{R}^{n \times m}$; $m = 21$ 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 $F_V$ is fed into a
four-layer SpiralConv-based [32] decoder to upsample hier-
archically by a factor of 2 for each layer and finally regress
the original-resolution mesh vertices. Since the feature be-
fore the last decoder layer should contain the least orienta-
tion information and already has the same number of ver-
tices as the output hand mesh, we adopt it as the input to
fuse the information from all frames. For clarity, we use
$F_{V_{t-ik}}$, $i \in \{0, 1, 2\}$ 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 $F_{V_{t-ik}}$ to differ-
ent fingers based on the predefined vertex indices. Since
non-occluded fingers are more informative, vertex features
belonging to each finger

\begin{align}
W_f^{t-ik} &= \frac{\sum_{j=0}^{N-1} \omega_{t-ik}^{j}}{\sum_{j=0}^{N-1} \omega_{t-ik}^{j}}, \quad \Omega_f^{t-ik} = \{\omega_{t-ik}^{d} : d \in \{0, ..., D-1\}\},
\end{align}

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

$$ F' = \sum_{i=0}^{N-1} W_{t-ik} \cdot F_{V_{t-ik}}. \quad (3) $$

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

$$ M'_{t-ik} = p(F_{V_{t-ik}}), \quad i \in \{0, 1, 2\}, $$

$$ \Delta M = q(F'), \quad M'_{t} = \Delta M + M'_{t-ik}, \quad (4) $$

where $p(\cdot)$ denotes the SpiralConv-based decoder and
$M'_{t-ik}$ denotes the predicted mesh vertices at the canonical
pose from the input frame $I_{t-ik}$.

3.3. Hand Orientation Regression

Considering that the current hand can be severely oc-
ccluded at times, although the object pose can provide cer-
tain cues, it is not always reliable. Therefore, we propose to
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 $F_O$ to obtain global features $G$ 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 $\Omega^h$ in two different ways. First, for the $i$th frame, we design a frame-by-frame enquiry strategy to violate the ill-posed issue when the hand is highly (or fully) occluded. We design a frame-by-frame enquiry strategy to violate the ill-posed issue when the hand is highly (or fully) occluded. Formally, we split the computation of $G_{t-ik}$ into two steps. First, for the $i$th frame, we encode and flatten the extracted feature $F_O$ to obtain global features $G$ for all frames. Then, we adopt the hand-level occlusions to exploit information from other frames.

\[ \Omega^h_{t-ik} = \prod_{d=0}^{D-1} \arg\max[\omega^d_{t-ik}], \ i \in \{0, 1, 2\}, \]  

where $\Omega^h_{t-ik}$ is the predicted hand-level occlusion of the $(t-ik)$-th frame and $\omega^d_{t-ik}$ 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 $\Omega^h$.

In detail, we check $\Omega^h$ frame by frame in a reverse chronological order (from $I_1$ to $I_{2-ik}$), and fetch the feature $G_{t-ik}$ as the output feature $G'$ if the $(t-ik)$-th frame is non-occluded. Formally, we split the computation of $G'$ into two steps. First, for the $(t-ik)$-th frame, we compute

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

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

To further obtain the hand orientation, we feed the original features $G$ 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 $R$. 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 $G'$ and the feature of the current frame $G_t$ into another MLP $g(\cdot)$ to regress the rotation offset $\Delta R$ and then update $R_t$

\[
\begin{align*}
R_{t-ik} &= f(G_{t-ik}), \ i = 0, ..., N-1. \\
\Delta R &= g\left(\text{cat}\left[\text{G}'_{t-ik}, G_t\right]\right), \ R_t = \Delta R R_t,
\end{align*}
\]

where $\text{cat}[\ldots]$ denotes the concatenate operation.

Finally, we apply $R_{t-ik}$ on $M_{t-ik}$ to obtain the rotated hand mesh $M'_{t-ik}$ in the camera coordinate system. Also, we calculate the 3D joint coordinates $J_{\text{c}_{t-ik}}$ at the canonical pose and $J'_{\text{c}_{t-ik}}$ in the camera coordinate system by multiplying $M'_{t-ik}$ and $M_{t-ik}$, 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 superposition 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 $L_M^c$, 3D joint loss $L_{J^c}$, and 2D joint loss $L_{J^2}$ as

\[
\begin{align*}
L_M^c &= \sum_{i=0}^{N-1} ||M'_{t-ik} - \hat{M}_{t-ik}||_1, \\
L_{J^c} &= \sum_{i=0}^{N-1} ||J_{\text{c}_{t-ik}} - \hat{J}_{\text{c}_{t-ik}}||_1, \\
L_{J^2} &= \sum_{i=0}^{N-1} ||J_{t}^{2D} - \hat{J}_{t}^{2D}||_1,
\end{align*}
\]

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 $L_M^r$, $L_{J^r}$, $L_{J^r}$, and $L_{J^r}$ 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:

\[
\begin{align*}
L_{N}^c &= \sum_{i=0}^{N-1} \sum_{f \in M_{t-ik}} ||(e, \ n)||_1, \\
L_{E}^c &= \sum_{i=0}^{N-1} \sum_{e \in f} ||e| - |\hat{e}||_1,
\end{align*}
\]

where $f$ denotes the triangle faces from the predicted hand mesh, $e$, and $n$ denote the edges and normal vector of the
Our method achieves the best performance on almost all metrics.

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

\[
\mathcal{L}_{\theta} = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{d=0}^{D-1} \omega_{i-k}^d \log \omega_{i-k}^d,
\]

where \(\omega_{i-k}^d\) means the predicted occlusion probability of \(d\)-th finger in the \((t-i)\)-th frame.

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

\[
\mathcal{L}_{\theta} = \frac{1}{N} \sum_{i=0}^{N-1} \|R_{i-k}^T \hat{R}_{i-k} - 1\|_2,
\]

where \(I\) is an identity matrix.

Table 1. Results on the DexYCB dataset. - : the results are unavailable. \(\dagger\): our method only uses single-frame input. The best and second-best results are marked in \textbf{bold} and \underline{underlined} for better comparison. Our method achieves the best performance on almost all metrics.

<table>
<thead>
<tr>
<th>Methods</th>
<th>PA-J-PE</th>
<th>PA-J-AUC</th>
<th>PA-V-PE</th>
<th>PA-V-AUC</th>
<th>PA-F@5</th>
<th>PA-F@15</th>
<th>J-PH</th>
<th>J-AUC</th>
<th>V-PH</th>
<th>V-AUC</th>
<th>F@5</th>
<th>F@15</th>
</tr>
</thead>
<tbody>
<tr>
<td>METRO [33]</td>
<td>7.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>15.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sparu et al. [50]</td>
<td>6.8</td>
<td>86.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.3</td>
<td>69.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Liu et al. [35]</td>
<td>6.6</td>
<td>-</td>
<td>87.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>15.3</td>
<td>68.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HandOccNet [15]</td>
<td>5.8</td>
<td>88.4</td>
<td>5.5</td>
<td>89.0</td>
<td>78.0</td>
<td>99.0</td>
<td>14.0</td>
<td>74.8</td>
<td>13.1</td>
<td>76.6</td>
<td>51.5</td>
<td>92.4</td>
</tr>
<tr>
<td>MobRecon [8]</td>
<td>6.4</td>
<td>87.3</td>
<td>5.6</td>
<td>88.9</td>
<td>78.5</td>
<td>98.8</td>
<td>14.2</td>
<td>73.7</td>
<td>13.1</td>
<td>76.1</td>
<td>50.8</td>
<td>92.1</td>
</tr>
<tr>
<td>Our H2ONet (\dagger)</td>
<td>5.7</td>
<td>88.7</td>
<td>5.5</td>
<td>89.1</td>
<td>80.1</td>
<td>99.0</td>
<td>14.0</td>
<td>74.6</td>
<td>13.0</td>
<td>76.2</td>
<td>51.3</td>
<td>92.1</td>
</tr>
<tr>
<td>Our H2ONet</td>
<td>5.3</td>
<td>89.4</td>
<td>5.2</td>
<td>89.6</td>
<td>80.5</td>
<td>99.3</td>
<td>13.7</td>
<td>74.8</td>
<td>12.7</td>
<td>76.6</td>
<td>52.1</td>
<td>92.3</td>
</tr>
</tbody>
</table>

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. \(\dagger\): our method with only a single frame as input. Our method achieves the best performance on all metrics.

<table>
<thead>
<tr>
<th>Methods</th>
<th>PA-J-PE</th>
<th>PA-J-AUC</th>
<th>PA-V-PE</th>
<th>PA-V-AUC</th>
<th>PA-F@5</th>
<th>PA-F@15</th>
<th>J-PH</th>
<th>J-AUC</th>
<th>V-PH</th>
<th>V-AUC</th>
<th>F@5</th>
<th>F@15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pose2Mesh [13]</td>
<td>12.5</td>
<td>-</td>
<td>12.7</td>
<td>-</td>
<td>44.1</td>
<td>90.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>I2L-MobNet [39]</td>
<td>11.2</td>
<td>-</td>
<td>13.9</td>
<td>-</td>
<td>40.9</td>
<td>93.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OhMan [20]</td>
<td>11.1</td>
<td>-</td>
<td>11.0</td>
<td>77.8</td>
<td>46.0</td>
<td>93.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HO3D et al. [16]</td>
<td>10.7</td>
<td>78.8</td>
<td>10.6</td>
<td>79.0</td>
<td>50.6</td>
<td>94.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>METRO [33]</td>
<td>10.4</td>
<td>-</td>
<td>11.1</td>
<td>-</td>
<td>48.4</td>
<td>94.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Liu et al. [35]</td>
<td>10.2</td>
<td>79.7</td>
<td>9.8</td>
<td>80.4</td>
<td>52.9</td>
<td>95.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>I2U-HandNet [7]</td>
<td>9.9</td>
<td>80.4</td>
<td>10.1</td>
<td>79.9</td>
<td>50.0</td>
<td>94.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tse et al. [35]</td>
<td>9.1</td>
<td>81.9</td>
<td>9.0</td>
<td>81.9</td>
<td>56.1</td>
<td>96.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HandOccNet [45]</td>
<td>9.2</td>
<td>-</td>
<td>9.4</td>
<td>-</td>
<td>53.8</td>
<td>95.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MobRecon (\ast) [8]</td>
<td>9.4</td>
<td>81.3</td>
<td>9.5</td>
<td>81.0</td>
<td>53.3</td>
<td>95.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Our H2ONet(\dagger)</td>
<td>9.0</td>
<td>82.0</td>
<td>9.0</td>
<td>81.9</td>
<td>55.4</td>
<td>96.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Our H2ONet</td>
<td>8.5</td>
<td>82.9</td>
<td>8.6</td>
<td>82.8</td>
<td>57.0</td>
<td>96.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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.

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

The overall loss is defined by:

\[
\mathcal{L}_{\text{total}} = \mathcal{L}_{\mathcal{M}} + \mathcal{L}_{\mathcal{J}^{2D}} + \lambda \mathcal{L}_{\mathcal{V}^{X}} + \mathcal{L}_{\mathcal{E}^P} + \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{R}}
\]

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-
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 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 $128 \times 128$ 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.

**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.
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 fusion 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


[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


[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


