Interacting Attention Graph for Single Image Two-Hand Reconstruction

Mengcheng Li[1], Liang An[1], Hongwen Zhang[1], Lianpeng Wu[2], Feng Chen[1], Tao Yu[1], Yebin Liu[1]
[1]Tsinghua University [2]Hisense Inc.

Abstract

*Graph convolutional network (GCN) has achieved great success in single hand reconstruction task, while interacting two-hand reconstruction by GCN remains unexplored. In this paper, we present **Int**eracting **A**ttention **G**raph **Hand** (IntagHand), the first graph convolution based network that reconstructs two interacting hands from a single RGB image. To solve occlusion and interaction challenges of two-hand reconstruction, we introduce two novel attention based modules in each upsampling step of the original GCN. The first module is the pyramid image feature attention (PIFA) module, which utilizes multiresolution features to implicitly obtain vertex-to-image alignment. The second module is the cross hand attention (CHA) module that encodes the coherence of interacting hands by building dense cross-attention between two hand vertices. As a result, our model outperforms all existing two-hand reconstruction methods by a large margin on InterHand2.6M benchmark. Moreover, ablation studies verify the effectiveness of both PIFA and CHA modules for improving the reconstruction accuracy. Results on in-the-wild images further demonstrate the generalization ability of our network. Our code is available at https://github.com/Dw1010/IntagHand.*

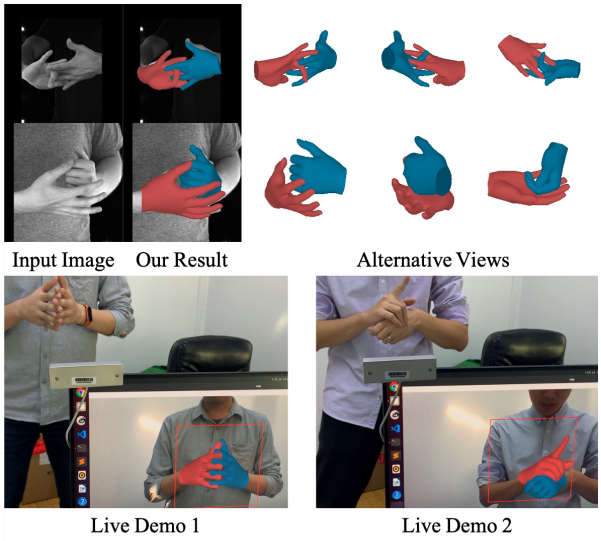

Figure 1. Illustration of our IntagHand for two-hand reconstruction. The top two rows are results on InterHand2.6M [25] dataset and the bottom two rows are results on in-the-wild images. Our method produces high quality two-hand mesh reconstruction of flexible hand poses under severe occlusions.

1. Introduction

Interacting two-hand reconstruction is one of the fundamental tasks towards manifold industrial applications such as virtual reality (VR), human-computer-interaction (HCI), robotics, holoportation, digital medicine, *etc*. Recently, monocular single hand pose and shape recovery has witnessed great success owing to deep neural networks [3, 12, 21, 49, 52] and large scale datasets [13, 16, 25, 26, 53, 54]. However, two-hand reconstruction is more challenging and remains unsolved for two reasons. First, severe mutual occlusions and appearance similarity confuse the feature extractors, making it difficult for networks to align hand poses with image features. Second, the interaction context between two hands is difficult to effectively formulate during network design and training.

Monocular depth-based two-hand tracking [22, 27, 28, 38–40] has been studied for years and promising results have been demonstrated. However, the energy demand and algorithm complexity restrict the ubiquitous application of depth-based methods. Recently, Wang *et al*. [40] contributes a monocular RGB based two-hand reconstruction by tracking dense matching map. However, the tracking procedure itself is inherently sensitive to fast motion, and does not take full advantage of prior knowledge between interacting hands. Since the proposal of the large scale two-hand dataset InterHand2.6M [25], learning based single image two-hand reconstruction methods have emerged. Existing methods [11, 18, 25, 46] either employ 2.5D heatmaps to estimate hand joint positions [11, 18, 25], or use them as attention maps to extract sparse image features [46]. However, sparse local image features encoded in the heatmaps could not effectively model hand surface occlusions, and could not extract dense interaction context. In contrast, vertex-based graph convolutional network

(GCN) has achieved great success in single hand reconstruction [12, 23, 24, 37], yet it has not been demonstrated in two-hand reconstruction, and the previously mentioned challenges remain to be addressed.

In this paper, we propose **Int**eracting **A**ttention **G**raph **Hand** (IntagHand), a novel GCN based single image two-hand reconstruction method. As a basic pipeline, we initially utilize GCN to regress mesh vertices of each hand in a coarse-to-fine manner, similar to traditional GCN [12]. However, for the two-hand task, naively using a two-stream GCN to generate two hand vertices fails to utilize the interaction context between two hands, making the network confused regarding two-hand mutually occluded parts. Moreover, without any image feature feedback, the network has difficulty aligning vertices to image features as suggested by [24,47]. To address these issues, we equip the GCN with two novel attention modules. The first module is a pyramid image feature attention (**PIFA**) module, which uses a transformer encoder to update the latent vertex features with patched image features. Unlike projection based vertex-image alignment [47], PIFA benefits from the global sensing ability of the attention mechanism to help each vertex seek alignment over all image patches. Furthermore, as GCN upsamples mesh vertices in a coarse–to–fine manner, we design an encoder–decoder based image feature extraction module to extract pyramid features, forcing the high resolution mesh to attend to high–resolution features. The second module is a cross hand attention (**CHA**) module that encodes interaction context into hand vertex features. The CHA module allows vertices of each hand to pay dense attention to the other hand's vertex features in order to disambiguate interhand occlusion. Benefiting from the GCN structure and the novel attention based modules, the IntagHand outperforms existing methods on InterHand2.6M [25] by a large margin (**8.8**mm v.s. 13.5mm). Beyond all existing methods, we demonstrate well-aligned two-hand reconstruction results on various in-the-wild images for the first time. Overall, our contributions are summarized as:

- We propose the first two-hand reconstruction method using GCN based mesh regression, named IntagHand, demonstrating the effectiveness of GCN for the two-hand reconstruction task.
- We propose a pyramid image feature attention (PIFA) module to distill local occlusion information with global image patch attention, producing better alignment between the hand vertices and the image features.
- We propose a cross hand attention (CHA) module to implicitly model the two-hand interaction context, improving the reconstruction accuracy for closely interacting poses.
- Our method achieves the new state-of-the-art performance by a large margin on the InterHand2.6M benchmark. Furthermore, We demonstrate the generalization ability of our method on in-the-wild images.

2. Related Works

2.1. Single Hand Reconstruction

Since the last century, hand pose estimation and gesture recognition have attracted a substantial interest [15, 42, 43]. In the deep learning era, estimating the 3D hand skeleton from a single image has achieved great success [4, 26, 35, 53]. Since the proposal of the popular parametric hand model MANO [30] and various large scale datasets [16, 25, 33, 54], reconstructing both hand pose and shape [1, 5, 12, 21, 23, 24, 37, 48, 49, 52] has become a mainstream approach. Among all of these methods, the most recent transformer–based models [23, 24] yield the best results, demonstrating the ability of the attention mechanism to learn the nonlocal relationship between any two vertices. This excellent performance inspires us to use the attention mechanism to improve mesh-image alignment and model mesh-mesh interaction.

2.2. Two-Hand Reconstruction

Although nearly all single-hand reconstruction methods could extend to two-hand reconstruction tasks, few works demonstrate a result for close interacting hands. Two-hand reconstruction is one of the key challenges for human total motion capture. Previous body and hand simultaneous reconstruction methods [6, 17, 31, 44, 50, 51] all treat each hand in a separate manner and thus cannot handle close hand interaction cases such as finger knots. A recent multiview tracking based method [34] could reconstruct high-quality interactive hand motions, however, its hardware setup is expensive, and the algorithm is time-consuming. Monocular kinematic tracking based two-hand motion estimation methods, regardless of whether a depth sensor [22, 27, 28, 38, 39] or an RGB camera [40] is incorporated, are sensitive to fast motion and possible tracking failure. However, their dense mapping strategy, which queries correspondences between hand vertices and image pixels, inspires us to seek mesh-image alignment using dense features. In contrast, deep learning based methods such as [11, 18, 25, 32, 46] directly reconstruct per-frame two-hand interaction. Unfortunately, all of these methods either employ 2.5D heatmaps to estimate hand joint positions [11, 25], or use them as attention maps to extract sparse image features [46], or reconstruct each hand respectively and fine-tune later [18, 31]. As hands are naturally 3D surfaces, sparse local image features encoded in the heatmaps may not effectively capture hand surface occlusions and hands interaction context. Therefore, the mentioned methods usually fail to obtain two-hand reconstructions well aligned to images.

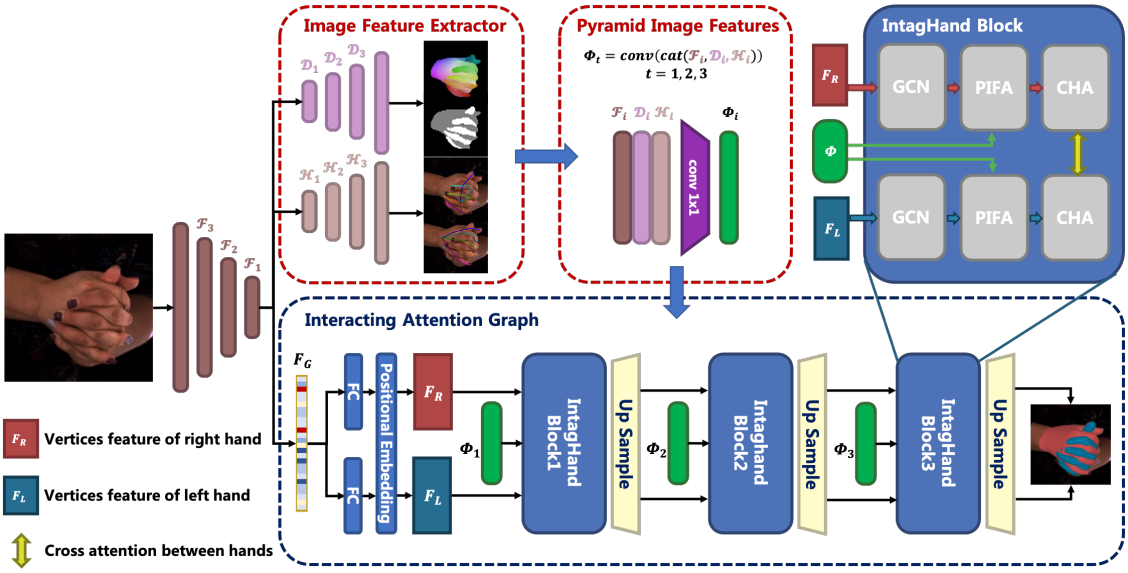

Figure 2. Our network structure. Given an RGB image as input, our network first distills a global feature vector F_G, a sequence of pyramid image features $\{\Phi_t, t = 1, 2, 3\}$ along with other auxiliary predictions (2D pose, segmentation, dense mapping encoding). Then our model directly regresses the 3D coordinates of two hands surface vertices after three steps of IntagHand blocks and upsampling. Each IntagHand block contains a GCN module, a pyramid image feature attention (PIFA) module and a cross-hand attention (CHA) module.

2.3. Convolutional Mesh Regression

Convolutional mesh regression (CMR), which directly regresses mesh vertices in a coarse-to-fine manner from image features using a graph convolutional network (GCN), has been proven successful for generating image-aligned 3D objects [7, 41], faces [29], bodies [20] or hands [12]. A typical CMR pipeline passes the global image feature vector through two or more cascaded graph convolution and upsampling layers and produces per-vertex 3D coordinates of the target object. Compared with joint based or rotation parameter based methods, the CMR method has denser and more semantic model representation and thus has the ability to better align image features in a per-vertex manner. However, existing CMR methods build a single forward pass without explicit image feature feedback strategy, limiting their mesh-image alignment performance as suggested by [47]. Some recent single hand reconstruction works [24, 37] also employ GCN as part of the network structure; however, they discard the coarse-to-fine nature of CMR and simply use a single GCN to enhance local sensing ability.

3. Formulation

3.1. Two-Hand Mesh Representation

Unlike previous two-hand reconstruction methods [18, 25, 32, 46] that use joints or articulated models as hand representations, we only require surface vertices with a fixed mesh topology of two hands. For convenience, we adopt the same mesh topology of the popular MANO [30] model for each hand which contains $N = 778$ vertices. To assist the attention mechanism, we define dense matching encoding for each vertex similarly to [40] as positional embedding. Specifically, we assign different colors for different vertices while maintaining smoothness among adjacent vertices, denoted as $\{c_i \in \mathbb{R}^3, i = 0, 1, ..., N\}$. As shown in Fig. 2, our IntagHand has a hierarchical architecture that reconstructs hand mesh using three coarse-to-fine blocks, with each block followed by the upsampling layer. To construct the coarse-to-fine mesh topology for each block, we leverage the graph coarsening method introduced by [9] and build $N_b = 3$ level submeshes with vertex number $N_0 = 63, N_1 = 126, N_2 = 252$ and reserve the topological relationship between adjacent levels for upsampling. After the third block, a simple linear layer is employed to upsample the final submesh ($N_2 = 252$) to the full MANO mesh ($N = 778$), producing the final two-hand vertices.

3.2. System Overview

Our system contains two main parts: the image encoder–decoder (red dashed wireframes in Fig. 2) and the interacting attention graph (blue dashed wireframe in Fig. 2). Given a single RGB image, we first feed it to an image encoder-decoder structure that yields an intermediate global feature vector F_G and several bundled feature maps $\{\Phi_t \in \mathbb{R}^{C_t \times H_t \times W_t}, t = 0, 1..., N_b - 1\}$, where t indicates that the tth feature level corresponds to the tth IntagHand block, $N_b = 3$ is the block number, $H_t \times W_t$ is the resolution of the feature maps which gradually increases, and

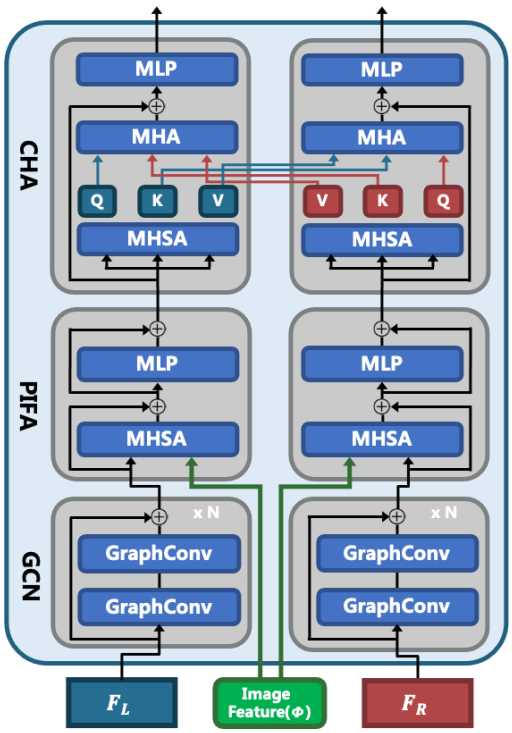

Figure 3. IntagHand Block. Our IntagHand block is formed by three parts: **1**.residual GCN module, **2**.pyramid image feature attention (PIFA) module and **3**.cross-hand attention (CHA) module.

C_t is the feature channel. Afterwards, the IntagHand takes in global feature vector F_G and produces vertices of both the left and right hands. Note that each block of the IntagHand is formed by 3 submodules: a graph convolutional network (GCN) and a pyramid image feature attention module (PIFA) for each hand, together with a cross–hand attention module (CHA) between two hands. These modules are illustrated in Fig. 2 and will be discussed in Sec.4.1, Sec.4.2 and Sec.4.3, respectively.

4. Interacting Attention Graph

4.1. Graph Convolution for Two-Hand Modeling

To directly produce two-hand vertices, our IntagHand is basically built upon previous GCN [12] by extending one hand stream to two-hand streams. However, different from vanilla GCN [12] which transforms the latent vector F_G to a larger unshared per-vertex feature, we utilize fully-connected (FC) layer $g_h(\cdot)$ to map F_G to a more compact feature vector $g_h(F_G)$ which is shared across vertices, and concatenate dense matching encoding (positional embedding) c_i of the ith vertex with the shared vector to form per-vertex feature F_V^i (Fig. 2), which can be denoted as:

$$F_V^i = \text{concat}(g_h(F_G), c_i), \quad i = 0, 1..., N_0; \quad h = L, R, \tag{1}$$

where $F_V^i \in \mathbb{R}^f$ is the initial graph feature, $N_0 = 63$ is the coarsest submesh vertex number, $f = 512$ is the feature length, and h indicates left (L) or right (R) hand. Such operation acts as part of the attention mechanism, and reduces the model size for faster training.

By stacking F_V^i, we obtain $F_V^t \in \mathbb{R}^{N \times f}$, $t = 0$. Afterwards, similar to [12], we perform the Chebyshev spectral graph CNN [7] operation (named as GraphConv for short in Fig. 3) at each $t^{th}(t = 0, 1, 2)$ block to transform input vertex features F_V^t to F_{GCN}^t as

$$F_{GCN}^t = \sigma(\sum_{k=0}^{K-1} T_k^t(\hat{L}^t) F_V^t W_k^t), \tag{2}$$

where $\hat{L}^t$ is the scaled Laplacian matrix, T_k^t is the kth term of the K-order Chebyshev polynomial, W_k^t is the learnable parameter and σ is a nonlinear activation function. F_{GCN}^t denotes the intermediate features that are passed to the PIFA module. Inspired by ResNet [14], we add residual connection for every two GraphConv operations to assist gradient propagation and enhance learning ability; see Fig. 3.

4.2. Pyramid Image Feature Attention Module

Directly reconstructing model mesh from a single global feature F_G without any feedback has difficulty in guaranteeing pixel alignment with the input image [47]. Additionally, a GCN is suggested to pay more attention to local vertex features [24]. To solve these issues, we progressively insert hierarchical image features $\{\Phi_t \in \mathbb{R}^{C_t \times H_t \times W_t}, t = 0, 1, 2\}$ into GCN to guarantee better mesh-image alignment using both local and global context. Note that, each image feature is a combination of both encoder feature and intermediate decoder feature for alignment to richer context (see Fig. 2). Specifically, the output from encoder's last layer is passed through different convolutional layers to predict certain 2D information similar to [37,47]. In our implementation, our model predicts (1) the heatmaps of joints $\mathcal{H}$, (2) the foreground mask of each hand $\mathcal{M}_L$, $\mathcal{M}_R$ and (3) the dense matching encoding of each hand $\mathcal{D}_L$, $\mathcal{D}_R$.

To effectively use image features, we evenly divide the image feature map $\Phi_t \in \mathbb{R}^{C_t \times H_t \times W_t}$ into $N_t \times N_t$ image patches at t^{th} block, and the size of each patch is $\frac{H_t}{N_t} \times \frac{W_t}{N_t}$. Then, the patches are flattened and compacted by a linear layer to yield a sequence of feature vectors $F_I^t \in \mathbb{R}^{(N_t \cdot N_t) \times f}$ with the same feature size f to the vertex features. Afterwards, the image features F_I^t are concatenated with vertex features F_{GCN}^t and fed into a Multi-Head Self-Attention (MHSA) module, producing attention enhanced vertex features F_{PIFA}^t using the following equation

$$F_{PIFA}^t = \text{MHSA}(\text{concat}(F_{GCN}^t, F_I^t)). \tag{3}$$

Although Mesh Graphformer [24] utilizes image feature attention (called 'grid feature') as well, they use the same low–resolution image feature (7×7) in the whole network while our image features are multi-scale ($8\times8 \rightarrow 16\times16 \rightarrow 32\times32$). While low–resolution image features encode more compact (or *global*) information, high–resolution features contain more semantic (or *local*) knowledge as they are closer to the input and output. Therefore, the pyramid structure forces the sparse mesh to attend to the global image features while the dense mesh to the local image features, and could yield better vertex-image alignment.

To demonstrate the function of PIFA, we compute the attention map between the vertex domain and the image domain (please refer to ViT [10] for details). By adding up the PIFA attention maps of all three blocks, we observe that our PIFA module could distinguish between left and right hands on image pixels, and we note that PIFA pays more attention to the area of close interaction. That means PIFA module learns correct vertex-image mapping as we expect.

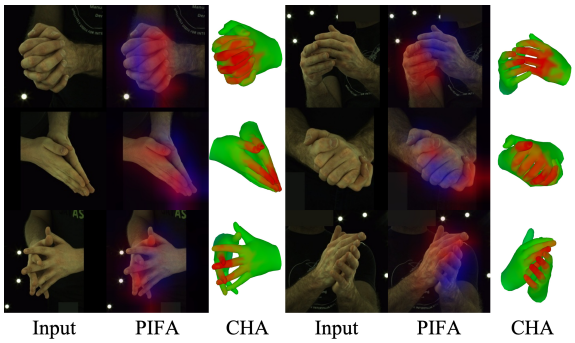

Figure 4. Visualization of attention maps in pseudo color. Six independent examples are shown. In each example, from left to right is input image, PIFA attention map overlaid on image, CHA attention map. For PIFA attention map, the red represents the attention from right hand and the blue represents the left hand. The brighter color means the stronger attention. For CHA attention map, the redder means the stronger cross-hand attention. All attention maps have been normalized for better visualization.

4.3. Cross Hand Attention Module

It has been shown that the poses of two interacting hands are correlated [46]; therefore, it is important to model hands interacting context for two-hand reconstruction. Instead of simply representing interaction as one hand's joints in the other hand's coordinate system [25,46], we use a symmetric cross-hand attention (CHA) module to implicitly formulate this correlation between two hands. For simplicity, we ignore t in F^t_{PIFA} and use F_L and F_R to indicate F_{PIFA} for the left and right hands, respectively.

As shown in Fig. 3, we first perform MHSA on each individual hand to get Q_h, K_h, V_h ($h \in L, R$) indicating the query, key and value feature of each hand. Then we use the query feature Q_h of one hand to fetch the key feature K_h and the value feature V_h of the other hand through Multi-Head Attention (MHA, see Fig. 3) as

$$\begin{aligned} F_{R\rightarrow L} &= \text{softmax}(\frac{Q_L K_R^T}{\sqrt{d}})V_R, \\ F_{L\rightarrow R} &= \text{softmax}(\frac{Q_R K_L^T}{\sqrt{d}})V_L, \end{aligned} \tag{4}$$

where $F_{R\rightarrow L}$ and $F_{L\rightarrow R}$ are the cross-hand attention features encoding the correlation between two hands, and d is a normalization constant. Afterwards, the cross-hand attention features are merged into the hand vertex features by a pointwise MLP layer $f_p(\cdot)$ as

$$\begin{aligned} F_L^{'} &= f_p(F_L + F_{R\rightarrow L}), \\ F_R^{'} &= f_p(F_R + F_{L\rightarrow R}), \end{aligned} \tag{5}$$

where $F_L^{'}$ and $F_R^{'}$ are the output hand vertex features, which act as F_V^{t+1} of both hands for the next $t+1^{th}$ block ($t < N_b$).

It is shown in Fig. 4 that CHA also pays more attention to the closely interacting area, especially the finger-tips. This indicates that the CHA module helps to address mutual collision between hands implicitly.

4.4. Loss Functions

For training the image encoder-decoder, we use smooth L1 loss to supervise the 2D dense matching encoding and mean square error (MSE) loss to supervise 2D heatmaps.

For training IntagHand, we utilize (1) vertex loss, (2) regressed joint loss and (3) mesh smooth loss.

Vertex Loss. We use L1 loss to supervise the 3D coordinates of hand vertices and MSE loss to supervise the 2D projection of vertices:

$$L_V = \sum_{i=1}^{N} \|V_{h,i} - V_{h,i}^{GT}\|_1 + \|\Pi(V_{h,i}) - \Pi(V_{h,i}^{GT})\|_2^2, \tag{6}$$

where $V_{h,i}$ is i^{th} vertex, $h = L, R$ means left or right hand, and Π is the 2D projection operation, the same below. Vertex loss is applied for each submesh, which we ignore here for simplicity.

Regressed Joint Loss. By multiplying the predefined joint regression matrix $\mathcal{J}$, hand joints can be regressed from the predicted hand vertices. We penalize the joint error by the following loss:

$$\begin{aligned} L_J = &\sum_{i=1}^{V} \|\mathcal{J}V_{h,i} - \mathcal{J}V_{h,i}^{GT}\|_1 \\ &+ \sum_{i=1}^{V} \|\Pi(\mathcal{J}V_{h,i}) - \Pi(\mathcal{J}V_{h,i}^{GT})\|_2^2. \end{aligned} \tag{7}$$

Mesh Smooth Loss. To ensure the geometric smoothness of the predicted vertices, two different smooth losses are applied. First, we regularize the normal consistency between the predicted and the ground truth mesh:

$$L_n = \sum_{f=1}^{F} \sum_{e=1}^{3} \| e_{f,i,h} \cdot n_{f,h}^{GT} \|_1, \tag{8}$$

where f is the face index of the hand mesh, $e_{f,i} (i = 1, 2, 3)$ are the three edges of face f and n_f^{GT} is the normal vector of this face calculated from the ground truth mesh. Second, we minimize the L1 distance of each edge length between the predicted mesh and the ground truth mesh:

$$L_e = \sum_{e=1}^{E} \| e_{i,h} - e_{i,h}^{GT} \|_1. \tag{9}$$

Note that, both the image encoder-decoder and the IntagHand are trained simultaneously in an end-to-end manner.

5. Experiments

5.1. Experimental Settings

Implementation Details. Our network is implemented using PyTorch. We use ResNet50 [14] pretrained on ImageNet [8] as backbone to encode the image feature. Following [45], our image decoders utilize three simple deconvolutional layers to predict 2D joint heatmaps, 2D segmentations and dense mapping encodings.

Training Details. We train our model using the Adam optimizer [19] on 4 NVIDIA RTX 2080Ti GPUs with the mini-batch size for each GPU set as 32. The whole training takes 100 epochs across 2.5 days, with the learning rate decaying to 1×10^{-5} at 50th epoch from the initial rate 1×10^{-4}. During training, data augmentations including scaling, rotation, random horizontal flip and color jittering are applied. Note that, we pretrain the last upsampling layer of GCN (see Fig. 2) using posed MANO meshes and fix its weights during further training.

Evaluation Metrics. To evaluate both the pose and shape accuracy of reconstructed hands, we compare the Mean Per Joint Position Error (MPJPE) and Mean Per Vertex Position Error (MPVPE) in millimeters. For fair comparison, we follow Zhang *et al*. [46] to scale the length of the middle metacarpal of each hand to $9.5cm$ during training and rescale it back to the ground truth bone length during evaluation. This is performed after root joint alignment of each hand. We also report the Percentage of Correct Keypoints (PCK) curve and Area Under the Curve (AUC) across linearly spanned thresholds between 0 and 50 millimeters to compare reconstruction accuracy.

5.2. Datasets

InterHand2.6M Dataset. As the only dataset with two-hand mesh annotation, all networks in this paper are trained on InterHand2.6M [25] dataset[1]. Because we only focus on two-hand reconstruction, we pick out the interacting two-hand (IH) data with both human and machine (H+M) annotated, and discard invalid labeling according to the $hand_type_valid$ annotation provided by [25]. Ultimately, 366K training samples and 261K testing samples from InterHand2.6M are utilized. At preprocessing, we crop out the hand region according to the 2D projection of hand vertices and resize it to 256×256 resolution.

RGB2Hands and EgoHands Datasets. RGB2Hands [40] dataset consists of 4 sequences of videos with different types of two-hand interactions, and EgoHands [2] dataset contains 48 egocentric videos capturing complex two people interactions such as playing chess. Both datasets have no mesh annotation, therefore we only use them for qualitative evaluation.

5.3. Qualitative Results

Our qualitative results on InterHand2.6M [25] are shown in Fig. 5 and Fig. 6. As shown in Fig. 5, our method could generate high quality two-hand reconstruction results under severe occlusions and various kinds of interaction context. Compared with previous state-of-the-art method [46], our method produces more realistic finger interactions and less mutual collisions of two hands (see Fig. 6).

Beyond existing methods [11, 25, 46] which only show results in dome setting [25], we further demonstrate the generalization ability of our method on in-the-wild images. As shown in Fig. 7, our method performs well on our real-life data captured by a common USB camera. Besides, without additional training, our model yields excellent results on images from RGB2Hands dataset and EgoHands dataset, showing the potential to be applied in both third/egocentric viewpoint conditions. Moreover, our model runs at 30fps on single NVIDIA RTX 3090 GPU during inference, which enables future real-time applications.

5.4. Quantitative Comparisons

We first compare our IntagHand network with state-of-the-art single-hand reconstruction methods, as shown in Tab. 1. Within single-hand reconstruction methods, each hand is cropped from image by ground truth bounding box and processed separately. It is shown that reconstructing each hand individually works poorly due to heavy occlusion and appearance confusion.

We further compare IntagHand with recent two-hand reconstruction methods. One is Moon *et al*. [25] which re-

[1] We use the v1.0_5fps version of the InterHand2.6M which is CC-BY-NC 4.0 licensed.

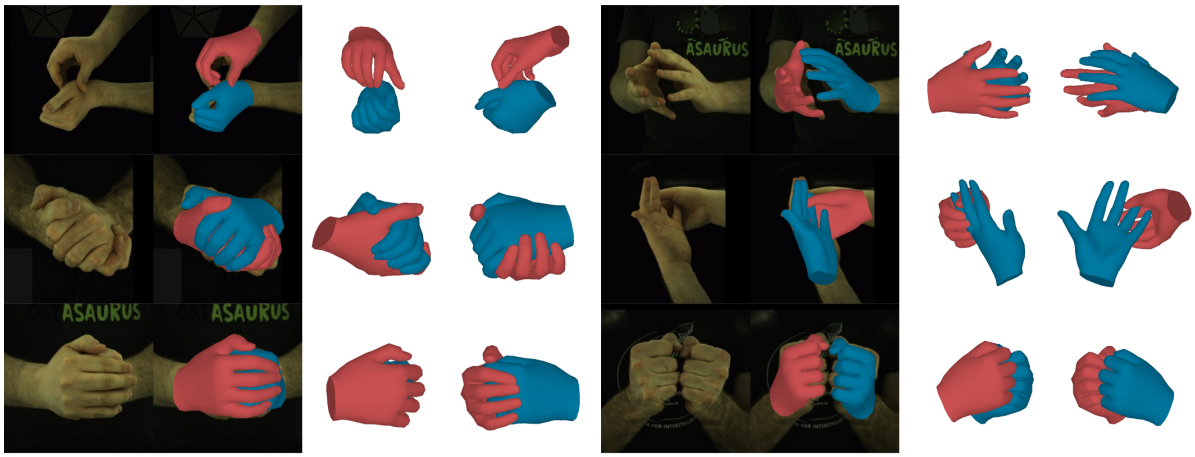

Figure 5. Qualitative results of our method on InterHand2.6M test dataset. Our method works well under various kinds of interactions. Note that, our method could even produce correct finger-level interactions without explicit collision detection.

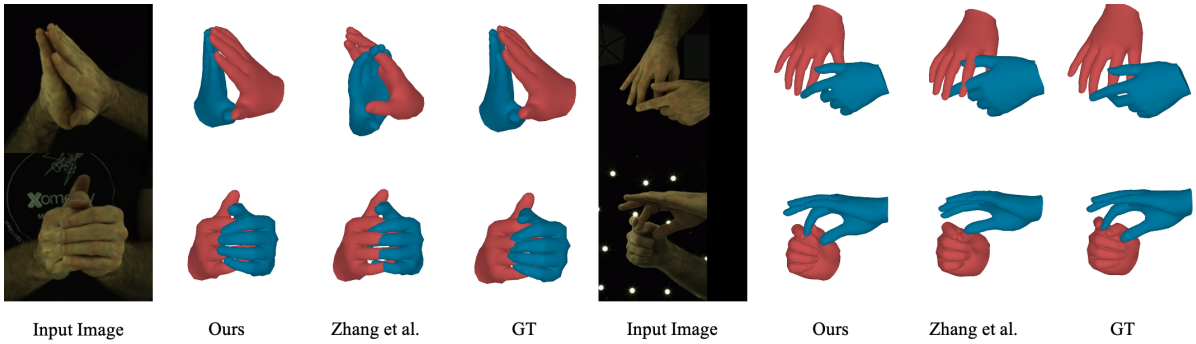

Figure 6. Qualitative comparison with Zhang *et al*. [46] on InterHand2.6M dataset. Our method produces more accurate two-hand poses, while Zhang *et al*. [46] produces more collisions and miscalculates relative depth between the left and right hands.

	MPJPE	MPVPE
† Zimmermann *et al*. [53]	36.36	-
† Zhou *et al*. [52]	23.48	23.89
† Boukhayma *et al*. [3]	16.93	17.98
† Spurr *et al*. [36]	15.40	-
Moon *et al*. [25]	16.00	-
Zhang *et al*. [46]	13.48	13.95
Ours	**8.79**	**9.03**

Table 1. Comparison on InterHand2.6M. † means single hand methods whose results are taken from [46]. We report MPJPE and MPVPE in mm, the lower the better. Our method outperforms all other methods by a huge margin.

gresses 3D skeletons of two hands directly. Another is Zhang *et al*. [46], which predicts the pose and shape parameters of two MANO [30] models. For a fair comparison, we run their released source code on the same subset of InterHand2.6M [25] to that we utilize (see Sec. 5.1). Comparison results are shown in Tab. 1 and Fig. 8. It is clearly shown in Tab. 1 that our method significantly reduces MPJPE and MPVPE. We attribute this success to the dense mesh reasoning ability of GCN and our novel attention based modules, which better align the mesh with the input image. The PCK curve in Fig. 8 further demonstrates the superior performance of our method at all error threshold levels.

5.5. Ablation study

Baseline GCN. We train a baseline GCN model by directly modifying the GCN decoder of Ge *et al*. [12] for two-hand output (see 'GCN baseline' in Tab. 2). Although directly leveraging GCN structure shows excellent numeric performance, inaccurate interaction reconstructions still exist without the attention modules.

Adding Attention Modules. Based on 'GCN baseline', we first add the CHA module to model interaction context (+CHA) and then add the PIFA module to further enhance vertex-mesh alignment (+CHA +PIFA), as shown in Tab. 2.

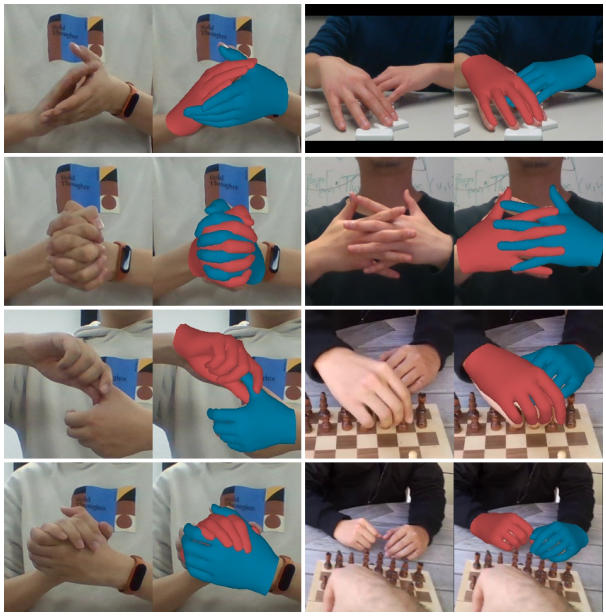

Figure 7. Qualitative results on in-the-wild images. Left 4 cases are real-life data captured using a USB camera. Top right 2 cases are taken from RGB2Hands [40] videos. Bottom right 2 cases are taken from EgoHands [2] videos.

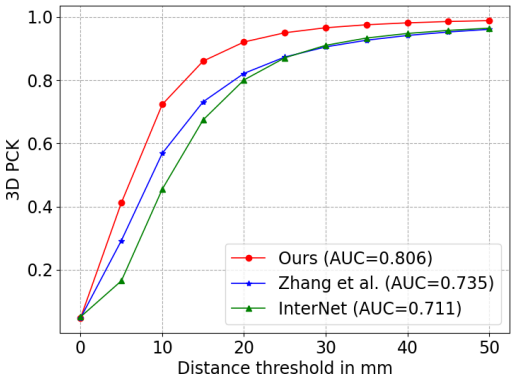

Figure 8. Comparison on InterHand2.6M dataset. 'Zhang et al' refers to [46], while 'InterNet' refers to [25].

By modeling interaction context with CHA, we achieve more than 0.6mm performance gain, proving the effectiveness of CHA for occlusion handling. By adding PIFA, our method further achieves more than 0.5mm performance improvement, affirming the ability of PIFA for vertex-image alignment. Qualitative comparison is shown in Fig. 9.

Pyramid or Not. Note that our model utilizes pyramid image features with increasing resolutions ($8\times8 \rightarrow 16\times16 \rightarrow 32\times32$). By removing pyramid structure, we use the consistent small (8×8) or large (32×32) image features in all the three IntagHand blocks (see 'IFA-8' and 'IFA-32' in Tab. 2). Similar to [24], we find that using the small image feature performs better than using the large one. More importantly, our pyramid structure further improves the reconstruction accuracy by leveraging both the global and local information for mesh regression.

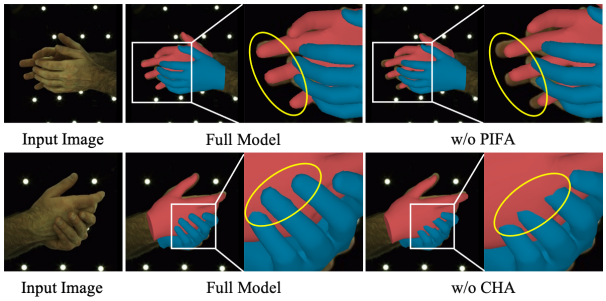

Figure 9. Qualitative ablation study on InterHand2.6M. 'w/o PIFA' means removing PIFA module from the full model, 'w/o CHA' means removing CHA module. It is shown that PIFA helps to align vertices and image features (top row), and CHA helps to address mutual occlusions (bottom row).

	MPJPE	MPVPE
GCN baseline	9.97	10.63
GCN + CHA	9.34	9.59
GCN + CHA + IFA-32	8.90	9.16
GCN + CHA + IFA-8	8.83	9.07
GCN + CHA + PIFA(**Ours**)	**8.79**	**9.03**

Table 2. Ablation study of module choice on InterHand2.6M.

6. Discussion

Conclusion. We present the interacting attention graph hand (IntagHand) method to reconstruct two interacting hands from a single RGB image. Specifically, we introduce a novel pyramid image feature attention (PIFA) module to formulate the attention relationship between hand meshes and image features, together with a novel cross-hand attention (CHA) module to encode the interaction context between two hands. Comprehensive experiments demonstrate the supreme performance of our network on InterHand2.6M dataset and in-the-wild images, and verify the effectiveness of our PIFA and CHA modules.

Limitation & Impact. The major limitation of our method is the absence of explicit mesh collision handling, resulting in occasional mesh intersections between hands. Note that, it is possible for our method to work with more than two hands where a preliminary detection network is necessary to extract hand regions and predict the hand number in each region. It is also possible to extend our method to other 2-way interactions (hand-object, human-human, *etc.*) as long as the subjects are encoded as vertex feature.

Acknowledgement: This paper is sponsored by NSFC No.62125107, NSFC No.62171255 and National Key R&D Program of China (2021ZD0113503).

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*, 2019. 2

[2] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *ICCV*, 2015. 6, 8

[3] Adnane Boukhayma, Rodrigo De Bem, and Philip H. S. Torr. 3D hand shape and pose from images in the wild. In *CVPR*, 2019. 1, 7

[4] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3D hand pose estimation from monocular rgb images. In *ECCV*, 2018. 2

[5] 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*, 2021. 2

[6] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020. 2

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 2016. 3, 4

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[9] I.S. Dhillon, Yuqiang Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *TPAMI*, 2007. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 5

[11] Zicong Fan, Adrian Spurr, Muhammed Kocabas, Siyu Tang, Michael J. Black, and Otmar Hilliges. Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation. In *3DV*, 2021. 1, 2, 6

[12] 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*, 2019. 1, 2, 3, 4, 7

[13] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3D annotation of hand and object poses. In *CVPR*, 2020. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 6

[15] Tony Heap and David Hogg. Towards 3D hand tracking using a deformable model. In *FG*, 1996. 2

[16] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 2019. 1, 2

[17] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018. 2

[18] Dong Uk Kim, Kwang In Kim, and Seungryul Baek. End-to-end detection and pose estimation of two interacting hands. In *ICCV*, 2021. 1, 2, 3

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[20] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 3

[21] Dominik Kulon, Riza Alp Güler, Iasonas Kokkinos, Michael Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 1, 2

[22] Nikolaos Kyriazis and Antonis Argyros. Scalable 3D tracking of multiple interacting objects. In *CVPR*, 2014. 1, 2

[23] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 2

[24] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 2, 3, 4, 5, 8

[25] 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*, 2020. 1, 2, 3, 5, 6, 7, 8

[26] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3D hand tracking from monocular rgb. In *CVPR*, 2018. 1, 2

[27] 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. In *SIGGRAPH*, 2019. 1, 2

[28] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012. 1, 2

[29] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3D faces using convolutional mesh autoencoders. In *ECCV*, 2018. 3

[30] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. In *SIGGRAPH Asia*, 2017. 2, 3, 7

[31] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3D hand and body motion capture by regression and integration. In *ICCVW*, 2021. 2

[32] Yu Rong, Jingbo Wang, Ziwei Liu, and Chen Change Loy. Monocular 3D reconstruction of interacting hands via collision-aware factorized refinements. In *3DV*, 2021. 2, 3

[33] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 2

[34] Breannan Smith, Chenglei Wu, He Wen, Patrick Peluse, Yaser Sheikh, Jessica K. Hodgins, and Takaaki Shiratori.

Constraining dense hand surface tracking with elasticity. *TOG*, 2020. 2

[35] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints. In *ECCV*, 2020. 2

[36] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 7

[37] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3D hand-mesh reconstruction. In *ICCV*, 2021. 2, 3, 4

[38] Jonathan Taylor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. Articulated distance fields for ultra-fast tracking of hands interacting. *TOG*, 2017. 1, 2

[39] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 2016. 1, 2

[40] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. RGB2Hands: real-time tracking of 3D hand interactions from monocular rgb video. In *SIGGRAPH Asia*, 2020. 1, 2, 3, 6, 8

[41] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3D mesh models from single rgb images. In *ECCV*, 2018. 3

[42] Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. Video-based hand manipulation capture through composite motion control. *TOG*, 2013. 2

[43] Yangang Wang, Cong Peng, and Yebin Liu. Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *TCSVT*, 2018. 2

[44] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*, 2019. 2

[45] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 6

[46] 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*, 2021. 1, 2, 3, 5, 6, 7, 8

[47] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. PyMAF: 3D human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, 2021. 2, 3, 4

[48] 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*, 2021. 2

[49] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *ICCV*, 2019. 1, 2

[50] Yuxiang Zhang, Zhe Li, Liang An, Mengcheng Li, Tao Yu, and Yebin Liu. Lightweight multi-person total motion capture using sparse multi-view cameras. In *ICCV*, 2021. 2

[51] Yuxiao Zhou, Marc Habermann, Ikhsanul Habibie, Ayush Tewari, Christian Theobalt, and Feng Xu. Monocular real-time full body capture with inter-part correlations. In *CVPR*, 2021. 2

[52] 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*, 2020. 1, 2, 7

[53] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single rgb images. In *ICCV*, 2017. 1, 2, 7

[54] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max J. Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019. 1, 2