Mining Multi-View Information: A Strong Self-Supervised Framework for Depth-based 3D Hand Pose and Mesh Estimation

Pengfei Ren, Haifeng Sun, Jiachang Hao, Jingyu Wang*, Qi Qi, Jianxin Liao
State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
rpf, hfsun, haojc, wangjingyu, qiqi8266@bupt.edu.cn; jxlbupt@gmail.com

Abstract

In this work, we study the cross-view information fusion problem in the task of self-supervised 3D hand pose estimation from the depth image. Previous methods usually adopt a hand-crafted rule to generate pseudo labels from multi-view estimations in order to supervise the network training in each view. However, these methods ignore the rich semantic information in each view and ignore the complex dependencies between different regions of different views. To solve these problems, we propose a cross-view fusion network to fully exploit and adaptively aggregate multi-view information. We encode diverse semantic information in each view into multiple compact nodes. Then, we introduce the graph convolution to model the complex dependencies between nodes and perform cross-view information interaction. Based on the cross-view fusion network, we propose a strong self-supervised framework for 3D hand pose and hand mesh estimation. Furthermore, we propose a pseudo multi-view training strategy to extend our framework to a more general scenario in which only single-view training data is used. Results on NYU dataset demonstrate that our method outperforms the previous self-supervised methods by 17.5% and 30.3% in multi-view and single-view scenarios. Meanwhile, our framework achieves comparable results to several strongly supervised methods.

1. Introduction

3D hand pose estimation plays an essential role in human-computer interaction, virtual reality, and augmented reality. With the development of deep learning and the increase of the amount of labeled data, depth-based 3D hand pose estimation has made significant progress [3, 12, 13, 20, 31, 32, 49, 54, 59]. However, acquiring large-scale hand datasets with 3D hand pose and mesh annotations is time-consuming and labor-consuming. Meanwhile, even after careful design and manual correction, the annotation quality of the existing automatic or semi-automatic annotation algorithms is difficult to guarantee [33, 57, 64].

Recently, some methods [8, 53, 55] achieve accurate 3D hand pose estimation through self-supervised learning. These methods introduce a 3D hand model into the neural network and optimize the network by penalizing the differences between the hand model and the input depth image. As mentioned in [53, 55], adopting multi-view information during training is the key to the success of the self-supervised methods. Complementary multi-view information can alleviate the uncertainty of estimation caused by self-occlusion or holes. The previous methods adopt a hand-crafted rule, *e.g.*, taking median value, to aggregate the estimated poses from multiple views as pseudo labels in order to supervise the network training in each view.

However, the hand-crafted rule only considers the coordinate information of the joint itself in multiple views. This method ignores the rich semantic information in visual features in each view and ignores the complex dependencies between different hand regions in different views. Thus, this method is susceptible to interference from the low-quality estimations that frequently occurs in a self-supervised framework. To better exploit multi-view information, some multi-view human pose estimation methods propose to perform pixel-wise cross-view interaction in 2D feature space by establishing point-to-point correspondence [19, 29, 38, 58, 67]. However, performing pixel-by-pixel matching based on feature similarity is computationally expensive and redundant. Meanwhile, these methods are sensitive to self-occlusion and holes [19]. For example, when some hand regions in one view are occluded or missing, the local features of these regions are difficult to be detected and matched robustly in other views.

To solve these problems, we propose a cross-view fusion network to fully exploit multi-view information to generate more accurate and robust pseudo labels. Specifically, we first encode high-dimensional visual features and the estimated hand pose in each view into multiple semantic nodes.

[*] Corresponding author

Then, we adopt a hierarchical graph convolutional network to perform intra-view and cross-view information interaction according to the hand bone structure and the cross-view joint correspondence. Furthermore, when constructing the graph nodes, we adopt a group-wise confidence encoding strategy to prevent high-quality features from being corrupted by low-quality features in information passing. Our method can fully mine the rich semantic information in each view and efficiently model the dependencies between different views. Meanwhile, performing cross-view information interaction according to the intrinsic hand structure avoids complex pixel-by-pixel matching and reduces the interference of self-occlusion and depth holes.

Based on the cross-view fusion network, we propose a strong self-supervised framework for 3D hand pose and mesh estimation. Considering that the multi-view setup may increase the difficulty of data collection and limits the application scenarios of our method, we further extend our framework to a more general scenario in which only single-view training data is used. By treating different augmented samples of the same input data as different view images, we propose a pseudo multi-view training strategy for the single-view scenario. Unlike previous methods [62,66] that perform self-supervised learning by maintaining the predictive consistency between different augmented samples, our method aggregates the multiple pseudo views information to generate more accurate estimations, which can more effectively guide network training in each view.

We conduct experiments on three 3D hand pose estimation datasets (NYU [52], ICVL [49], and MSRA [47]). On NYU dataset, our method improves the state-of-the-art (SOTA) self-supervised methods by 17.5% in the multi-view scenario and by 30.3% in the single-view scenario. Meanwhile, our method achieves comparable results to strongly supervised methods. Qualitative experiments on ICVL and MSRA datasets show that our method generates more accurate hand poses than the annotations. We evaluate our method in real-world scenarios and the results also verify the effectiveness of our model. Code is available at https://github.com/RenFeiTemp/MMI.

Our contributions can be summarized as follows:

- We propose a cross-view fusion network to fully mine the rich semantic information in each view and efficiently model the dependencies between different views.
- We propose a strong self-supervised framework for depth-based 3D hand pose and mesh estimation. Furthermore, we extend our framework to a more general scenario in which only single-view training data is used.
- Our method outperforms existing self-supervised methods by a large margin and achieves comparable results to several strongly supervised approaches. In addition, our method can yield more accurate hand poses than the annotations of some existing datasets.

2. Related Work

2.1. Depth-based 3D Hand Pose Estimation

Depth-based 3D hand pose estimation can be categorized into three classes: model-based methods, learning-based methods, and hybrid methods. Model-based methods [23,45,48,50,51,63] adopt a pre-defined 3D hand model to fit the depth input by minimizing a set of model-fitting terms. This kind of method requires no labeled data, but it is sensitive to the parameters of model initialization and the design of the energy function. In addition, it is easily trapped in error accumulation. Learning-based methods [3,4,11–13,15,17,20,30,31,34,35,41,54,59,69] use labeled data to learn a hand pose estimator. With the development of deep learning, learning-based methods achieve more accurate and robust estimates compared with the model-based method. However, these methods may overfit the annotation errors of the dataset [33]. Hybrid methods [37, 45, 52] use a learning-based method to initialize the hand model or re-initializing when tracking fails and perform temporal hand tracking using model-based methods.

2.2. Self-supervised 3D Hand Pose Estimation

Recently, given that obtaining accurate 3D pose and mesh annotations is time-consuming and labor-intensive, some depth-based methods [8,53,55] and RGB-based methods [5, 6] propose to perform self-supervised training on unlabeled real data. RGB-based self-supervised methods require additional annotations such as 2D poses [6] or 3D poses [5] to assist the self-supervised training. Benefiting from the rich geometric structure information of depth data, with the help of some prior loss items, depth-based methods only require synthetic data to pre-train the network. Depth-based methods [53, 55] usually adopt multi-view information to alleviate the estimation ambiguity caused by self-occlusion and image missing. On the one hand, they minimize the difference between the estimated hand model from one view and the depth data under other views. On the other hand, they take the median value of the multi-view estimations as pseudo labels to supervise the network's training in each view. However, their method discards the visual features of each view while ignoring the dependencies between cross-view hand regions.

2.3. Multi-view 3D Human Pose Estimation

Multi-view information has been widely explored in 3D human pose estimation. Some methods [21, 24, 27, 43, 56] train the network in a weakly-/self-supervised manner by constraining the consistency of the estimated poses in multiple views. However, these methods do not really aggregate multi-view information to obtain more accurate results. Some methods [22, 36, 38] project estimated 2D heatmaps in each view to a 3D volume and estimate the 3D pose from

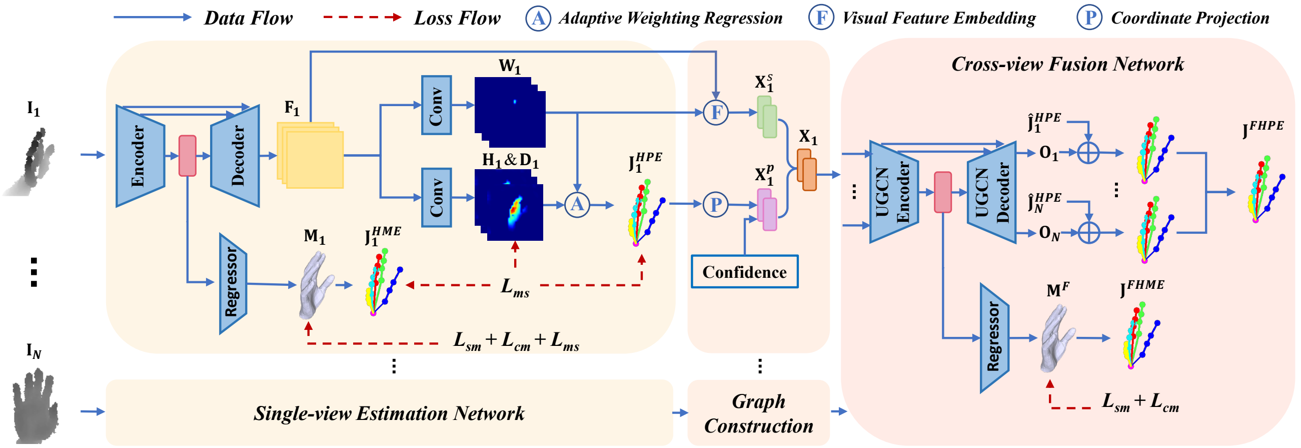

Figure 1. Overview of the framework. The single-view estimation network predicts 3D hand pose and 3D hand model in parallel. The cross-view fusion network utilizes visual features and estimated hand poses from multiple views to generate more accurate results in the candidate view. Here, we choose view 1 as the candidate view. First, we use synthetic data for pre-training. Then, the two networks adopt a single-view model-fitting loss L_{sm} and a cross-view model-fitting loss L_{cm} to perform self-supervised learning on unlabeled real data. Meanwhile, the single-view estimation network is trained to produce results consistent with the cross-view fusion network, which is called multi-view self-distillation loss L_{ms}. During the inference, only the single-view estimation network is utilized.

the 3D volumes with volumetric convolutional networks or pictorial structures model, which is computationally complex. Some methods [38, 58, 67] use the epipolar geometry to aggregate estimated 2D heatmaps from different views. However, these methods ignore the semantic information contained in image features. Some methods [19, 29] perform cross-view fusion on image features, enabling the 2D image feature to perceive 3D geometric information. However, pixel-wise interaction is computationally demanding, and the feature matching process based on feature similarity is sensitive to occlusion and image missing. Our method encodes visual features into compact nodes and performs information interaction according to the semantic relationship between different hand parts, which is robust to occlusion and can aggregate multi-view information efficiently.

2.4. Graph Convolutional Network

Graph Convolutional Network (GCN) shows a strong ability to perform message passing on structured data. It is widely used in many visual tasks, *e.g.*, action recognition [7, 46, 61], human pose estimation [6, 28, 65, 68, 71] and hand pose estimation [2, 9, 11, 14, 25]. Kulon *et al.* [25] and Ge *et al.* [14] adopt GCN for 3D hand mesh generation. Doosti *et al.* [9] propose an adaptive GCN to convert 2D hand pose to 3D. They construct the graph node by concatenating the global features extracted by CNN and the 2D coordinates of each joint, which may introduce the features of irrelevant regions. Fang *et al.* [11] adopt an attention mechanism to construct graph node features from the visual feature and perform graph reasoning to capture the relationship between nodes in order to enhance the original visual feature maps. Cai *et al.* [2] propose a hierarchical GCN to estimate the 3D hand pose from a short sequence of 2D poses. Different from these methods, we adopt GCN to perform cross-view information interaction and our method fully exploits the multi-view information, including the visual features, joint coordinates, and estimation confidence.

3. Method

As shown in Fig. 1, our framework consists of two networks, one is a single-view estimation network that predicts 3D hand pose and 3D hand model from a single-view image, and the other is a cross-view fusion network that fuses multi-view information to generate more accurate results.

3.1. Single-view Estimation Network

Self-supervised 3D hand pose estimation methods [8, 53, 55] introduce a hand model into the neural network and adopt a set of carefully designed model-fitting terms to train the network. However, directly regressing model parameters is a highly non-linear process and is difficult to perceive fine-grained image features, resulting in image-model misalignment. Thus, we adopt an encoder-decoder structure, using the global features from the encoder to regress the hand model parameters, and using 2D feature maps from the decoder to perform pixel-wise pose estimation.

For 3D hand model estimator (HME), we adopt a parametric hand model, MANO [44] and use a fully connected (FC) layer to regress MANO parameters. We can obtain a 3D hand mesh $\mathbf{M} \in \mathbb{R}^{778\times3}$ and 3D hand joints $\mathbf{J}^{HME} \in \mathbb{R}^{21\times3}$ from the estimated hand model. For hand pose estimator (HPE), we adopt a 3D heat map $\mathbf{H} \in \mathbb{R}^{21\times h\times w}$ and

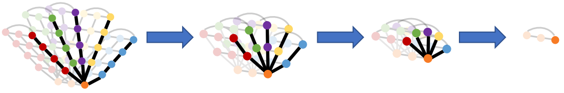

Figure 2. Hierarchical graph structure in the encoding process of UGCN. The spatial edges represent the physical connection of joints. The cross-view edges connect the same joints between any two different views. For easy illustration, we only plot a part of the cross-view edges.

a directional unit vector $\mathbf{D} \in \mathbb{R}^{(21\times 3)\times h\times w}$ as intermediate representations [20, 54], which represent 3D Euclidean distance and 3D unit direction from each pixel to the target joint, respectively. Specifically, for a pixel i in the input depth image $\mathbf{I}$, the corresponding ground-truth value $\hat{\mathbf{H}}_j(i)$ and $\hat{\mathbf{D}}_j(i)$ about the target joint j can be formulated as:

$$\hat{\mathbf{H}}_j(i) = \begin{cases} \lambda \frac{d-\|\mathbf{I}_i-\mathbf{J}_j\|_2}{d} & \|\mathbf{I}_i - \mathbf{J}_j\|_2 \le d, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

$$\hat{\mathbf{D}}_j(i) = \begin{cases} \lambda \frac{\mathbf{I}_i-\mathbf{J}_j}{\|\mathbf{I}_i-\mathbf{J}_j\|_2} & \|\mathbf{I}_i - \mathbf{J}_j\|_2 \le d, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathbf{I}_i \in \mathbb{R}^3$ and $\mathbf{J}_j \in \mathbb{R}^3$ are the image plane coordinate of the pixel i and of the joint j respectively; λ is equal to zero if i belongs to the background, otherwise equal to one; d denotes the maximum distance from the pixel in the depth image to the target hand joint. Furthermore, we estimate a weight map $\mathbf{W} \in \mathbb{R}^{21\times h\times w}$ to indicate the reliability of the estimation result for each pixel to the target joint. The 3D coordinate $\mathbf{J}_j^{HPE}$ of the joint j can be obtained as follow:

$$\mathbf{J}_j^{HPE} = \sum_{i\in\mathbf{I}} \mathbf{W}_j(i)\left(\mathbf{I}_i - (d - d\mathbf{H}_j(i))\mathbf{D}_j(i)\right), \tag{3}$$

3.2. Cross-view Fusion Network

We consider a setting in which N spatially calibrated and temporally synchronized cameras capture the depth image of a single hand. Given the 2D image features $\mathbf{F}_c$ and the estimated hand pose $\mathbf{J}_c$ in each camera view c, we aim to fully mine the information in each view to predict more accurate results. However, performing pixel-by-pixel cross-view interaction requires a huge computation and is susceptible to interference from the self-occlusion and depth holes. Thus, we encode visual features and the estimated hand pose into multiple compact nodes and then adopt a U-shaped graph convolutional network (UGCN) to perform hierarchical cross-view information interaction and fusion.

Graph Construction. The cross-view graph is organized as an undirected graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each graph node $\mathcal{V}$ refers to a hand joint in a camera view. As shown in Fig. 2, the edges of the graph $\mathcal{E}$ are composed of spatial edges that represent physical skeleton connections among different joints and cross-view edges that connect the same joint across views. Let $\mathbf{X}_{cj} \in \mathbb{R}^Z$ denote a feature vector associated with each node j in a camera view c. $\mathbf{X}_{cj}$ consists of two parts: position features $\mathbf{X}_{cj}^p \in \mathbb{R}^{\frac{Z}{2}}$ and visual features $\mathbf{X}_{cj}^s \in \mathbb{R}^{\frac{Z}{2}}$. For position features, we project the hand pose $\mathbf{J}_c$ in view c to the candidate camera space as $\hat{\mathbf{J}}_c$ and encode the 3D coordinate of a joint j to $\mathbf{X}_{cj}^p$ through a FC layer. For visual features, we adopt an attention mechanism to reduce the influence of the features of irrelevant regions. Specifically, we obtain the node visual features $\mathbf{X}_{cj}^s$ by multiplying the normalized weight map $Softmax(\mathbf{W}_{cj})$ with each channel of the embedded feature maps $\varphi(\mathbf{F}_c)$ and performing a spatial global average pooling. Here, φ is an 1×1 convolutional layer used to embed the visual features into the node feature space.

U-shaped Graph Convolutional Network. We adopt an encoder-decoder structure to capture multi-scale dependencies between multi-view nodes. As shown in Fig. 2, for the encoding process, we gradually cluster the whole nodes based on the skeletal structure of the hand. For the decoding process, similar to [2, 60], we duplicate the coarse nodes to generate fine-grained nodes according to the inverse clustering process. We adopt the graph convolution to perform information interaction between nodes in the encoder. In the decoder, we adopt a per-node FC layer to fuse the same-scale nodes from the encoding and decoding processes. The global features from the encoder are used to regress MANO parameters, from which we can obtain the refined hand mesh $\mathbf{M}^F$ and refined hand pose $\mathbf{J}^{FHME}$. The enhanced node features from the decoder are used to estimate relative offsets $\mathbf{O}_c \in \mathbb{R}^{21\times 3}$ of the initial poses $\hat{\mathbf{J}}_c$, from which we can obtain the refined pose $\mathbf{J}^{FHPE}$ by averaging the results of all views.

Quality-aware Fusion. The fusion strategy introduced in the previous section treats all nodes equally, which does not consider the feature quality of each node. However, if a joint in one view cannot be observed due to self-occlusion or image hole, its initial estimated location is often inaccurate, and its visual features are likely corrupted. The weight map $\mathbf{W}$ represents the reliability of the estimation results of each pixel. Generally, the larger the maximum response value of the weight map, the higher the confidence of the estimated joints. Thus, we regard the maximum response of the normalized weight map as the quality of the node and encode it as part of the node features. In particular, since we do not supervise the weight map during training, the weight map tends to be adaptively distributed to joint-related regions. Therefore, the weight map of different joint has different shapes, resulting in significant differences in the interval of the maximum value of each joint. In order to solve this problem, we classify joints into multiple groups and then use different FC layers for different groups for quality embedding. In addition, the cross-view fusion network will re-estimate the quality of each node. Then, we perform a weighted average to obtain the refined pose from all views, which hardly improves the network performance but makes the self-supervised training process more stable.

3.3. Self-supervision Loss

The self-supervised loss L consists of four parts including a single-view model-fitting loss L_{sm}, a cross-view model-fitting loss L_{cm}, a multi-view self-distillation loss L_{ms} and a prior loss L_{prior}.

$$L = w_{sm}L_{sm} + w_{cm}L_{cm} + w_{ms}L_{ms} + w_{prior}L_{prior}, \quad (4)$$

where w_{sm}, w_{cm}, w_{ms} and w_{prior} are constant weights.

Single-view Model-fitting Loss. The single-view loss L_{sm} updates the network by penalizing the difference between the estimated 3D hand model and the input depth data. Similar to [53, 55], L_{sm} consists of a model-to-data term L_{m2d} and a data-to-model term L_{d2m}. L_{m2d} aligns the hand model as close as possible to the input depth image. We adopt a differentiable renderer [40] to render the estimated 3D hand mesh to a depth image $\mathbf{I}^r$.

$$L_{m2d} = \sum_{i \in \mathbf{I}} \left| \mathbf{I}_i^r - \mathbf{I}_i \right|. \quad (5)$$

L_{d2m} minimizes the distance between each point on the input depth image and its projection on to the estimated hand mesh $\mathbf{M}$.

$$L_{d2m} = \sum_{i \in \mathbf{I}} D\left(\mathbf{I}_i^{xyz}, \mathbf{M}\right), \quad (6)$$

where $\mathbf{I}_i^{xyz} \in \mathbb{R}^3$ are the world coordinates of the pixel i; D represents the 3D Euclidean distance of $\mathbf{I}_i^{xyz}$ to the closest triangular face in hand mesh $\mathbf{M}$.

Cross-view Model-fitting Loss. The cross-view model-fitting loss L_{cm} trains the network by maintaining multi-view consistency. Specifically, we project the estimated hand model from a source view to other view and evaluate the difference between the hand model and the depth image captured from other views according to L_{m2d} and L_{d2m}.

Multi-view Self-distillation loss. For multi-view self-distillation loss L_{ms}, the fusion results $\mathbf{J}^{FHPE}$ and $\mathbf{M}^F$ are projected back to the target camera view c as pseudo labels $\widetilde{\mathbf{J}}_c$ and $\widetilde{\mathbf{M}}_c$ to guide the training of the single-view estimation network. For the HPE, we adopt $L_{ms,p}$ and $L_{ms,i}$ to supervise estimated pose and intermediate representations.

$$L_{ms,p} = \sum_{c}\sum_{j} L1\left(\mathbf{J}_{cj}^{HPE}, \widetilde{\mathbf{J}}_{cj}\right), \quad (7)$$

$$L_{ms,i} = \sum_{c}\sum_{j} L1\left(\mathbf{H}_{cj}, \widetilde{\mathbf{H}}_{cj}\right) + \sum_{c}\sum_{j} L1\left(\mathbf{D}_{cj}, \widetilde{\mathbf{D}}_{cj}\right). \quad (8)$$

Here, $\widetilde{\mathbf{H}}_{cj}$ and $\widetilde{\mathbf{D}}_{cj}$ are the pseudo labels of the 3D heatmap and unit 3D directional vector fields, which are generated from $\widetilde{\mathbf{J}}_{cj}$. $L1$ represents Smooth L1 loss in [3, 17, 42] to make the loss less sensitive to the outliers.

For the HME, we adopt $L_{ms,m}$ to supervise the hand joint and the hand mesh of the estimated hand model.

$$L_{ms,m} = \sum_{c}\sum_{j} L1\left(\mathbf{J}_{cj}^{HME}, \widetilde{\mathbf{J}}_{cj}\right) + \sum_{c}\sum_{m} L1\left(\mathbf{M}_{cm}, \widetilde{\mathbf{M}}_{cm}\right). \quad (9)$$

Prior Loss. To make the estimated hand model plausible, we introduce the prior loss L_{prior}, including a shape term [18] and a collision term [53]. To avoid extreme mesh deformations, the shape term constrains the predicted mesh shape as close as possible to the average shape. The collision term is adopted to avoid self-intersection, which is achieved by placing multiple spheres in the hand model and then penalizing overlaps between these spheres.

3.4. Training

The training of our framework includes three stages. The first stage is pre-training the single-view estimation network using synthetic data. The second stage is pre-training the cross-view fusion network using multi-view synthetic data. The last stage is fine-tuning the whole network on multi-view real data. In particular, we use both labeled synthetic data and unlabeled real data in the last stage to stabilize self-supervised training. To reduce the domain gap between the synthetic and the real data, we adopt CycleGAN [70] to translate synthetic data into more realistic data. Meanwhile, we randomly erase some depth regions to simulate the depth holes in the real depth image.

Requiring multi-view data limits the application scenarios of the self-supervised method. Thus, we extend our framework to a more general single-view scenario. Specifically, in the third stage, we treat different augmented samples of the same depth image as pseudo multi-view images. Compared with the multi-view scenario, we adopt a stronger data augmentation for the unlabeled real data to increase the diversity of generated images. By default, we generate three augmented samples for a single depth image.

4. Experiment

4.1. Dataset and Evaluation Metrics

NYU dataset [52] is a publicly available multi-view depth dataset, including a frontal view and two side views. For each view, it contains 72K training images and 8.2K testing images. It is a challenging dataset with a wide coverage of hand poses and image noise. Similar to the previous methods [53,55], we only adopt the ground-truth annotation to calculate camera extrinsics. **ICVL dataset** [49] consists of 22K training and 1.6K testing depth images captured by

Fusion Method	J	F	C	GC	HME	HPE
No-Fusion					23.71	14.14
Average	✓				23.03	13.34
Weight	✓				22.76	12.13
Median	✓				22.15	11.68
UGCN	✓				13.05	11.16
UGCN	✓	✓			12.62	10.46
UGCN	✓	✓	✓		11.83	10.32
UGCN	✓	✓		✓	11.64	10.29

Table 1. Effect of the cross-view fusion. We report the mean joint error (mm) of the fusion results for HME and HPE. J, F, C, and GC indicate the use of joint coordinates, visual features, confidence, and group-wise confidence encoding strategy, respectively.

an Intel Real-sense camera. The training images are collected from 10 subjects and the testing images are collected from 2 subjects. The annotation of the hand pose contains 16 joints. **MSRA dataset** [47] contains 76.5K images captured by an Intel Real-sense camera from 9 subjects. Each subject contains 17 hand gestures with 21 annotated joints.

We evaluate our method using two widely used metrics: the mean joint error and the percentage of successful frames. The mean joint error is the mean 3D Euclidean distance between the predicted coordinates and the ground-truth coordinates for each joint over the whole test set. The percentage of successful frames is defined as the proportion of good frames in all testing frames. If the maximum value of the joint error in a frame is less than a certain threshold, it will be judged as a good frame. Considering the different joint settings between the MANO and the annotation in these datasets, we ignore three joints (two wrist joints and one palm joint) of the NYU and ignore the palm joint of the ICVL and MSRA during evaluation. Meanwhile, we slightly adjust MANO's default joint settings to match the joint settings in these datasets better.

4.2. Implementation Details

We train and evaluate our method on a single server with an NVIDIA RTX 3090 GPU. The network is implemented within PyTorch and trained using AdamW [26] with a batch size of 32. We adopt the hand center provided by [31] to crop the hand from the original depth image. We resize the cropped image to a fixed size of 128×128. The depth values are normalized to [-1, 1] for the cropped image. To generate synthetic data, we randomly sample 200K hand pose data from the BigHand2.2M dataset [64]. Then, we use an iterative optimization method [1] to obtain the MANO parameters and render the 3D hand mesh obtained from the MANO as the synthetic depth image. More details are provided in the supplementary material.

4.3. Ablation Study

Study of the Cross-view Fusion. First, we compare the performance of the hand-crafted fusion method and our UGCN-based fusion method. Inspired by previous work [53], we adopt three hand-crafted fusion methods to directly aggregate the estimated results from multiple views, including taking average value (Average), taking median value (Median), and weighted average according to the confidence (Weight). As shown in Table 1, the hand-crafted methods significantly reduce the error of the estimated hand pose, but have a relatively small improvement on the estimated hand model, which is due to the low quality of the initial hand model. When only joint coordinates (J) are used, the UGCN-based fusion method further improves the accuracy of pose estimation and dramatically improves the accuracy of the estimated 3D hand model, which is important for subsequent self-supervised learning. It shows that it is meaningful to model the dependencies between multi-view joints and perform information interaction. Then, we show the effect of adopting different node features. Adopting visual features (F) brings notable improvement for the hand pose and hand model. Adopting the joint confidence (C) helps the mean joint error of the estimated hand model decrease by 6.3%. Group-wise confidence encoding strategy (GC) further improves the performance of cross-view fusion. The above results show that it is necessary to fully exploit the semantic information of each view.

L_{sm}	L_{cm}	L_{ms}	L_{prior}	HME	HPE
✓			✓	17.49	12.47
✓	✓		✓	13.49	12.32
✓		✓	✓	12.52	10.71
✓	✓	✓	✓	12.43	10.50
✓	✓	✓		12.39	10.66

Table 2. Effect of the self-supervised loss during fine-tuning. We report the mean joint error (mm) of HME and HPE.

Effect of the Self-supervised Loss. We study the contributions of four self-supervised loss terms during fine-tuning. As shown in Table 2 and Fig. 3 (a), when adopting the single-view model-fitting loss L_{sm}, the performances of HME and HPE are significantly improved. However, the accuracy of the estimated hand model is still unsatisfactory. Adopting multi-view information significantly improves the performance of the network. By fully mining semantic information from multiple views, the multi-view self-distillation loss L_{ms} brings more improvement than the cross-view model-fitting loss L_{cm}. L_{ms} greatly decreases the error of HME from 17.49 mm to 12.52 mm (28.4%) and the HPE from 12.47 mm to 10.71 mm (14.1%). Besides, although the prior loss L_{prior} has little effect on accuracy,

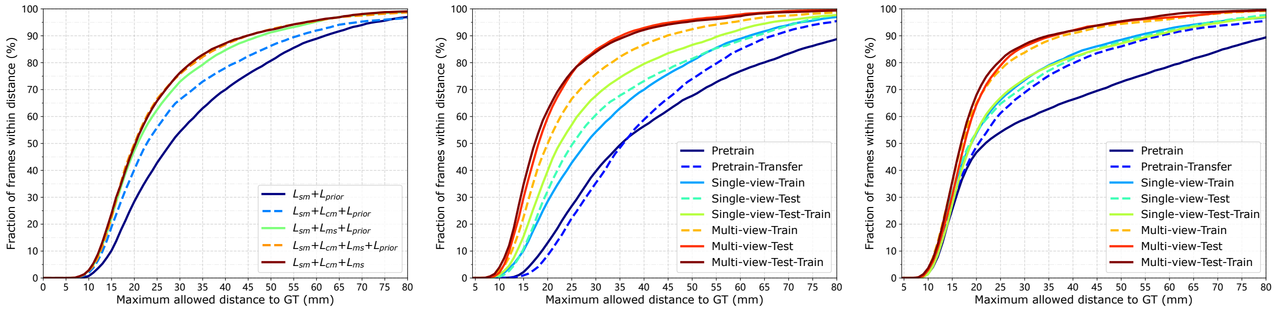

Figure 3. The proportions of successful frames over different thresholds. (a) The impact of different self-supervised losses on HME. (b) The impact of different training data on HME. (c) The impact of different training data on HPE.

	Transfer	Train Data	Test Data	HME	HPE
Pretrain				24.32	16.21
	✓			23.71	14.14
Multi-view Finetune	✓	✓		12.43	10.50
	✓		✓	10.81	10.43
	✓	✓	✓	10.49	10.23
Single-view Finetune	✓	✓		15.05	12.04
	✓		✓	14.90	11.93
	✓	✓	✓	13.49	11.95

Table 3. Effect of the training data during pre-training and fine-tuning. We report the mean joint error (mm) of HME and HPE.

Method	**NYU**	**ICVL**	**MSRA**
Strongly Supervised Method			
DeepModel [69]	19.02	11.73	-
Pose-REN [3]	12.05	6.90	8.64
DenseReg [54]	9.60	7.30	7.15
CrossInfoNet [10]	10.43	6.82	7.89
Point-to-Point [16]	9.30	6.35	7.65
V2V-PoseNet [31]	8.43	6.34	-
A2J [59]	8.61	6.52	-
SRN [42]	7.95	6.34	7.13
FeatureMapping [39]	7.81	-	-
AWR [20]	7.53	6.06	7.17
Sinlge-View Self-supervised Method			
SM [53]	17.79	-	-
MM [55]	16.96	-	-
Ours-HME	12.91 (12.40)	14.44	12.97
Ours-HPE	11.82 (11.07)	15.57	12.56
Multi-View Self-supervised Method			
SM [53]	12.26	-	-
MM [55]	13.09	-	-
Ours-HME	11.78 (11.28)	-	-
Ours-HPE	10.11 (10.33)	-	-

Table 4. Comparison with SOTA methods on NYU, ICVL, and MSRA datasets. The parenthesis indicates the performance of the method crops the hand using annotations.

it prevents unreasonable self-intersections and extreme deformations of the hand mesh.

Effect of the Training Data. We investigate how different training data, including synthetic data for pre-training and real data for fine-tuning, influences the resulting network. As shown in Table 3, adopting style transfer improves the performance of pre-training, especially for HPE. When investigating the impact of real data on fine-tuning, we first train only with the testing samples to check how well our self-supervised method can fit the training data. Then, we show the results of training the network using both testing and training data. As shown in Table 3 and Fig. 3, contrary to previous works [53, 55], training with only testing samples outperforms training with only training samples in our method. This indicates that our method can effectively exploit the 3D geometric information in the depth data and has a stronger fitting ability. Meanwhile, the performance of our method is further improved when training with both training and testing data. In supplementary material, we show that our method has good semi-supervised learning ability and robustness to the sampling strategy of synthetic data.

4.4. Comparisons with State-of-the-arts

Similar to previous methods [53], we adopt a two-stacked single-view estimation network, which further improves the performance of our method. We compare our method with SOTA self-supervised methods, including parametric model-based method (PM) [8], sphere-model based method (SM) [53], and mesh-model based method (MM) [55]. These methods do not describe which type of hand center they used to crop the hand image. Thus, we also report the results of cropping the hand image using the hand center obtained from joint annotations [32]. As shown in Table 4, on NYU dataset, our method reduces the mean joint error by 17.5% (10.11 mm vs 12.26 mm) and 30.3% (16.96 mm vs 11.82 mm) in multi-view and single-view

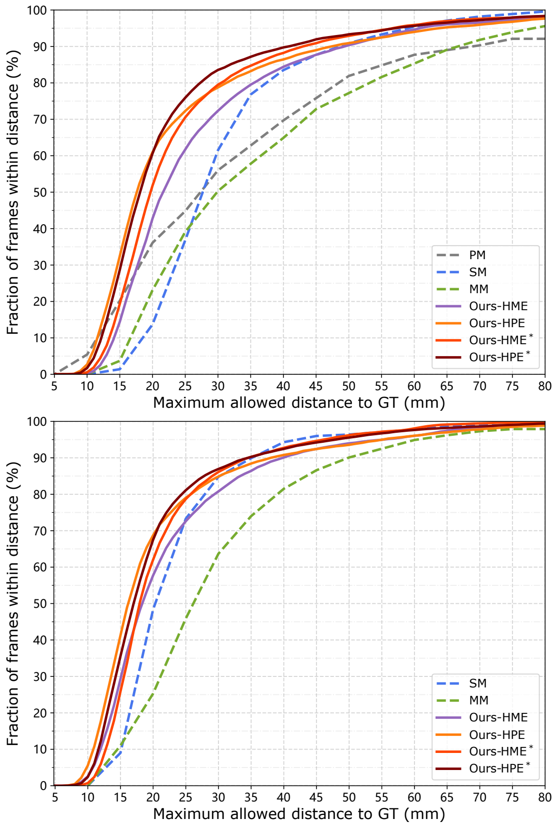

Figure 4. Comparison with SOTA self-supervised methods on NYU dataset in single-view scenario (top) and multi-view scenario (bottom). Methods with $*$ crop the hand image using the annotation of the joints.

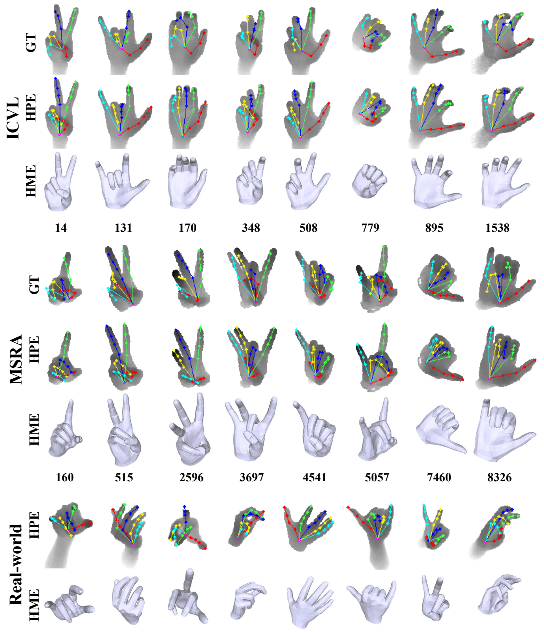

Figure 5. Qualitative results for ICVL, MSRA and real-world. The numbers represent the frame ID in the test set.

scenarios compared with SOTA self-supervised methods. As shown in Fig. 4, on the error threshold of 20 mm, our method significantly increases the percentage of successfully frames from 48% to 67% in the multi-view scenario and from 36% to 60% in the single-view scenario. Meanwhile, when comparing with strongly supervised methods [3, 10, 16, 20, 31, 39, 54, 59, 69], our model can get comparable performance. However, on ICVL and MSRA datasets, the mean joint error of our method is much higher than the strongly supervised methods. We attribute this to the fact that the two datasets have significant annotation errors and bias. As mentioned in previous works [33], the strongly supervised method tends to over-fit the incorrect annotations. We show some examples with large estimation errors. As shown in Fig. 5, our method predicts more accurate 3D hand pose than ground-truth (GT). Furthermore, we provide supplementary videos to show the performance of our method on the whole test set of the ICVL and MSRA datasets, which shows that our method appears systematically better. We also show the performance of our method on real-world data. During testing, our two-stacked network has an average run time of 9.4 ms per image (106 FPS) on a single NVIDIA RTX 3090 GPU

5. Conclusion and Limitations

In this paper, we propose a strong self-supervised framework for depth-based 3D hand pose and mesh estimation. First, we propose a cross-view fusion network to fully mine multi-view information, which provides accurate and robust guiding information for the single-view estimation network during self-supervised training. Then, by adopting a pseudo multi-view training strategy, we extend our framework to the single-view scenario. On the NYU dataset, our method outperforms the previous self-supervised methods by a large margin in both single-view and multi-view scenarios. On the ICVL and MSRA datasets, our method generates more accurate poses than annotations. However, in the multi-view scenario, our method must use the camera extrinsic parameters, which is unfriendly to the scene where the camera position is constantly changing. Besides, we find that although we adopt the collision loss, the estimated hand model still can not completely avoid self-intersection, especially for fingertips.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grants (62071067, 62171057, 62101064, 62001054), in part by the Ministry of Education and China Mobile Joint Fund (MCM20200202). This work was also supported by BUPT Excellent Ph.D. Students Foundation CX2020121.

References

[1] Anil Armagan, Guillermo Garcia-Hernando, Seungryul Baek, Shreyas Hampali, Mahdi Rad, Zhaohui Zhang, Shipeng Xie, MingXiu Chen, Boshen Zhang, Fu Xiong, et al. Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3d hand pose estimation under hand-object interaction. *arXiv preprint arXiv:2003.13764*, 2020. 6

[2] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2272–2281, 2019. 3, 4

[3] Xinghao Chen, Guijin Wang, Hengkai Guo, and Cairong Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 395:138–149, 2020. 1, 2, 5, 7, 8

[4] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6961–6970, 2019. 2

[5] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10451–10460, 2021. 2

[6] Zheng Chen, Sihan Wang, Yi Sun, and Xiaohong Ma. Self-supervised transfer learning for hand mesh recovery from binocular images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11626–11634, 2021. 2, 3

[7] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020. 3

[8] Endri Dibra, Thomas Wolf, Cengiz Oztireli, and Markus Gross. How to refine 3d hand pose estimation from unlabelled depth data? In *2017 International Conference on 3D Vision (3DV)*, pages 135–144, 2017. 1, 2, 3, 7

[9] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6608–6617, 2020. 3

[10] Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. Crossinfonet: Multi-task information sharing based hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9896–9905, 2019. 7, 8

[11] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. In *European Conference on Computer Vision*, pages 120–137. Springer, 2020. 2, 3

[12] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018. 1, 2

[13] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3593–3601, 2016. 1, 2

[14] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10833–10842, 2019. 3

[15] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 475–491, 2018. 2

[16] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 475–491, September 2018. 7, 8

[17] Hengkai Guo, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, and Huazhong Yang. Region ensemble network: Improving convolutional network for hand pose estimation. In *Proceedings of the IEEE International Conference on Image Processing*, pages 4512–4516, 2017. 2, 5

[18] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019. 5

[19] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7779–7788, 2020. 1, 3

[20] Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. Awr: Adaptive weighting regression for 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11061–11068, 2020. 1, 2, 4, 7, 8

[21] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5243–5252, 2020. 2

[22] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2019. 2

[23] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2540–2548, 2015. 2

[24] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view

geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1077–1086, 2019. 2

[25] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4990–5000, 2020. 3

[26] Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael M. Bronstein, and Stefanos Zafeiriou. Single image 3d hand reconstruction with mesh convolutions. In *Proceedings of the British Machine Vision Conference*, 2019. 6

[27] Yang Li, Kan Li, Shuai Jiang, Ziyue Zhang, Congzhentao Huang, and Richard Yi Da Xu. Geometry-driven self-supervised method for 3d human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11442–11449, 2020. 2

[28] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *European Conference on Computer Vision*, pages 318–334. Springer, 2020. 3

[29] Haoyu Ma, Liangjian Chen, Deying Kong, Zhe Wang, Xingwei Liu, Hao Tang, Xiangyi Yan, Yusheng Xie, Shih-Yao Lin, and Xiaohui Xie. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. *arXiv preprint arXiv:2110.09554*, 2021. 1, 3

[30] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7113–7122, 2020. 2

[31] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, June 2018. 1, 2, 6, 7, 8

[32] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 585–594, 2017. 1, 7

[33] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4957–4965, 2016. 1, 2, 8

[34] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. In *Proceedings of the Computer Vision Winter Workshop*, pages 21–30, 2015. 2

[35] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324, December 2015. 2

[36] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6988–6997, 2017. 2

[37] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, 2014. 2

[38] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4342–4351, 2019. 1, 2, 3

[39] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4663–4672, 2018. 7, 8

[40] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[41] Pengfei Ren, Haifeng Sun, Weiting Huang, Jiachang Hao, Daixuan Cheng, Qi Qi, Jingyu Wang, and Jianxin Liao. Spatial-aware stacked regression network for real-time 3d hand pose estimation. *Neurocomputing*, 437:42–57, 2021. 2

[42] Pengfei Ren, Haifeng Sun, Qi Qi, Jingyu Wang, and Weiting Huang. Srn: Stacked regression network for real-time 3d hand pose estimation. In *Proceedings of the British Machine Vision Conference*, page 112, 2019. 5, 7

[43] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8437–8446, 2018. 2

[44] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, 2017. 3

[45] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015. 2

[46] Lei Shi, Yifan Zhang, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 3

[47] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, June 2015. 2, 6

[48] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust

articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, volume 34, pages 101–114, 2015. 2

[49] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014. 1, 2, 5

[50] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics*, 35(4):143:1–143:12, 2016. 2

[51] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics*, 35(6):222:1–222:11, 2016. 2

[52] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169:1–169:10, 2014. 2, 5

[53] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3d hand pose estimation through training by fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10853–10862, 2019. 1, 2, 3, 5, 6, 7

[54] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2018. 1, 2, 4, 7, 8

[55] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dual grid net: Hand mesh vertex regression from single depth maps. In *European Conference on Computer Vision*, pages 442–459. Springer, 2020. 1, 2, 3, 5, 7

[56] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304, 2021. 2

[57] Jiayi Wang, Franziska Mueller, Florian Bernard, and Christian Theobalt. Generative model-based loss to the rescue: A method to overcome annotation errors for depth-based hand pose estimation. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 93–100, 2020. 1

[58] Rongchang Xie, Chunyu Wang, and Yizhou Wang. Metafuse: A pre-trained fusion model for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13686–13695, 2020. 1, 3

[59] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 793–802, 2019. 1, 2, 7, 8

[60] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16105–16114, 2021. 4

[61] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7444–7452, 2018. 3

[62] Linlin Yang, Shicheng Chen, and Angela Yao. Semihand: Semi-supervised hand pose estimation with consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11364–11373, 2021. 2

[63] Mao Ye and Ruigang Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2352, 2014. 2

[64] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, July 2017. 1, 6

[65] Yuxiang Zhang, Liang An, Tao Yu, Xiu Li, Kun Li, and Yebin Liu. 4d association graph for realtime multi-person motion capture using multiple video cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1324–1333, 2020. 3

[66] Yachao Zhang, Yanyun Qu, Yuan Xie, Zonghao Li, Shanshan Zheng, and Cuihua Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15520–15528, 2021. 2

[67] Zhe Zhang, Chunyu Wang, Weichao Qiu, Wenhu Qin, and Wenjun Zeng. Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild. *International Journal of Computer Vision*, 129(3):703–718, 2021. 1, 3

[68] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019. 3

[69] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. Model-based deep hand pose estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2421–2427, 2016. 2, 7, 8

[70] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 5

[71] Zhiming Zou, Kenkun Liu, Le Wang, and Wei Tang. High-order graph convolutional networks for 3d human pose estimation. In *BMVC*, 2020. 3