Two Heads Are Better than One: Image-Point Cloud Network for Depth-Based 3D Hand Pose Estimation

Pengfei Ren, Yuchen Chen, Jiachang Hao, Haifeng Sun, Qi Qi*, Jingyu Wang*, Jianxin Liao

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
{rpf, cyc99, haojc, hfsun, qiqi8266, wangjingyu, liaojx}@bupt.edu.cn

Abstract

Depth images and point clouds are the two most commonly used data representations for depth-based 3D hand pose estimation. Benefiting from the structuring of image data and the inherent inductive biases of the 2D Convolutional Neural Network (CNN), image-based methods are highly efficient and effective. However, treating the depth data as a 2D image inevitably ignores the 3D nature of depth data. Point cloud-based methods can better mine the 3D geometric structure of depth data. However, these methods suffer from the disorder and non-structure of point cloud data, which is computationally inefficient. In this paper, we propose an Image-Point cloud Network (IPNet) for accurate and robust 3D hand pose estimation. IPNet utilizes 2D CNN to extract visual representations in 2D image space and performs iterative correction in 3D point cloud space to exploit the 3D geometry information of depth data. In particular, we propose a sparse anchor-based "aggregation-interaction-propagation" paradigm to enhance point cloud features and refine the hand pose, which reduces irregular data access. Furthermore, we introduce a 3D hand model to the iterative correction process, which significantly improves the robustness of IPNet to occlusion and depth holes. Experiments show that IPNet outperforms state-of-the-art methods on three challenging hand datasets.

Introduction

Hands are essential for humans to interact with the world and convey intent. 3D hand pose estimation is the core technology of human-computer interaction, virtual reality and augmented reality. Although great progress has been made in depth-based 3D hand pose estimation, there are still some challenging scenarios that are difficult to solve, mainly due to severe occlusions, including self-occlusion and object-occlusion, and self-similarity between fingers.

Depth-based 3D hand pose estimation can be broadly categorized into two classes according to the input data: 2D image-based methods and 3D data based-methods. The 2D image-based method (Oberweger and Lepetit 2017; Wan et al. 2018; Xiong et al. 2019; Ren et al. 2019; Huang et al. 2020b) treats the depth data as a single-channel depth image, and then uses a 2D Convolutional Neural Network (CNN) for 3D hand pose estimation. With well-explored network structures and inherent inductive biases, 2D CNNs has advantages in extracting local visual features. Thank to the regular structure of image and highly parallelized data processing mechanism (sliding window), 2D CNN is computationally efficient on modern hardware. 3D data based-methods convert depth data into 3D point cloud (Ge et al. 2018a; Ge, Ren, and Yuan 2018a; Chen et al. 2018, 2019; Huang et al. 2020a) or volumetric representation (Ge et al. 2018b; Moon, Chang, and Lee 2018; Malik et al. 2021), and use the point cloud network or 3D CNN for pose estimation. These methods can avoid information loss due to projection and better perceive the 3D geometric structure of depth data.

Although both types of methods have made remarkable progress in 3D hand pose estimation, they both have their shortcomings. For 2D image-based methods, the learning process of visual representations ignores the 3D characteristics of depth data, and is limited by the local receptive field mechanism of convolution operation, which makes it challenging to capture long-range dependency. For point-based methods, due to the disordered, unstructured and non-uniform nature of point clouds, the popular point cloud network requires densely and dynamically building local neighborhoods and employing sophisticated feature extractors (Qi et al. 2017; Wang et al. 2019; Thomas et al. 2019; Li et al. 2018). Due to irregular memory access and dynamic kernel overhead, point cloud networks are computationally expensive. In addition, these methods often require additional pre- or post-processing to assist the learning of point cloud features. For 3D voxel-based methods, the 3D convolution operator is more inefficient and the 3D voxelization requires a large amount of memory to represent the input data and the intermediate features, making it memory-prohibitive to scale up the input resolution and network structure.

In previous methods, depth data is independently regarded as the 2D image or 3D point cloud, which either ignore the 3D geometric structure of depth data or consumes huge computation to extract point-wise features. In this paper, we propose an Image-Point cloud Network (IPNet) that combines the two data representations to take advantage of their unique properties. First, our method represents the depth data as a 2D image and adopts the 2D CNN for visual representation learning and initial pose estimation. Then, we project the 2D image features into the point cloud space and perform iterative feature enhancement and pose correction

*Corresponding author.

in the 3D space, which can fully exploit the geometric structure information of the depth data. In particular, we abandon the dense local feature extraction and hierarchical network structure in popular point cloud networks (Qi et al. 2017; Thomas et al. 2019). Instead, we propose an "aggregation-interaction-propagation" paradigm based on sparse anchors. Specifically, we use hand joints as anchors to aggregate the 3D spatial information of local neighborhoods, and then use Graph Convolution Network (GCN) to perform information interaction between anchors to capture long-range dependencies between different hand regions. Finally, we propagate the anchor information back to update the point cloud features and perform point-wise hand pose estimation.

Furthermore, to improve the robustness of IPNet to severe occlusion and depth holes, we introduce a 3D hand model during iterative refinement. Specifically, in each refinement stage, we additionally estimate a 3D hand model (Romero, Tzionas, and Black 2017). Then, we construct the corresponding semantic features of the point cloud sampled from the model surface through a 3D-aware re-parameterization. The point cloud information from the hand model can provide strong disambiguation cues for subsequent refinement stages, especially for the hand regions lacking depth information due to occlusions and depth holes. Code is available at https://github.com/PengfeiRen96/IPNet.

We conduct experiments on three challenging hand datasets (NYU (Tompson et al. 2014), HO-3D (Hampali et al. 2020), and DexYCB (Chao et al. 2021)). Experiments show that IPNet outperforms State-Of-The-Art (SOTA) methods, especially for hand-object interaction scenarios.

Our contributions can be summarized as follows:

- We propose a hybrid network architecture (IPNet) that utilizes both the 2D depth image and the 3D point cloud for robust and accurate 3D hand pose estimation.
- We propose a sparse anchor-based iterative correction paradigm, which can effectively and efficiently exploit the 3D geometric structure information of depth data.
- IPNet achieves SOTA performance on three challenging datasets, outperforming previous methods on hand-object interaction scenarios by a large margin.

Related Work

Depth-based 3D Hand Pose Estimation

In recent years, advances in deep neural networks have facilitated the development of depth-based 3D hand pose estimation. Prior arts can be roughly divided into two categories, 2D image-based method (Wan et al. 2018; Xiong et al. 2019; Oberweger and Lepetit 2017; Ren et al. 2019; Huang et al. 2020b; Fang et al. 2020) and 3D data-based method (Ge et al. 2018a; Ge, Ren, and Yuan 2018a; Ge et al. 2018b; Moon, Chang, and Lee 2018; Malik et al. 2020; Huang et al. 2020a). 2D image-based methods ignore the 3D geometric properties of depth data. Thus, some methods propose to mine the 3D spatial information of the depth data by estimating 3D-aware representations (Huang et al. 2020b; Wan et al. 2018) or performing multi-view prediction (Ge et al. 2016; Cheng et al. 2022). However, these methods only incorporate 3D information in the network prediction phase and ignore the 3D structural information of the depth data in the process of visual feature learning.

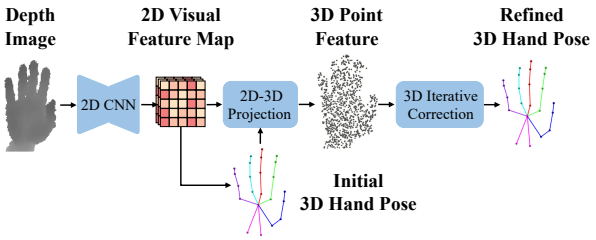

Figure 1: Overview. IPNet utilizes a 2D CNN for visual feature extraction and initial hand pose estimation. Then, IPNet obtains the initial point cloud features through a 2D-3D projection module. Finally, IPNet iteratively updates point features and refines hand pose in the 3D point cloud space.

3D data-based methods usually represent depth data as 3D voxel or 3D point cloud. However, 3D voxel-based methods (Moon, Chang, and Lee 2018; Ge et al. 2018b) use 3D CNNs to regress 3D hand pose from 3D volumes, which is computationally inefficient. 3D point cloud, as a natural and compact representation of depth data, has attracted extensive attention. However, point cloud networks (Qi et al. 2017; Wang et al. 2019) rely on dense and sophisticated local geometric extractors and a hierarchical feature learning paradigm, which suffer from prohibitive computations and the overhead of memory access. In addition, it is difficult to improve the performance of point cloud networks by increasing the depth or width of the network (Ma et al. 2022). Therefore, previous methods have to adopt complex pre-processing such as surface normal estimation (Ge et al. 2018a) and oriented bounding box calculation (Ge, Ren, and Yuan 2018a), auxiliary tasks (Chen et al. 2018) or post-processing such as fingertip refinement (Ge et al. 2018a) to improve the pose estimation accuracy, which further reduces the inference speed of the point cloud-based method.

Iterative Correction Mechanism

Iterative correction mechanisms have been extensively explored in hand and body pose estimation tasks. Some works propose to use the estimated pose to extract local depth patches (Oberweger, Wohlhart, and Lepetit 2015) or feature patches (Chen et al. 2020) in order to regress the refined pose. However, these methods ignore the 3D properties of depth data and are prone to fall into local optima. Some methods (Ren et al. 2019, 2021) re-parameterize the estimated pose as 3D-aware representations and feed it to a subsequent corrector. However, these methods use the 2D CNN as the corrector and perform pose refinement in 2D feature space, which still cannot fully utilize the 3D spatial information of input depth data. Some human body mesh estimation methods propose to project the estimated 3D model information back to the 1D global feature space (Kanazawa et al. 2018) or the 2D visual feature space (Zhang et al. 2021) for iterative correction. Our method performs refinement correction directly in the 3D point cloud space, which can better utilize the spatial structure information of the estimated

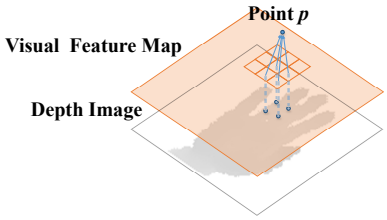

Figure 2: Illustration of the projection from 2D visual features to 3D point cloud features.

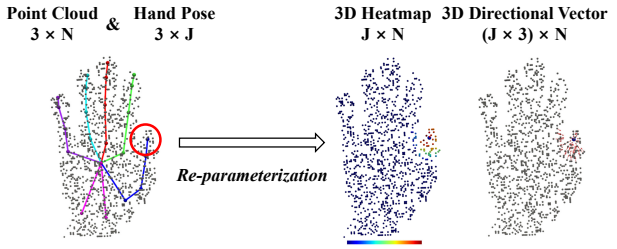

Figure 3: Pose re-parameterization. We only visualize the 3D heatmap and the vector field of one joint. The color bar represents the relationship between 3D distance and color.

3D hand pose and the 3D hand mesh.

Method

Overview

As shown in Fig. 1, IPNet mainly consists of two parts, the first part performs 2D visual feature learning and initial pose estimation based on 2D images, and the second part performs 3D geometric feature learning and pose refinement based on 3D point clouds. In the 2D image space, we utilize a fully convolutional network to extract semantic visual features and perform pixel-wise pose regression. In the 3D point cloud space, we sparsely construct local neighborhoods and extract 3D geometric structure information, which can iteratively refine the initial hand pose.

Visual Feature and Pose Estimation in 2D Space

Depth data can be represented as a 2D image, where each pixel value represents the distance from the image plane to the object surface. Similar to (Huang et al. 2020b; Wan et al. 2018), we adopt an encoder-decoder architecture to keep the 2D spatial structure of visual features and better exploit local evidence. We estimate three pixel-level pose representations from the 2D visual feature $\mathbf{F}^{2d} \in \mathbb{R}^{C\times H\times W}$, including a 3D heatmap $\mathbf{H}^{pixel} \in \mathbb{R}^{J\times H\times W}$, a 3D directional vector field $\mathbf{D}^{pixel} \in \mathbb{R}^{(3\times J)\times H\times W}$, and a weight map $\mathbf{W}^{pixel} \in \mathbb{R}^{J\times H\times W}$. Here, C, H, W, J represent the number of channels, the height and the width of the feature map, and the number of joints. $\mathbf{H}^{pixel}$ and $\mathbf{D}^{pixel}$ represent 3D Euclidean distance and 3D unit direction from each pixel to target joint. $\mathbf{W}^{pixel}$ represents the importance of each pixel to the target joint. Based on these three pixel-level representations, we can obtain an initial 3D hand pose $\mathbf{J}^{init} \in \mathbb{R}^{3\times J}$ by a weighted average algorithm (Ren et al. 2022).

2D-3D Projection

Given the 3D point cloud $\mathbf{P} \in \mathbb{R}^{3\times N}$ of the input depth data, the 2D visual feature map $\mathbf{F}^{2d}$ and the initial 3D hand pose $\mathbf{J}^{init}$, we construct the initial point cloud feature $\mathbf{F}^{3d} \in \mathbb{R}^{C\times N}$ through a 2D-3D projection module. The 3D point cloud feature $\mathbf{F}^{3d}$ consists of three features, including the projected 2D visual feature, the re-parameterized hand pose information and the point position information. As shown in Fig. 2, in order to obtain projected feature $\mathbf{F}_p^{proj} \in \mathbb{R}^C$ of the point p, we select K closest elements to p from the 2D visual feature map $\mathbf{F}^{2d}$ and perform interpolation according to their 3D Euclidean distance $\mathbf{d}$ to the point p:

$$\mathbf{F}_p^{proj} = \frac{\sum_{k=1}^{K} \mathbf{d}_{p,k} \cdot \mathbf{F}_k^{2d}}{\sum_{k=1}^{K} \mathbf{d}_{p,k}}, \tag{1}$$

where we set K to 4 by default. We downsample the depth image to the same size as the visual feature map and calculate the 3D coordinates of each pixel in 3D point cloud space through the camera intrinsics, which is regarded as the coordinates of each 2D visual feature element in the 3D space.

As shown in Fig. 3, according to the initial hand pose $\mathbf{J}^{init}$, we re-parameterize the coordinates of each 3D point as the 3D heatmap $\mathbf{H}^{point} \in \mathbb{R}^{J\times N}$ and 3D unit direction vector $\mathbf{D}^{point} \in \mathbb{R}^{(3\times J)\times N}$ to each joint j as follows:

$$\mathbf{H}_j^{point}(p) = \begin{cases} 1 - \frac{\left\|\mathbf{P}_p - \mathbf{J}_j^{init}\right\|_2}{r} & \left\|\mathbf{P}_p - \mathbf{J}_j^{init}\right\|_2 \le r, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$$\mathbf{D}_j^{point}(p) = \begin{cases} \frac{\mathbf{P}_p - \mathbf{J}_j^{init}}{\left\|\mathbf{P}_p - \mathbf{J}_j^{init}\right\|_2} & \left\|\mathbf{P}_p - \mathbf{J}_j^{init}\right\|_2 \le r, \\ 0 & \text{otherwise.} \end{cases}, \tag{3}$$

where r represents the maximum perceived distance of each point. Since different point features (i.e., 3D coordinate, projected visual feature, and re-parameterized hand pose) have different properties and distributions, we equally fuse them through the channel de-differentiation (Ran, Liu, and Wang 2022), which improves training stability. Each type of feature is passed through an independent normalization layer:

$$\begin{aligned} \mathbf{F}^{3d} = ReLU(&BN\left(\mathbf{W}_0\mathbf{P}\right) + BN\left(\mathbf{W}_1\mathbf{F}^{proj}\right) + \\ &BN\left(\mathbf{W}_2\left[\mathbf{H}^{point}; \mathbf{D}^{point}\right]\right)), \end{aligned} \tag{4}$$

where $\mathbf{W}_0$, $\mathbf{W}_1$ and $\mathbf{W}_2$ are three learnable parameter matrices for point feature embedding; $ReLU$ and BN represent ReLU activation function and batch normalization layer (Ioffe and Szegedy 2015), respectively. Based on the point cloud features and the estimated joint coordinates, we aggregate the initial anchor features $\mathbf{F}^{anchor}$ according to a point-wise weight map $\mathbf{W}^a$ as follow:

$$\mathbf{F}^{anchor} = ReLU\left(BN\left(\mathbf{W}_3\mathbf{J}^{init}\right) + BN\left(\mathbf{W}_4\mathbf{F}^{3d}\mathbf{W}^a\right)\right), \tag{5}$$

where $\mathbf{W}_3$ and $\mathbf{W}_4$ are two learnable parameter matrices for anchor feature embedding. The point-wise weight map $\mathbf{W}^a$ can be obtained from the pixel-wise weight map $\mathbf{W}^{pixel}$ by a similar interpolation process as $\mathbf{F}^{proj}$.

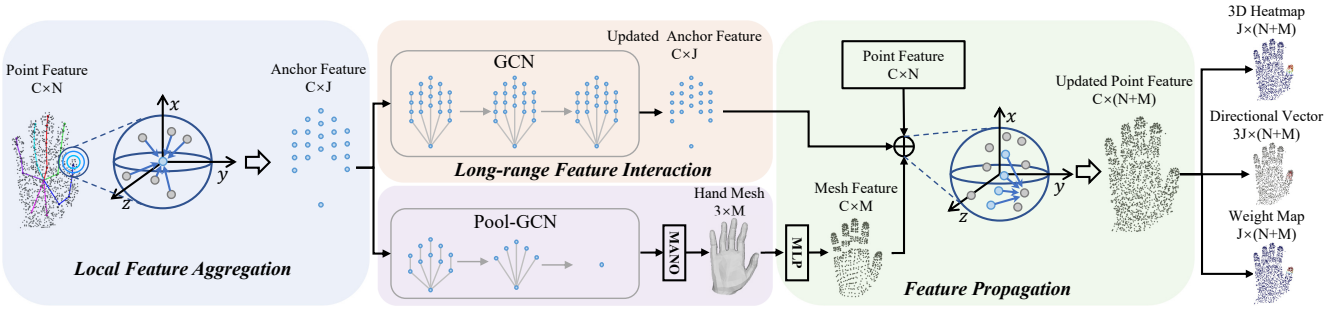

Figure 4: Illustration of the "aggregation-interaction-propagation" paradigm. The blue dots represent anchors and the grey dots represent the point cloud. The gray lines between the anchors represent the bone structure of the hand. For illustration purposes, we only show local feature aggregation and feature propagation in a single neighborhood.

Iterative Correction in 3D Space

3D geometric information in depth data is important for accurate and robust 3D hand pose estimation. Point cloud, as a natural representation of depth data, can avoid the information loss caused by projection or voxelization. However, popular point cloud networks (Qi et al. 2017; Thomas et al. 2019; Li et al. 2018) densely construct local neighborhoods by the farthest point sampling algorithm and perform sophisticated local feature extraction, greatly increasing the inference time of the network. Meanwhile, these methods adopt a hierarchical network structure to gradually increase the receptive field and capture the long-range dependencies between different point cloud regions, which also increases the computational complexity. To address these problems, we propose an anchor-based "aggregation-interaction-propagation" paradigm to efficiently extract 3D geometric structure information in 3D point cloud space.

Benefiting from the strong representation of the initial point cloud features $\mathbf{F}^{3d}$, our method neither needs to perform local feature extraction densely nor employ time-consuming pre-processing. Instead, our method focuses on capturing fine-grained geometric features of key regions of the 3D point cloud. As shown in Fig. 4, we use the estimated hand joints as anchors to build local neighborhoods and aggregate features, which greatly reduce irregular memory accesses and feature extraction operations. First, we obtain the K neighbors of each anchor by the ball query. In practice, we set K to 64 by default. Then, these neighbor features are transformed through a series of Multi-Layer Perceptrons (MLPs). A reduction layer (max-pooling) is used to aggregate local features. For the i-th anchor, the local operation can be formulated as follows:

$$\mathbf{F}_i^{anchor} = MAX\left(\Phi_{cd}\left(\mathbf{F}_{i,j}, \mathbf{P}_{i,j}\right), |j = 1, \cdots, K)\right), \tag{6}$$

where Φ_{cd} denotes the feature mapping function with channel de-differentiation; $\mathbf{F}_{i,j}$ and $\mathbf{P}_{i,j}$ are the j-th neighbor relative point feature $(\mathbf{F}_i - \mathbf{F}_j)$ and relative position $(\mathbf{P}_i - \mathbf{P}_j)$ of the i-th anchor. The Φ_{cd} consists of multiple Fully-Connected (FC) layer, batch normalization layer, and activation layer. In particular, the first layer of the feature mapping function Φ_{cd} adopts the channel de-differentiation, which can be formulated as:

$$\mathbf{F}_i^{anchor,1} = ReLU\left(BN\left(\mathbf{W}_p\mathbf{P}_{i,j}\right) + BN\left(\mathbf{W}_f\mathbf{F}_{i,j}\right)\right) \tag{7}$$

Similar to previous work (Qi et al. 2017), we adopted a multi-scale aggregation approach. Specifically, for each anchor, we perform the local feature aggregation in the range of 0.1, 0.2 and 0.4, and then the multi-scale features are concatenated together and embedded as anchor features.

Inspired by CNN, point cloud networks usually adopt a hierarchical network structure to gradually increase the receptive field and model long-range dependencies. However, the hierarchical structure significantly increases the computational burden of the network and reduces the inference speed of the network. In our method, according to the bone structure of the hand, we utilize the SemGCN (Zhao et al. 2019) to perform information passing between anchors in order to model long-range relationships. Then, similar to (Qi et al. 2017), for each point, we select the nearest K anchors to obtain the interpolated features through inverse distance weighted average, where we set K to 4 by default. The interpolated features and the original features are concatenated together and passed through a point-wise MLP to obtain the updated point cloud features. Finally, we regress three point-wise representations from the updated point feature, including a 3D heatmap $\mathbf{H}^{point} \in \mathbb{R}^{J\times N}$, a 3D unit directional vector field $\mathbf{D}^{point} \in \mathbb{R}^{(3\times J)\times N}$, and a weight field $\mathbf{W}^{point} \in \mathbb{R}^{J\times N}$, from which we can obtain the refined 3D hand pose the weighted regression.

Incorporating 3D Hand Model

Self-occlusion and object occlusion is one of the most challenging problems for 3D hand pose estimation. Even if we fully exploit the local geometry information of the depth data and perform long-range information interaction, it is difficult to deal with some severely occluded samples. Therefore, we introduce a hand model in the iterative correction, which provide prior structural information of the hand and strong disambiguation cues for the subsequent network. As shown in Fig. 4, we adopt a hierarchical GCN (Pool-GCN) to progressively aggregate the joint features according to the hand structure by average pooling and use graph convolution to perform information interaction on down-

sampled joints at each level. Then, we regress the parameters of a parameterized hand model (Romero, Tzionas, and Black 2017) through a FC layer. We convert the M vertices of the estimated hand mesh $\mathbf{P}^{mesh} \in \mathbb{R}^{3\times M}$ to a 3D heatmap $\mathbf{H}^{mesh} \in \mathbb{R}^{J\times M}$ and a unit direction vector field $\mathbf{D}^{mesh} \in \mathbb{R}^{(J\times 3)\times M}$ via pose re-parameterization. Finally, $\mathbf{P}^{mesh}$, $\mathbf{H}^{mesh}$ and $\mathbf{D}^{mesh}$ are embedded as vertex features $\mathbf{F}^{mesh} \in \mathbb{R}^{J\times M}$ by a point-wise MLP. The vertex features and point cloud features will be concatenated for hand pose regression and subsequent iterative correction. Hand model information can make up for the lack of point cloud information due to occlusion and depth holes, which significantly improves the robustness of our method.

Experiment

Dataset and Evaluation Metrics

Single-Hand dataset **NYU** (Tompson et al. 2014) contains 72,757 training images and 8,252 testing images. It is a challenging dataset with a wide coverage of hand poses and image noise. This dataset provides 3D annotations for 36 joints. Following previous works (Oberweger and Lepetit 2017; Moon, Chang, and Lee 2018; Wan et al. 2018), we selected 14 joints from all annotated joints. In particular, since NYU dataset does not provide annotations for MANO, we use an iterative optimization method (Armagan et al. 2020) to obtain the corresponding MANO parameters from the joint annotations. It is worth mentioning that, as mentioned in many recent works (Oberweger et al. 2016; Ren et al. 2022), many images in ICVL (Tang et al. 2014), MSRA (Sun et al. 2015), and BigHand2.2M (Yuan et al. 2017) suffer from severe annotation errors. Thus, we do not evaluate our method on these datasets.

Hand-Object dataset **DexYCB** (Chao et al. 2021) is a recent real hand-object dataset captured by multiple RGB-D cameras. It consists of 582,000 image frames with 10 different subjects and 20 YCB objects from 8 views. This dataset has four official dataset split settings, namely S0, S1, S2 and S3. S0, S1, S2, S3 represent the dataset is split by the sequences, subjects, views and objects, respectively. **HO-3D** (Hampali et al. 2020) is another challenging dataset that contains precise hand-object pose during the interaction. The most commonly used version of HO-3D is the **HO-3D_v2**. It consists of 66,034 training images and 11,524 testing images from 10 subjects with 10 different objects. Recently, **HO-3D_v3** (Hampali, Sarkar, and Lepetit 2021) is released, which has more accurate annotations and more data including 83,325 training images and 20,137 testing images. Evaluation for HO-3D_v2 and HO-3D_v3 are performed through an online submission website.

We evaluate our method using two widely used metrics: Mean Per Joint Position Error (MPJPE) and Percentage of Successful Frames (PSF). MPJPE is the mean 3D Euclidean distance between the predicted coordinates and the ground-truth coordinates for each joint. PSF is defined as the proportion of good frames in all testing frames. If the maximum value of the joints error in a frame is less than a certain threshold, it will be judged as a good frame.

Method	Input	MPJPE					IF
		S0	S1	S2	S3	AVG	
ResNet-18	Depth	11.20	12.03	11.02	11.70	11.48	4.3
ConvNeXt-T	Depth	10.57	11.32	10.29	11.11	10.82	6.4
ConvNeXt-S	Depth	10.56	11.30	10.05	11.06	10.74	8.5
ConvNeXt-B	Depth	10.67	11.31	10.10	11.03	10.78	8.6
PointNet++	Point	10.02	10.93	11.68	9.78	10.60	9.5
PointMLP	Point	9.66	10.83	10.89	9.60	10.25	10.2
PointMLP*	Point	9.78	11.01	11.19	9.75	10.43	13.7
IPNet-1Stage	Depth &Point	8.43	9.43	9.34	8.48	8.92	10.6

Table 1: Comparisons between different data representations with different backbone networks. We report the MPJPE (mm) on DexYCB dataset. The brackets under our method indicate that we adopt ConvNeXt-T to extract 2D visual features. IF represents the inference time (ms). $*$ represents increasing the number of layers and channels of the network.

Implementation Details

We train and evaluate our method with an NVIDIA RTX 3090 GPU. The network is implemented within PyTorch and trained using AdamW with an initial learning rate of 0.001. For NYU and HO-3D, the learning rate is divided by 10 at 25-th epochs and the training stops at 30-th epochs. For DexYCB, the learning rate is divided by 10 at 10-th epochs and the training stops at 15-th epochs. We crop the input depth image to 128×128 and sample the number of point clouds to 1024. The values of the depth image and the 3D point cloud are normalized to [-1, 1]. We perform data augmentation including rotation ([-180, 180]), random scaling ([0.9, 1.1]) and random translation ([-10, 10]). More details about network structure and training are provided in the supplementary material.

Ablation Study

Considering that DexYCB has a large amount of data, diverse subjects and objects, and multiple splits, we conduct ablation experiments on DexYCB.

Comparing with Different Data Representation For image-based pixel-wise regression, we adopt the most frequently used ResNet-18 (He et al. 2016) and the latest proposed ConNeXt (Liu et al. 2022) as the backbone. For point-based point-wise regression, we adopt the most representative network structures, PointNet++ (Qi et al. 2017), and the latest method, PointMLP (Ma et al. 2022), as the backbone. In particular, for the fairness of the comparison, the IPNet only adopts a single correction stage and does not use the 3D hand model. As shown in Table 1, we mainly get the following conclusions: **1)** Methods based on different data representations exhibit different properties. Point cloud-based methods are better at predicting examples with unseen objects (S3), and image-based methods are better at handling examples with unseen viewpoints (S2). **2)** Adopting the well-designed network structure can improve network performance. However, on this basis, simply increasing the width of the network or the number of layers yields a little of gains. **3)** Using both depth images and point clouds

ID	PE	HE	FE	LA	GI	MC	HM	MPJPE (mm)				
								S0	S1	S2	S3	AVG
1	✓							8.96	9.95	9.79	8.91	9.40
2	✓	✓		✓				8.62	9.63	9.61	8.61	9.12
3	✓		✓					8.50	9.55	9.46	8.81	9.08
4	✓	✓	✓					8.43	9.43	9.34	8.48	8.92
5		✓			✓			8.75	9.55	9.39	8.90	9.14
6				✓				8.67	9.50	9.35	8.65	9.04
7		✓		✓		✓		8.23	9.25	9.05	8.27	8.71
8		✓		✓		✓	✓	8.03	9.01	8.60	7.80	8.36

Table 2: We report the MPJPE on DexYCB dataset. 'PE', 'HE', and 'FE' represent the point position information, hand pose information and 2D visual features used in the initial point cloud feature embedding, respectively. 'LA' represents adopting the local feature aggregation. 'GI' represents performing GCN-based long-range interaction. 'MC' stands for multi-stage correction. 'HM' stands for the hand model.

together performs significantly better than using the images or point clouds alone. In particular, the framework of pixel-wise regression (Huang et al. 2020b) and point-wise regression (Ge, Ren, and Yuan 2018a) is already the previous SOTA methods and we further adopt the latest backbone with stronger capabilities. Nonetheless, IPNet still surpasses them by a large margin with comparable inference speed.

Effect of 2D-3D Projection 2D CNN provides 2D visual features and 3D hand pose information for the initial point cloud features, which is important to avoid adopting dense local feature extraction and complex pre-processing. As shown in Table 2, adopting the hand pose information (ID 2) or the projected 2D visual features (ID 3) alone can significantly improve the performance of the IPNet. Furthermore, when all the information is used together (ID 4), the performance of the network can be further improved. In the subsequent experiments, we use all the information by default to construct the initial point cloud features.

Effect of Components in Iterative Correction We first verified the role of the local feature aggregation and the graph-based long-range information interaction. As shown in Table 2, discarding local feature aggregation (ID 5) or long-range information interaction (ID 6) can lead to a decrease in network performance, which illustrates the importance of explicitly learning local geometric information and long-range dependencies. Next, we explore the effect of the number of correction stages. As shown in Table 2, from single-stage (ID 4) to three-stage (ID 7), the average MPJPE decreased by 0.21 mm. Finally, we introduce hand model information (ID 8). Even though the network with three-stage correction already has very superior performance, the prior information of the hand model can improve the robustness of the network to occlusion and depth holes, so it can still significantly improve the performance of the network. IPNet with three-stage runs in real-time at about 30.9 FPS.

Method	Input	MPJPE
DeepPrior++ (Oberweger et al. 2017)	Depth	12.24
Pose-REN (Chen et al. 2020)	Depth	11.81
DenseReg (Wan et al. 2018)	Depth	10.21
SRN (Ren et al. 2019)	Depth	7.79
A2J (Xiong et al. 2019)	Depth	8.61
JGR-P2O (Fang et al. 2020)	Depth	8.29
AWR (Huang et al. 2020b)	Depth	7.48
3DCNN (Ge et al. 2018b)	Voxel	10.02
V2V-PoseNet (Moon et al. 2018)	Voxel	8.41
HandVoxNet (Malik et al. 2020)	Voxel	8.72
SO-HandNet (Chen et al. 2019)	Point	11.20
HandPointNet (Ge et al. 2018a)	Point	10.54
P2P (Ge, Ren, and Yuan 2018b)	Point	9.04
PEL (Li and Lee 2019)	Point	8.35
NARHT (Huang et al. 2020a)	Point	9.80
IPNet	Depth &Point	**7.17**

Table 3: Comparison with SOTA methods of the MPJPE (mm) on NYU (Tompson et al. 2014) dataset.

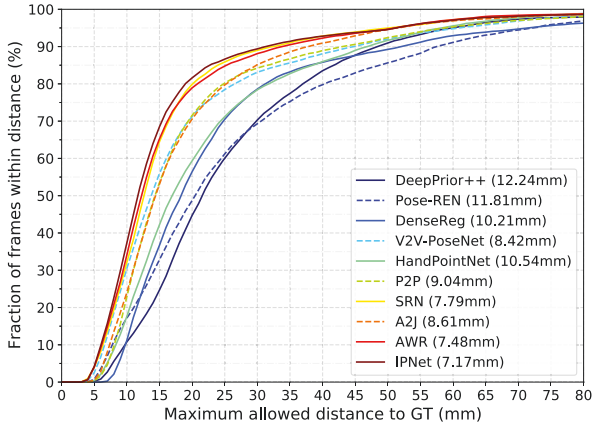

Figure 5: Comparison with SOTA methods of the PCK under different thresholds on NYU (Tompson et al. 2014) dataset.

Comparisons with State-of-the-arts

On NYU dataset, We compare our approach with SOTA depth-based methods. Since some methods do not provide result files, we cannot draw them in Fig. 5. As shown in Table 3 and Fig. 5, our method achieves the lowest MPJPE and has the best PCK at almost all thresholds. Under a similar element-wise regression framework, IPNet achieves the smallest MPJPE compared to adopting depth images alone (DenseReg (Wan et al. 2018), AWR (Huang et al. 2020b) and A2J (Xiong et al. 2019)) or point clouds alone (P2P (Ge, Ren, and Yuan 2018b) and PEL (Li and Lee 2019)).

On HO3D dataset, some methods only report the MPJPE after alignment (rotation, scaling and translation are performed according to the annotation information), which does not meet the requirements of some practical application scenarios such as augmented reality (Tang, Wang, and Fu 2021), and may ignore some model defects. Therefore, here we focus on MPJPE and compare with those methods reporting MPJPE. More other metrics are included in our

Method	Input	HO3D_v2	HO3D_v3
Hybrik (Li et al. 2021)	RGB	2.89	-
ArtiBoost (Yang et al. 2022)	RGB	2.53	2.34
HandOccNet (Park et al. 2022)	RGB	2.49	-
HandVoxNet++ (Malik et al. 2021)	Voxel	2.46	-
IPNet	Depth &Point	**1.81**	**1.93**

Table 4: The comparison with SOTA methods of the MPJPE (cm) on HO3D_v2 and HO3D_v3 dataset.

supplementary material. As shown in Table 4, IPNet outperforms the previous RGB-based methods by a large margin on both HO3D_v2 and HO3D_v3, which illustrates the importance of depth information for 3D hand pose estimation. Furthermore, IPNet also outperforms the depth-based method, which shows that fully exploiting the 3D spatial information in depth data is also important.

Method	Input	MPJPE				
		S0	S1	S2	S3	AVG
A2J	Depth	23.93	25.57	27.65	24.92	25.52
Spurr et al.	RGB	17.34	22.26	25.49	18.44	18.44
METRO	RGB	15.24	-	-	-	-
Tse et al.	RGB	16.05	21.22	27.01	17.93	20.55
HandOccNet	RGB	14.04	-	-	-	-
IPNet	Depth &Point	**8.03**	**9.01**	**8.60**	**7.80**	**8.36**

Table 5: The comparison with SOTA methods of the MPJPE (mm) on DexYCB (Chao et al. 2021) dataset.

On the DexYCB dataset, since some methods are evaluated on their own setting, we only compare those methods that are experimented on the official setting, including A2J (Xiong et al. 2019), Spurr et al. (Spurr et al. 2020), METRO (Lin, Wang, and Liu 2021), Tse et al. (Tse et al. 2022) and HandOccNet (Park et al. 2022). As shown in Table 5, IPNet greatly improves the performance on the four setting compared to previous SOTA methods, which shows the importance of depth information. IPNet outperforms previous SOTAs in both the single-hand scenario and the more challenging hand-object interaction scenario.

Qualitative Results

Some qualitative results for NYU, HO3D and DexYCB datasets are shown in Fig. 6. First, for the fine-grained pose (1-th row) and the pose that suffer from finger self-similarity (2-th row), IPNet shows a stronger perception of 3D spatial information and can more accurately predict the position of the fingertip. Second, for some complex poses with serious self-occlusion (3-th row and 4-th row), IPNet is able to deal with uncertainty caused by self-occlusion well, which obtains more accurate and reasonable poses than other SOTA methods. Meanwhile, our method is also very robust to the occlusion caused by objects. First, object occlusion may cause the collapse of the whole or part of the hand pose. Our method can produce more reasonable hand pose through the

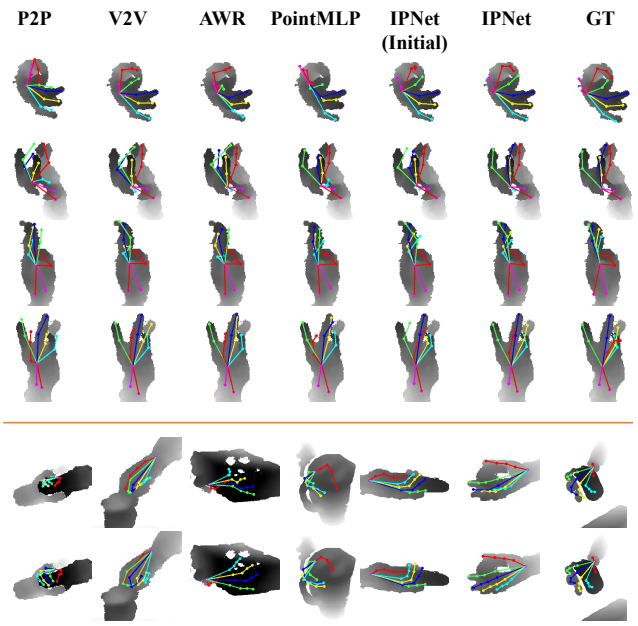

Figure 6: Top: Qualitative results for NYU dataset. Each column shows the estimated hand pose for one method. Bottom: Qualitative results for HO3D and DexYCB datasets. The first row represents the initial hand pose of IPNet, and the second row represents the corrected pose of IPNet.

iterative correction (the first four columns). Second, the estimated hand pose may be offset due to the interference of the object, and IPNet can correct it by exploring local geometric information (the last three columns).

Conclusion

In this paper, we propose the IPNet for efficient and robust 3D hand pose estimation. The key insight is to combine the complementary advantages of depth image and 3D point cloud. We adopt 2D CNN to efficiently extract visual representations and perform initial pose estimation in image space. Then, we utilize projected visual features and estimated pose to construct the initial point cloud features and perform iterative correction in 3D point space. In particular, we propose a sparse anchor-based "aggregation-interaction-propagation" paradigm to exploit the 3D geometry structure of depth data, which significantly reduces irregular data access. Furthermore, we introduce a 3D hand model to the iterative correction in order to improve the robustness of IPNet to occlusion and depth holes. Experiments on the three challenging hand pose datasets demonstrate the effectiveness of the IPNet, where IPNet outperforms the previous methods by a large margin in the hand-object interaction scenario.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants (62071067, 62171057, 62201072, 62101064, 62001054), China Postdoctoral Science Foundation under Grant

2022M710468, in part by the Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Innovation Center, in part by BUPT innovation and entrepreneurship support program.

References

Armagan, A.; Garcia-Hernando, G.; Baek, S.; Hampali, S.; Rad, M.; Zhang, Z.; Xie, S.; Chen, M.; Zhang, B.; Xiong, F.; et al. 2020. Measuring Generalisation to Unseen Viewpoints, Articulations, Shapes and Objects for 3D Hand Pose Estimation under Hand-Object Interaction. *arXiv preprint arXiv:2003.13764*.

Chao, Y.-W.; Yang, W.; Xiang, Y.; Molchanov, P.; Handa, A.; Tremblay, J.; Narang, Y. S.; Van Wyk, K.; Iqbal, U.; Birchfield, S.; et al. 2021. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9044–9053.

Chen, X.; Wang, G.; Guo, H.; and Zhang, C. 2020. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 395: 138–149.

Chen, X.; Wang, G.; Zhang, C.; Kim, T.-K.; and Ji, X. 2018. Shpr-net: Deep semantic hand pose regression from point clouds. *IEEE Access*, 6: 43425–43439.

Chen, Y.; Tu, Z.; Ge, L.; Zhang, D.; Chen, R.; and Yuan, J. 2019. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 6961–6970.

Cheng, J.; Wan, Y.; Zuo, D.; Ma, C.; Gu, J.; Tan, P.; Wang, H.; Deng, X.; and Zhang, Y. 2022. Efficient Virtual View Selection for 3D Hand Pose Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Fang, L.; Liu, X.; Liu, L.; Xu, H.; and Kang, W. 2020. 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*, 120–137. Springer.

Ge, L.; Cai, Y.; Weng, J.; and Yuan, J. 2018a. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8417–8426.

Ge, L.; Liang, H.; Yuan, J.; and Thalmann, D. 2016. 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*, 3593–3601.

Ge, L.; Liang, H.; Yuan, J.; and Thalmann, D. 2018b. Real-time 3D hand pose estimation with 3D convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4): 956–970.

Ge, L.; Ren, Z.; and Yuan, J. 2018a. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision*, 475–491.

Ge, L.; Ren, Z.; and Yuan, J. 2018b. Point-to-Point Regression PointNet for 3D Hand Pose Estimation. In *Proceedings of the European Conference on Computer Vision*, 475–491.

Hampali, S.; Rad, M.; Oberweger, M.; and Lepetit, V. 2020. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3196–3206.

Hampali, S.; Sarkar, S. D.; and Lepetit, V. 2021. HO-3D_v3: Improving the Accuracy of Hand-Object Annotations of the HO-3D Dataset. *arXiv preprint arXiv:2107.00887*.

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

Huang, L.; Tan, J.; Liu, J.; and Yuan, J. 2020a. Hand-Transformer: Non-Autoregressive Structured Modeling for 3D Hand Pose Estimation. In *European Conference on Computer Vision*, 17–33. Springer.

Huang, W.; Ren, P.; Wang, J.; Qi, Q.; and Sun, H. 2020b. AWR: Adaptive Weighting Regression for 3D Hand Pose Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11061–11068.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. PMLR.

Kanazawa, A.; Black, M. J.; Jacobs, D. W.; and Malik, J. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7122–7131.

Li, J.; Xu, C.; Chen, Z.; Bian, S.; Yang, L.; and Lu, C. 2021. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3383–3393.

Li, S.; and Lee, D. 2019. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11927–11936.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31.

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

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11976–11986.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*.

Malik, J.; Abdelaziz, I.; Elhayek, A.; Shimada, S.; Ali, S. A.; Golyanik, V.; Theobalt, C.; and Stricker, D. 2020. 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*, 7113–7122.

Malik, J.; Shimada, S.; Elhayek, A.; Ali, S. A.; Golyanik, V.; Theobalt, C.; and Stricker, D. 2021. HandVoxNet++: 3D Hand Shape and Pose Estimation using Voxel-Based Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Moon, G.; Chang, Y.; and Lee, K. M. 2018. 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*, 5079–5088.

Oberweger, M.; and Lepetit, V. 2017. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 585–594.

Oberweger, M.; Riegler, G.; Wohlhart, P.; and Lepetit, V. 2016. Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4957–4965.

Oberweger, M.; Wohlhart, P.; and Lepetit, V. 2015. Hands deep in deep learning for hand pose estimation. In *Proceedings of the Computer Vision Winter Workshop*, 21–30.

Park, J.; Oh, Y.; Moon, G.; Choi, H.; and Lee, K. M. 2022. HandOccNet: Occlusion-Robust 3D Hand Mesh Estimation Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1496–1505.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 5099–5108.

Ran, H.; Liu, J.; and Wang, C. 2022. Surface Representation for Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18942–18952.

Ren, P.; Sun, H.; Hao, J.; Wang, J.; Qi, Q.; and Liao, J. 2022. Mining Multi-View Information: A Strong Self-Supervised Framework for Depth-Based 3D Hand Pose and Mesh Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20555–20565.

Ren, P.; Sun, H.; Huang, W.; Hao, J.; Cheng, D.; Qi, Q.; Wang, J.; and Liao, J. 2021. Spatial-aware stacked regression network for real-time 3d hand pose estimation. *Neurocomputing*, 437: 42–57.

Ren, P.; Sun, H.; Qi, Q.; Wang, J.; and Huang, W. 2019. SRN: Stacked Regression Network for Real-time 3D Hand Pose Estimation. In *Proceedings of the British Machine Vision Conference*, 112.

Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6): 245:1–245:17.

Spurr, A.; Iqbal, U.; Molchanov, P.; Hilliges, O.; and Kautz, J. 2020. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *European Conference on Computer Vision*, 211–228. Springer.

Sun, X.; Wei, Y.; Liang, S.; Tang, X.; and Sun, J. 2015. Cascaded Hand Pose Regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 824–832.

Tang, D.; Jin Chang, H.; Tejani, A.; and Kim, T.-K. 2014. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3786–3793.

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

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Tompson, J.; Stein, M.; Lecun, Y.; and Perlin, K. 2014. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5): 169:1–169:10.

Tse, T. H. E.; Kim, K. I.; Leonardis, A.; and Chang, H. J. 2022. Collaborative Learning for Hand and Object Reconstruction with Attention-guided Graph Convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1664–1674.

Wan, C.; Probst, T.; Van Gool, L.; and Yao, A. 2018. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5147–5156.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5): 146:1–146:12.

Xiong, F.; Zhang, B.; Xiao, Y.; Cao, Z.; Yu, T.; Zhou, J. T.; and Yuan, J. 2019. 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*, 793–802.

Yang, L.; Li, K.; Zhan, X.; Lv, J.; Xu, W.; Li, J.; and Lu, C. 2022. ArtiBoost: Boosting Articulated 3D Hand-Object Pose Estimation via Online Exploration and Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2750–2760.

Yuan, S.; Ye, Q.; Stenger, B.; Jain, S.; and Kim, T.-K. 2017. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4866–4874.

Zhang, H.; Tian, Y.; Zhou, X.; Ouyang, W.; Liu, Y.; Wang, L.; and Sun, Z. 2021. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11446–11456.

Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; and Metaxas, D. N. 2019. Semantic graph convolutional networks for 3D human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3425–3435.