PIXEL-ALIGNED NON-PARAMETRIC HAND MESH RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Non-parametric mesh reconstruction has recently shown significant progress in 3D hand and body applications. In these methods, mesh vertices and edges are visible to neural networks, enabling the possibility to establish a direct mapping between 2D image pixels and 3D mesh vertices. In this paper, we seek to establish and exploit this mapping with a simple and compact architecture. The network is designed with these considerations: 1) aggregating both local 2D image features from the encoder and 3D geometric features captured in the mesh decoder; 2) decoding coarse-to-fine meshes along the decoding layers to make the best use of the hierarchical multi-scale information. Specifically, we propose an end-to-end pipeline for hand mesh recovery tasks which consists of three phases: a 2D feature extractor constructing multi-scale feature maps, a feature mapping module transforming local 2D image features to 3D vertex features via 3D-to-2D projection, and a mesh decoder combining the graph convolution and self-attention to reconstruct mesh. The decoder aggregate both local image features in pixels and geometric features in vertices. It also regresses the mesh vertices in a coarse-to-fine manner, which can leverage multi-scale information. By exploiting the local connection and designing the mesh decoder, Our approach achieves state-of-the-art for hand mesh reconstruction on the public FreiHAND dataset.

1 INTRODUCTION

Reconstructing 3D hand mesh from a single RGB image has attracted tremendous attention as it has numerous applications in human-computer interactions (HCI), VR/AR, robotics, *etc*. Recent studies have made great efforts in the accurate hand mesh reconstruction and achieved very promising results (Lin et al., 2021b; Moon & Lee, 2020; Kulon et al., 2020; Ge et al., 2019; Hasson et al., 2019). Recent state-of-the-art approaches address the problem mainly by deep learning. These learning-based methods can be roughly divided into two categories according to the representation of the hand meshes, *i.e.*, the parametric approaches, and the non-parametric ones. The parametric approaches use a parametric model that projects hand meshes in a low dimensional space (*e.g.*, MANO (Romero et al., 2022)) and regresses the coefficients in the space (*e.g.*, the shape and pose parameters of MANO) to recover the 3D hand (Hasson et al., 2019). The non-parametric ones instead directly regress the mesh vertices using graph convolution neural network (Moon & Lee, 2020; Kulon et al., 2020; Chen et al., 2021b) or transformer (Lin et al., 2021b).

Non-parametric approaches have shown substantial improvement over the parametric ones in recent work, owing to the mapping between the image and the vertices is less non-linear than that between the image and the coefficients of the hand models (Taheri et al., 2021). Their pipelines (Kulon et al., 2020; Chen et al., 2021b; Lin et al., 2021b) usually consist of three stages: a 2D encoder extracts the global image feature, which is mapped to 3D mesh vertices before fed into a 3D mesh decoder operating on the vertices and edges to get the final mesh.

Despite the success, the potential of non-parametric approaches has not been fully uncovered with this pipeline. In parametric methods, vertices and edges are not visible to the network, and no operation is carried out in the manifold of the meshes; 2D image features are extracted only to learn a mapping between the image content and the hand model parameters. Conversely in non-parametric methods, operations on vertices and edges, such as graph convolutions or attention modules, are designed to aggregate the geometric features of the meshes. With the operation, vertices and edges are visible to

the networks; thus direct connections between pixels of the 2D image feature space and vertices of the 3D mesh can be established and operations in the decoder can aggregate both image features and geometric features, which can not be realized in the parametric methods. This connection and the aggregation, however, have not been fully explored by previous work.

In this paper, we seek to establish the connections and merge the local hand features from appearance in the input and geometry in the output. To this end, we utilize the pixel-aligned mapping module to establish the connections and propose a simple and compact architecture to deploy the connections. We design our network by making the following philosophical choices: 1) For the 2D feature extractor, we keep feature maps of different scales in the encoder instead of using the final global feature to enable 2D local information mapping to 3D. 2) We decode the mesh in the coarse-to-fine manner to make the best use of the multi-scale information. 3) Both image features and geometric features are aggregated in the operations of the mesh decoder rather than only geometric features.

Our design is shown in Figure 1. Multi-scale image features are naturally passed to the 3D mesh decoder. Our experiments show the design enables better alignment between the image and the reconstructed mesh. The aggregation of features not only improves the graph convolution network substantially but also gains large superiority over the attention mechanism with global features.

To summarize, our key contributions are 1) Operations are capable of aggregating both local 2D image features and 3D geometric features on meshes in different scales. 2) Connections between pixels of 2D image appearance in the encoder and vertices of 3D meshed in the decoder are established by a pixel-vertex mapping module. 3) A novel graph convolution architecture achieves state-of-the-art results on the FreiHAND benchmark.

2 Related Work

Mesh Reconstruction. Previous research methods employ pre-trained parametric human hand and human models, namely MANO (Romero et al., 2022), SMPL (Loper et al., 2015). And estimate the pose and shape coefficients of the parametric model. However, it is challenging to regress pose and shape coefficients directly from input images. Researchers propose to train network models with human priors, such as using skeletons (Lassner et al., 2017) or segmentation maps. Some researchers have proposed regressed SMPL parameters by relying on human key points and contour maps (Pavlakos et al., 2018; Tan et al., 2017) of the body. Coincidentally (Omran et al., 2018) utilized the segmentation map of the human body as a supervision condition. A weakly supervised approach (Kanazawa et al., 2018) using 2D keypoint reprojection and adversarial learning regression SMPL parameters. Hsiao-Yu Tung (Tung et al., 2017) proposed a self-supervised approach to regression of human parametric models.

Recently, model-free methods (Choi et al., 2020; Moon & Lee, 2020; Kolotouros et al., 2019) for directly regressing human pose and shape from input images have received increasing attention. Because it can express the nonlinear relationship between the image and the predicted 3D space. Researchers have explored various ways to represent the human body and hand using 3D mesh (Lin et al., 2021b; Kolotouros et al., 2019; Choi et al., 2020; Lin et al., 2021a; Litany et al., 2018; Ranjan et al., 2018; Verma et al., 2018; Wang et al., 2018; Moon & Lee, 2020), voxel spaces (Varol et al., 2018), or occupancy fields (Saito et al., 2019; Niemeyer et al., 2020; Xu et al., 2019; Saito et al., 2020; Peng et al., 2020). Among them, the voxel space method adopts a completely non-parametric method, which requires a lot of computing resources, and the output voxel needs to fit the body model to obtain the final human 3D mesh.Among the recent research methods, Graph Convolution Neural Networks (GCNs) (Kolotouros et al., 2019; Choi et al., 2020; Lin et al., 2021a; Litany et al., 2018; Ranjan et al., 2018; Verma et al., 2018; Wang et al., 2018; Moon & Lee, 2020) is one of the most popular methods. Because GCN is particularly convenient for convolution operations on mesh data. However, GCN is good for representing the local features of the mesh, and the global features of the long-distance interaction between human vertices and joints cannot be well represented. Transformer-based methods (Lin et al., 2021b) use a self-attention mechanism to take full advantage of the information interaction between vertex and joints and use the global information of the human body to reconstruct more accurate vertex positions. But whether it is a GCN-based method or an attention mechanism-based method. Neither considers pixel-level semantic feature alignment information. Local pixel-level semantic feature alignment can compensate for the global information that GCN and transformer methods focus on.

Graph Neural Networks. Graph deep learning generalizes neural networks to non-Euclidean domains, and we hope to apply graph convolution neural networks to learn shape-invariant features on triangular meshes. For example, spectral graph convolution neural network methods (Bruna et al., 2013; Defferrard et al., 2016; Kipf & Welling, 2016; Levie et al., 2018) perform convolution operations in the frequency domain. Local graph methods (Masci et al., 2015; Boscaini et al., 2016; Monti et al., 2017) based on spatial graph convolutions make deep learning on Manifold more convenient.

In the application of mesh reconstruction. (Ranjan et al., 2018) used fast local spectral filters to learn nonlinear representations of human faces. (Kulon et al., 2019) extended autoencoder networks to 3D representations of hands.Kolotouros proposed GraphCMR (Kolotouros et al., 2019) to regression 3D mesh vertices using a GCN (Kolotouros et al., 2019; Choi et al., 2020; Lin et al., 2021a; Litany et al., 2018; Ranjan et al., 2018; Verma et al., 2018; Wang et al., 2018; Moon & Lee, 2020). Pose2Mesh (Choi et al., 2020) proposes to reconstruct a human mesh from a given human pose representation based on a cascaded GCN. (Lim et al., 2018) proposed spiral convolution to handle mesh in the spatial domain. Based on SpiralConv, Kulon (Kulon et al., 2020) introduced an automatic method to generate training data from unannotated images for 3D hand reconstruction and pose estimation. (Chen et al., 2021b;a) propose a novel aggregation method to collect effective 2D cues and exploit high-level semantic relations for root-relative mesh recovery. Kevin Lin proposed Graphormer (Lin et al., 2021a), combining Transformer and GCN to simulate the global interaction between joints and mesh vertices.

Mesh-image alignment. In the field of 2D image processing, most deep learning methods employ a "fully convolution" network framework that maintains spatial alignment between images and outputs (Kirillov et al., 2020; Long et al., 2015; Tompson et al., 2014). Several research methods also consider alignment relationships in the 3D domain. For example, PIFu (Saito et al., 2019) proposed an implicit representation that locally aligns the pixels of a 2D image with the global context of their corresponding 3D objects. PyMAF (Zhang et al., 2021) introduced a mesh alignment feedback loop, where evidence of mesh alignment is used to correct parameters for better-aligned reconstruction results. The alignment can take advantage of more informative features that are sensitive to position to predict mesh.

Existing mesh recovery works (Tang et al., 2021; Li et al., 2022) face the shortcomings of complex network structure when mesh-images alignment. Furthermore, the initial input of the 3D decoder is a high-resolution mesh, which makes network optimization difficult. This is critical for practical applications. To address these issues, we propose a compact network framework to map 2d image pixel features to 3d mesh vertex locations. We apply a multi-scale structure to the 2D feature encoder and 3D mesh decoder respectively, to achieve coarse-to-fine pixel alignment at corresponding resolutions. Using multi-scale pixel-aligned features can achieve better mesh-image alignment than previous methods.

3 METHODOLOGY

Given a monocular RGB image I, our goal is to predict the 3D positions of all the N vertices $\mathcal{V} = \{v_i\}_{i=1}^{N}$ of the predefined hand mesh $\mathcal{M}$. The overall architecture of our network, as shown in Figure 1, has two major components: a 2D feature extractor, as well as a 3D mesh decoder that consists of feature mapping modules and mesh-conv layers. The 2D feature extractor is an hourglass that encodes the image content into features at S levels of scale. Respectively the 3D mesh decoder also recovers the vertices in a coarse-to-fine manner in S different scales. By design the mesh decoder at level $s \in S$ leverages the 2D feature map at level s. In the following sections, we will describe the architecture of the 2D feature extractor in 3.1, the pixel-aligned feature mapping module in 3.2, the mesh decoder in 3.3, as well as training details in 3.4.

3.1 2D FEATURE EXTRACTOR

In order to extract 2D features at different scales/receptive fields, we adopt a simple hourglass model with skip connections as the feature extractor. Previous works (Lin et al., 2021b; Kulon et al., 2020; Kolotouros et al., 2019) extract a global image feature vector and feed it to the decoder. Since all the vertices share the same global feature, only the geometric features relating to the overall deformation

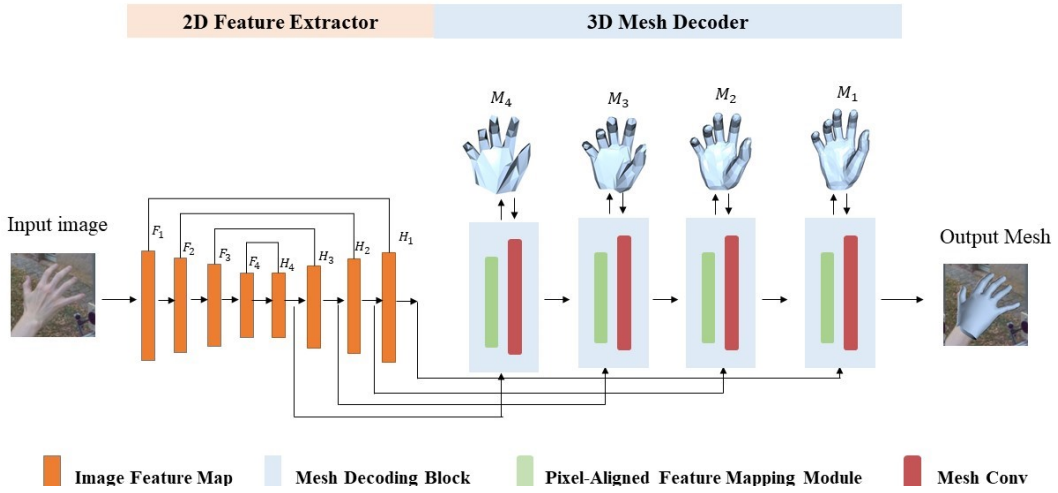

Figure 1: **Our full pipeline.** Given a single-hand RGB image as input, our network first extracts 2D feature maps with an hourglass module. Downsampled F_s and upsampled feature maps H_s are aggregated to have the multiscale image feature pyramids, which are mapped to vertices of meshes at different scales by our pixel-aligned mapping module and GCN blocks. Vertex position of the meshes can then be predicted in a coarse-to-fine manner.

of meshes are aggregated and updated for each vertex. To enable mapping local image features to the vertices, we combine the downsampled feature F_s and upsampled feature H_s from the respective level s of the hourglass network to have a fusion feature map $Q_s \in \mathbb{R}^{h_s * w_s * c_s}$ for the level s of the mesh decoder. Specifically,

$$Q_s = \mathbf{Conv1D}(\bigoplus(F_s, H_s)), \tag{1}$$

where $\bigoplus$ denotes concatenation, $\mathbf{Conv1D}$ denotes 1D convolution. h_s, w_s, and c_s represent height, width, and the number of channels of Q_s respectively.

3.2 Pixel-aligned Feature Mapping Module

Given a 2D image feature map $Q \in \mathbb{R}^{h*w*c}$, the mapping module needs to transform it into 3D vertex features $G \in \mathbb{R}^{N*c}$ of the corresponding mesh decoder layer, where N denotes the number of vertices as mentioned. For this purpose, previous methods (Kulon et al., 2020; Kolotouros et al., 2019; Lin et al., 2021b) either simply repeat the global feature from a feature extraction network to have the vertex features, or use a fully connected layer to map the global feature vector from $\mathbb{R}^c$ to $\mathbb{R}^{N*c}$ (the vertex features G are obtained by reshaping $F_g^{'}$). This mapping manner can not well build the relationship between the 2D and 3D domain and have difficulty in guaranteeing mesh-alignment with the input image (Zhang et al., 2021).

Inspired by (Wang et al., 2018; Saito et al., 2019), we utilize a pixel-aligned feature mapping module to transform the feature map Q_s to 3D vertex features $G_s \in \mathbb{R}^{N_s * c_s}$. Each predicted vertex $v \in \mathcal{V}_s$ is projected to pixel x in image space, as illustrated in Figure 2. After that, we sample the feature map Q_s to extract pixel-aligned vertex features G_s using the following equation:

$$G_s = f(Q_s, \pi(\mathcal{M}_s)), \tag{2}$$

where $\pi(.)$ denotes 2D projection, and $f(.)$ is a sampling function.

Similar to (Wang et al., 2018; Zhang et al., 2021) we use bilinear interpolation around each projected vertices on the feature maps to extract pixel-aligned feature vectors. The 2D feature maps $\{Q_s\}$ are multi-scale (7x7->14x14->28x28->56x56). Lower-resolution image features have a larger receptive field, hence more global information, while higher-resolution feature maps contain more local information. The pyramid feature maps and pixel-aligned feature mapping modules can provide richer content for vertices.

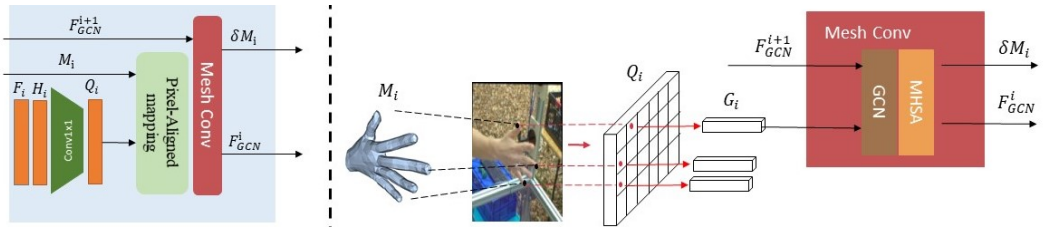

Figure 2: Left: The architecture of the mesh decoder. Right: The data flow of the mesh decoder. 3D vertices are projected to the 2D image plane to retrieve pixel-aligned features. The GCN blocks and the attention module form a mesh-conv layer to predict offset mesh.

3.3 Mesh Decoder

In the 2D feature extractor, feature maps of different scales are kept and in the mapping module, the connections of the features of pixels of these feature maps and features for the 3D vertices in the corresponding decoding layers are established. In this section, we aim to design a mesh decoder structure that makes full use of the connections that are made possible via the non-parametric mesh representation. To be specific, we aim to 1) design a mechanism to leverage the multi-scale image features in the hierarchical decoding architecture; 2) aggregate both local features from the feature extractor and the features extracted in the previous mesh decoder layers.

For the first aim, our decoder reconstructs hand meshes in a coarse-to-fine manner along with the hierarchical layers, shown in Figure 1. Our blocks skip connects the different vertex feature G_i from the mapping module. In each decoding layer, based on the reconstructed mesh, image features of the corresponding scale are sampled by the feature mapping module and then concatenated to the vertices features. Rather than directly regressing the 3D coordinates, we predict the offset mesh δM_i of last-block-estimated M_i. The coarse-to-fine mesh topologies for each block are acquired by mesh simplification (Gong et al., 2019). The number of vertices of different meshes is 98, 195, 389, and 778. The M_1 has the same mesh topology as MANO (Romero et al., 2022). The meshes at different scales are predicted by several additional mesh decoding headers. We use the strategy based on edge contraction (Garland & Heckbert, 1997) to do the pooling and unpooling operation. The input features are multiplied with a pre-computed transform matrix to obtain the output features.

We achieve the second aim by deploying operations that can work on vertices of meshes, *i.e.*, graph convolution operating on the manifold of the meshes and the attention module. In the ablation studies, we demonstrate that local features can significantly boost the performance of both operations compared with global features. Though in our approach, the feature mapping and the hierarchical decoding are only applied to decoders with graph convolutions and the attention modules, they can be injected into other 3D mesh decoders. In the experiment, we show that our components are also very effective for the transformer-based method (Lin et al., 2021b).

Graph Convolution For the graph convolution operators, spectral convolutions and spatial convolutions have been used in the mesh reconstruction. We use a spiral patch operator (Gong et al., 2019) to process vertex features in the spatial domain as it demonstrates superior performance in recent work. The spiral operator collects vertex neighbors of the center v. The graph convolution produces the offset mesh $\delta\mathcal{M}_s$ and feature F^s_{GCN} for each vertex. The spiral convolution of input $F^s_{GCN}(v)$ can be defined as:

$$(F^s_{GCN} * g)_v = \sum_{l=1}^{L} g_l F^s_{GCN}(O(v)) \tag{3}$$

where g_l is the convolution kernel, and $O(v)$ denotes the pre-computed vertices ordered sequence.

Self-Attention While GCN is useful for capturing fine-grained neighborhood information, it is less efficient at extracting long-range dependencies. We inject the multi-head self-attention module (Vaswani et al., 2017) into GCN blocks to address this challenge.

Given the vertex features F^s_{GCN} generated by GCN, we strengthen the global interactions and get a new feature F^s_{MHSA} for each vertex with the help of MHSA:

$$F^i_{MHSA} = MHSA(F^s_{GCN}), \tag{4}$$

The GCN blocks and the attention module form a mesh-conv layer as shown in Figure 2.

3.4 TRAINING

We denote a dataset as a set of tuples $\{(I, P, C, \mathcal{M})\}$, where I is the input image and $\mathcal{M}$ is the hand mesh; P and C are the heatmaps of 2D pose and silhouette, respectively. Following (Moon & Lee, 2020), We apply the Gaussian distribution to construct 2D pose heatmaps.

3D Supervision One concern for non-parametric methods is the lack of kinematic and shape constraints. Adding 3D shape supervision can mitigate that. To this end, we adopt an L1 loss norm L_{mesh} to supervise the 3D coordinates of vertices in the coarse-to-fine manner. Besides, we use the normal loss L_{norm} and edge length loss L_{edge} in (Wang et al., 2018) for smoother reconstruction meshes in the last mesh decoding layer. So the loss for a sample in the dataset is defined as:

$$\begin{aligned} L_{mesh} &= \sum_{s=1}^{S} \lambda_m^s * ||\hat{\mathcal{M}}_s - \mathcal{M}_s||_1 \\ L_{edge} &= \sum_{t \in \mathcal{T}} \sum_{e \in t} |||\hat{e}| - |e||||_1 \\ L_{norm} &= \sum_{t \in \mathcal{T}} \sum_{e \in t} ||\hat{e} \cdot n||_1 \end{aligned} \tag{5}$$

where t is a triangle face from all the faces $\mathcal{T}$ of $\mathcal{M}$, e denotes an edge of t, and n the normal vector of t computed from the ground truth. To ensure the performance of the reconstruction of the finest mesh, we set a weight λ_m^s for different scale mesh recontruction loss.

2D Auxiliary Supervision For the 2D feature extractor, we add auxiliary supervision for better feature extraction. We apply binary-cross-entropy (BCE) loss to formulate both the silhouette loss L_{sil} and 2D pose loss L_{2Dpose} as follows:

$$\begin{aligned} L_{sil} &= -(\sum_j (x_j^C \log(\hat{x_j^C}) + (1 - x_j^C) \log(1 - \hat{x_j^C}))), x^C \in C \\ L_{2Dpose} &= -(\sum_j (x_j^P \log(\hat{x_j^P}) + (1 - x_j^P) \log(1 - \hat{x_j^P}))), x^P \in P, \end{aligned} \tag{6}$$

where x_j^C and x_j^P denotes the j-th pixel value of the silhouette and pose heatmaps respectively, and $\hat{}$ denotes the prediction.

The overall loss is a weighted sum of all losses, $L_{total} = L_{mesh} + L_{edge} + \lambda_n * L_{norm} + \lambda_s * L_{sil} + \lambda_p * L_{2Dpose}$, where $\lambda_n = 0.1$, $\lambda_p = 10$, $\lambda_s = 2.5$ are set empirically.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUPS

FreiHAND (Zimmermann et al., 2019) is the most widely-used benchmark for hand mesh reconstruction. It contains 130K images for training with 3D and 2D annotations, and the test set has 4K images. As the annotations of the test set are not available, we submit our results to their provided online server for evaluation.

Training Procedure. Our network is based on the HRNet-W64 (Wang et al., 2020) and ResNet-50 (He et al., 2016) backbone pre-trained on ImageNet (Krizhevsky et al., 2012). We trained our model in an end-to-end manner. 2D keypoints and masks are used to train the image feature extractors as auxiliary supervision. We resize the input image to 224×224. Following previous work, data augmentations including rotation, translation, color jitter, etc are applied during training. Our method is implemented in Pytorch and trained on Nvidia RTX 3090 GPU with a batch size of 64 for 50 epochs. We optimize the network by Adam with a learning rate of 1e-4 and set a decay rate of 0.1 at 35 epochs.

Table 1: Analysis of the self-attention and mapping module

Feat.	Mapping Module	Self-attention	PA-MPJPE↓	PA-MPVPE↓
Global	Repeat	✗	10.1	10.3
Global	MLP	✗	9.1	9.2
Feature maps(w/o 2D)	Pixel-aligned	✗	8.3	8.4
Feature maps(w/ 2D)	Pixel-aligned	✗	7.8	7.9
Global	MLP	✓	8.4	8.6
Feature maps	Pixel-aligned	✓	7.5	7.7

Evaluation metrics. We report the results of our approach using several evaluation metrics. PA-MPJPE: It first performs the rigid alignment between the prediction and ground-truth using Procrustes Analysis (Gower, 1975), then calculates the mean-per-joint-position-error. PA-MPVPE: similar to PA-MPJPE, but it measures the difference between the vertices predicted and the ground truth. The unit for the PA-MPJPE/PA-MPVPE metrics is millimeter (mm). F-scores: It measures the harmonic mean between recall and precision between two meshes. We report the F@5mm and F@15mm as existing works.

Table 2: Analysis of adding skip connections in different decoder layers.

M_4	M_3	M_2	M_1	PJ↓	PV↓
✓	✗	✗	✗	8.25	8.31
✓	✓	✗	✗	8.06	8.13
✓	✓	✓	✗	7.93	8.02
✓	✓	✓	✓	7.83	7.93

Table 3: Effectiveness of coarse-to-fine for pixel-aligned mapping module

Methods	PJ↓	PV↓
refine(A time)	8.31	8.42
refine(Three times)	8.12	8.22
Ours	7.83	7.93

4.2 Ablation Studies

We conduct ablation studies under various settings on FreiHAND (Zimmermann et al., 2019) to investigate the key components of our model. We use ResNet-18 as the backbone and report the results using PA-MPJPE and PA-MPVPE. Table 1 shows all the comparisons. Our final model (last row) improves the baseline with the global feature (1st row) by 26%.

Effectiveness of Mapping Module for Graph Convolution To establish the relationships between the 2D pixels and 3D vertices, We utilize the pixel-aligned feature mapping module. To verify the efficacy, we construct baselines based on GCN and compare their reconstruction accuracy in Table 1 with the variant of our method. Similar to (Kulon et al., 2020), we implement a baseline that directly repeats the global feature and concatenates it to the mesh vertices (1st row) and a baseline that maps the global feature to the vertex features by MLP layers(2nd row). For fair comparisons, we construct our variant with the pixel-aligned module which has the same architecture for the mesh decoder with these two baselines and do not add the 2D supervision (3rd row). Table 1 shows that the pixel-aligned module improves the reconstruction performance by a large margin, about 18% and 8%.

Effectiveness of Attention and Mapping Module for Attention Adding attention to graph convolution can complement the long-range information aggregation and hence improve the network capacity. To verify this, we compare a network using the graph convolution only (2nd row) and a network with an attention module in between graph convolution layers (5th row) in Table 1. The results show that the attention mechanism can improve the mesh reconstruction with only global features. Based on the same graph attention architecture, adding the mapping module can further improve the performance by 0.9mm in PA-MPJPE.

Effectiveness of 2D Auxiliary Supervision We analyze how feature maps affect. We implement two models with the same architecture: one has 2D supervision from silhouette and 2D pose (4th row) and one without (3rd row). It shows that the feature maps with supervision are helpful for improving

Table 4: Performance comparison with the state-of-the-art methods on the FreiHAND dataset.↓ means the lower the better, ↑ means the higher the better. The unit of PA-MPJPE/PA-MPVPE is mm.

Methods	PA-MPJPE↓	PA-MPVPE↓	F@5mm↑	F@15mm↑
(Hasson et al., 2019)	13.3	13.3	0.429	0.907
(Zimmermann et al., 2019)	10.9	11.0	0.516	0.934
(Kulon et al., 2020)	8.4	8.6	0.614	0.966
Pose2Mesh(Choi et al., 2020)	7.7	7.8	0.674	0.969
I2LMeshNet(Moon & Lee, 2020)	7.4	7.6	0.681	0.973
(Tang et al., 2021)	7.1	7.1	0.706	0.977
(Chen et al., 2021b)	6.9	7.0	0.715	0.977
METRO(Lin et al., 2021b)	6.8	6.7	0.717	0.981
Graphormer(Lin et al., 2021a)	6.0	5.9	0.764	0.986
Ours with METRO	**6.1**	**6.2**	**0.757**	**0.984**
Ours with GCN	**5.9**	**6.0**	**0.766**	**0.985**

performance. As 3D mesh annotation is expensive to acquire while 2D pose can be relatively cheap to label, we expect our method can benefit more from extra 2D auxiliary supervision.

Adding Skip Connections in Different Decoder Layers To analyze the finer local features in the mesh reconstruction, variants of our method are constructed by gradually removing the skip connections in the last decoding layer and the results of these variants are shown in Table 2. Note that, all approaches do not add the self-attention module. We observe a gradual reduction in the errors when multi-scale mesh-alignment evidences are removed from decoder layers. It proves that our skip connection design improves accuracy by leveraging multi-scale information.

Effectiveness of Coarse-to-fine for Pixel-aligned Mapping Module We reproduced (Tang et al., 2021) pipeline with our mesh decoder architecture to compare with another pixel-aligned-based method. Rather than regress hand mesh vertices in a coarse-to-fine manner like ours, they use the full mesh to refine. In the first row, we follow their pipeline to refine the rough mesh one time, and for a more intuitive comparison with our method, we refine the mesh three times. Table 3 shows our coarse-to-fine manner can achieve better results for both two cases.

The Generality of Mapping Module and Skip Connections We verify that our pixel-aligned mapping module and skip connections are not only effective for our approach but also can improve the performance of other non-parametric models. Based on METRO (Lin et al., 2021b) which are transformer-based methods, we re-implement their networks with our design. We provide the details of network architecture in the appendix due to the space limitation. As Table 4 shows, pixel-aligned mapping module and skip connections can improve the METRO by a large margin.

4.3 Comparisons with State-of-the-Art

We follow the former works (Zimmermann et al., 2019; Hasson et al., 2019; Kulon et al., 2020; Choi et al., 2020; Moon & Lee, 2020; Lin et al., 2021b;a) to quantitatively compare our method with state-of-the-art methods on the FreiHAND eval set. For GCN, we re-design our 2D feature extractor as CMR (Chen et al., 2021b) for further performance improvement. For Transformer, we design the architecture as introduced in 4.2. As shown in Table 4, All of our approaches achieve state-of-the-art for all the metrics. It demonstrates that our designs can be effective for both non-parametric models.

Qualitative Results Figure 3 shows reconstructed meshes from several testing examples of FreiHAND. On the left side, our method aligns the meshes better to the input images than the baseline with global features only. On the right side, decoded meshes in different scales are shown. Notice that the meshes from the last layers adjust both the global locations to align the mesh to the inputs (*e.g.*, the meshes in the 3rd row) and the geometry of the meshes (*e.g.*, the thickness and smoothness of the meshes).

Figure 4 shows three typical failure cases of our method. In the first row, when the hand is severely occluded and the hands are not bounded by the crop size, some parts of the hand out of the image, our method fails to recover a correct hand mesh. In the second row, we observe when only a small portion

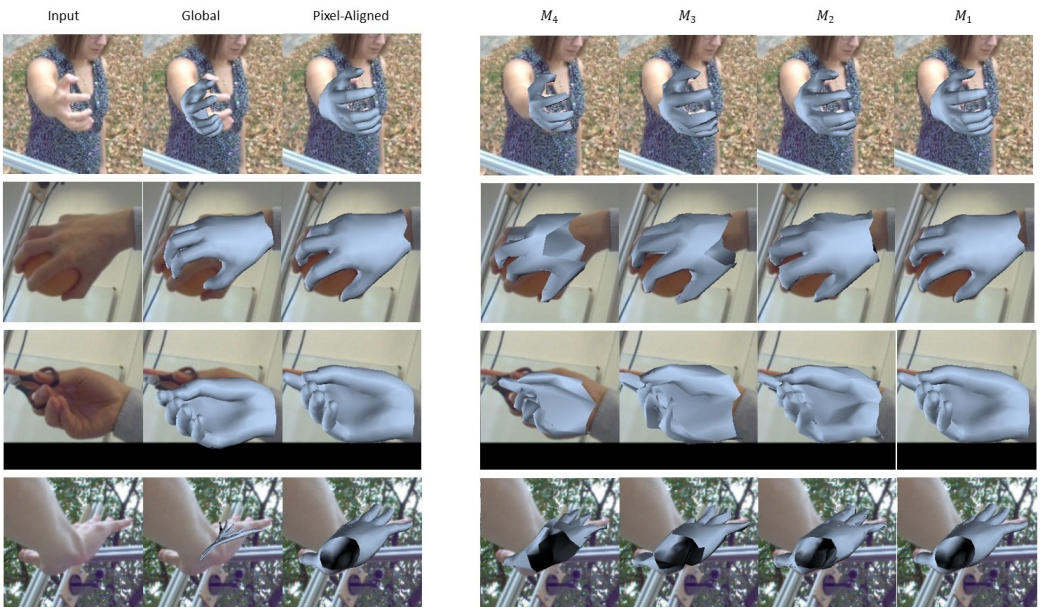

Figure 3: Left: comparison of the baseline with global features and our method; Right: Example meshes from M_4 to M_1.

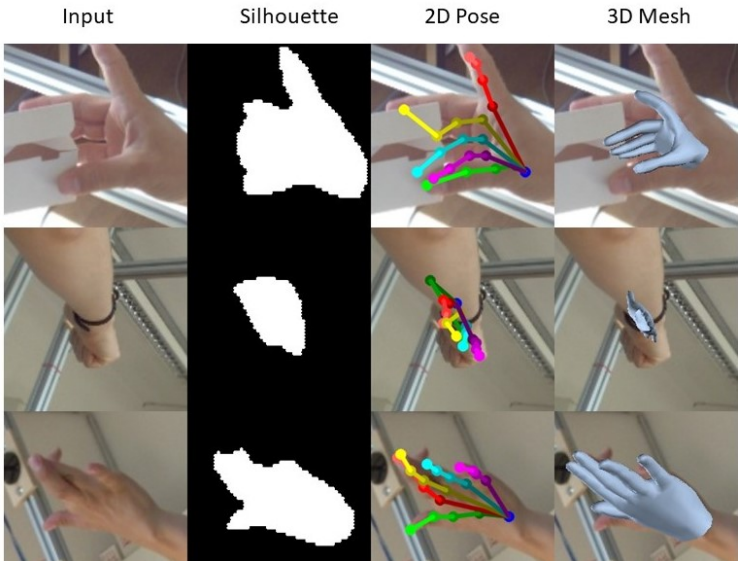

Figure 4: Failure cases of our method.

of the hand is visible, our method predicts the wrong 2D pose and silhouette as well as the hand mesh. Referring to the last row, although the overall shape seems to be reasonable, it is difficult to obtain an accurate 3D mesh due to the heavy self-occlusion. The self-occlusion is one of the biggest challenges for 3D hand mesh reconstruction or pose estimation.

5 Conclusions

We present a new pipeline to reconstruct a hand from a single RGB image. Specifically, we introduce a simple and compact architecture that can help align the mesh to the image, together with adding the self-attention module to improve vertices interactions. Comprehensive experiments show our method achieves the state-of-the-art on the FreiHAND dataset and verifies the effectiveness of our proposed key components. We further demonstrated that our design can also improve the performance of the transformer-based method. It shows that our proposed components have great generality for non-parametric models.

REFERENCES

Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. *arXiv preprint arXiv:2112.02753*, 2021a.

Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13274–13283, 2021b.

Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pp. 769–787. Springer, 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 209–216, 1997.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10833–10842, 2019.

Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

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/CVF conference on computer vision and pattern recognition*, pp. 11807–11816, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7122–7131, 2018.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9799–9808, 2020.

Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4501–4510, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael Bronstein, and Stefanos Zafeiriou. Single image 3d hand reconstruction with mesh convolutions. *arXiv preprint arXiv:1905.01326*, 2019.

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/CVF conference on computer vision and pattern recognition*, pp. 4990–5000, 2020.

Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6050–6059, 2017.

Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. Interacting attention graph for single image two-hand reconstruction. *arXiv preprint arXiv:2203.09364*, 2022.

Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12939–12948, 2021a.

Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1954–1963, 2021b.

Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1886–1895, 2018.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 37–45, 2015.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5115–5124, 2017.

Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pp. 752–768. Springer, 2020.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020.

Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 international conference on 3D vision (3DV)*, pp. 484–494. IEEE, 2018.

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 459–468, 2018.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pp. 523–540. Springer, 2020.

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 704–720, 2018.

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2304–2314, 2019.

Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 84–93, 2020.

Omid Taheri, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas. Goal: Generating 4d whole-body motion for hand-object grasping. *arXiv preprint arXiv:2112.11454*, 2021.

Jun Kai Vince Tan, Ignas Budvytis, and Roberto Cipolla. Indirect deep structured learning for 3d human body shape and pose prediction. 2017.

Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11698–11707, 2021.

Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in neural information processing systems*, 27, 2014.

Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. *Advances in Neural Information Processing Systems*, 30, 2017.

Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–36, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2598–2606, 2018.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 52–67, 2018.

Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in Neural Information Processing Systems*, 32, 2019.

Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11446–11456, 2021.

Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 813–822, 2019.