This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Skeleton-Aware 3D Human Shape Reconstruction From Point Clouds

Haiyong Jiang[1]*, Jianfei Cai[1,2], Jianmin Zheng[1]
[1]Nanyang Technological University, Singapore, [2]Faculty of Information Technology, Monash University

Abstract

This work addresses the problem of 3D human shape reconstruction from point clouds. Considering that human shapes are of high dimensions and with large articulations, we adopt the state-of-the-art parametric human body model, SMPL, to reduce the dimension of learning space and generate smooth and valid reconstruction. However, SMPL parameters, especially pose parameters, are not easy to learn because of ambiguity and locality of the pose representation. Thus, we propose to incorporate skeleton awareness into the deep learning based regression of SMPL parameters for 3D human shape reconstruction. Our basic idea is to use the state-of-the-art technique PointNet++ to extract point features, and then map point features to skeleton joint features and finally to SMPL parameters for the reconstruction from point clouds. Particularly, we develop an end-to-end framework, where we propose a graph aggregation module to augment PointNet++ by extracting better point features, an attention module to better map unordered point features into ordered skeleton joint features, and a skeleton graph module to extract better joint features for SMPL parameter regression. The entire framework network is first trained in an end-to-end manner on synthesized dataset, and then online fine-tuned on unseen dataset with unsupervised loss to bridges gaps between training and testing. The experiments on multiple datasets show that our method is on par with the state-of-the-art solution.

1. Introduction

3D human reconstruction is of great interest in computer graphic and computer vision due to its wide applications, such as personalized human model in VR and AR applications, human body measurements in virtual dressing rooms, and human modeling in video-based motion capture [48]. Though the reconstruction from images has made great progress, it is still difficult to efficiently and reliably reconstruct accurate body models due to the ambiguity caused by 3D projection. On the other hand, the increasing popularity of depth cameras such as Kinect and 3D scanning devices makes the capturing of point clouds become easier, which opens another door for more reliable 3D reconstruction. In this paper, we focus on the problem of 3D human reconstruction from point clouds.

Reconstructing a high-quality human shape is challenging because of non-rigid human deformations, low-quality input data, and joint articulations. Emerging deep learning techniques make it possible to reconstruct human shape in an end-to-end fashion [42, 31, 16, 13, 20]. Since 3D human mesh is of high-dimension (e.g. with 6890 vertices in Dyna dataset [32]), directly learning such a high-dimensional mesh with articulations is extremely difficult. Previous works have explored deep neural networks for 3D human reconstruction, but the results can be either rugged [20], blurring [42], or even twisted [13]. Fortunately, SMPL [21] offers a nice compact representation for 3D human shape, and it has been integrated with deep neural networks for 3D human reconstruction from RGB images in [31, 16]. The basic pipeline is to use deep neural networks to extract powerful image features, then directly regress SMPL shape and pose parameters, and finally use the off-the-shelf SMPL model to generate the reconstructed mesh. However, to our knowledge, no work has been done to use the SMPL model in deep learning based 3D human reconstruction from point clouds.

On a separate track, deep learning based point cloud analysis has also made great progress. The state-of-the-art techniques, PointNet and PointNet++ [34, 35], have proven their capability in extracting powerful features from 3D point clouds for classification and segmentation tasks. Thus, for 3D human reconstruction from point clouds, a natural idea would be combining PointNet++ with SMPL model, i.e. using PointNet++ to extract features from point clouds, then directly regressing SMPL parameters from the extracted point features, and finally using SMPL model to get the 3D mesh.

However, there is a major issue for such a pipeline. That is, it is hard to directly regress SMPL parameters from image features according to [31, 16] or point cloud features according to our study. This is because SMPL shape and pose parameters interact in a highly nonlinear way. By noticing that SMPL parameters are joint-sensitive and the pose

*mail: hyjiang@ntu.edu.sg

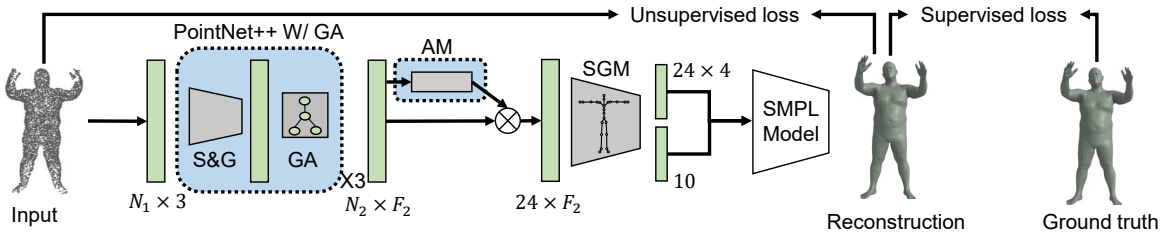

Figure 1. Overview of the propose network architecture which takes 3D point cloud as input and outputs SMPL shape and pose parameters. The entire network consists of three main modules: a modified PointNet++ module (PointNet++ W/ graph aggregation (GA)) to extract point cloud features, an attention module (AM) to help map unordered point features into ordered skeleton joint features, and a skeleton graph module (SGM) that uses the graph convolution to extract joint features to regress SMPL parameters. The obtained SMPL parameters are then fed to the off-the-shelf SMPL model to obtain the reconstructed 3D human mesh, which is compared with the ground truth for supervised training and compared with the input point cloud during testing as an online tuning.

parameters largely rely on skeleton joints, we propose to introduce the skeleton awareness into the pipeline. Particularly, we propose to replace the mapping from point features to SMPL parameters by the mapping from point features to skeleton features and then from skeleton features to SMPL parameters.

Nevertheless, this new pipeline introduces another obstacle. As we know, the point features extracted by PointNet++ is orderless since it needs to accommodate point permutations, while the subsequent joint features we need follow the special order of the skeleton graph. Mapping from unorder point features to ordered joint features while being robust to permutations of points is not trivial, for which we propose an attention module (AM). In addition, by noticing that PointNet++ still learns features on individual points independently with a multi-layer perception (MLP) and accumulates local contexts by pooling among neighbors, we propose a local graph aggregation (GA) module based on the graph convolution to leverage local contexts among neighbors without the burden of huge memory demands and loss of point interactions. Moreover, we also propose a skeleton graph module (SGM) based on the graph convolution to learn better joint features by leveraging the joint dependencies in the skeleton graph. Fig. 1 depicts the entire network pipeline, which is being trained end-to-end on synthesized data. An online tuning step is also introduced to exploit unsupervised loss to alleviate dataset gaps.

The major contributions of this paper are twofold.

- We propose to incorporate skeleton awareness into the deep learning based regression of SMPL parameters for 3D human reconstruction. Particularly, we introduce the general pipeline of mapping from point features to skeleton joint features and then to SMPL parameters for the reconstruction from point clouds.
- We develop an end-to-end framework, where we propose a graph aggregation module that is added into PointNet++ to extract better point features, an attention module to better map unordered point features that is added into ordered skeleton joint features, and a skeleton graph module to extract better joint features for SMPL parameter regression.

We conduct experiments on four datasets, which show that our method is on par with the state-of-the-art.

2. Related Work

In this section, we review related works on 3D human shape reconstruction, skeleton-based human analysis, point cloud analysis, and graph neural networks.

3D Human Shape Reconstruction: With the proliferation of deep learning, recent works try to use a neural network to directly learn to reconstruct 3D human from point clouds [13, 20] or images [42, 31, 16, 5, 27, 1, 28]. Groueix et al. [13] directly learned to deform a given template for human reconstruction, but often obtained twisted human shapes, especially in the shape arms. Litany et al. [20] proposed a variational auto-encoder to learn for deformable shape completion, which often results in rugged surfaces. Varol et al. [42] learned to reconstruct volumetric human shapes with a low resolution volumetric representation. Reconstruction from a single images [31, 16, 5] has also become feasible by leveraging parametric human models. Another different methodology [40, 33, 47] is to reconstruct human shape by predicting dense correspondence to a body surface. In our work, we also use SMPL model [21] as the human shape representation.

Another line of related research is dynamic human reconstruction or motion capture, which explores temporal consistency of acting persons [48, 15, 14] and can even reconstruct clothes and textures [54, 4, 2]. However, robust human pose estimation is still an open problem, especially for fast motion, and sequence-based human reconstruction heavily relies on a good initialization of human pose. Thus, IMU sensor is introduced for robust pose estimation in recent works [55, 45]. Our work focuses on a different scenario, i.e. 3D human reconstruction from raw point clouds.

Skeleton-based Human Analysis: Skeleton information is widely used in motion capture [48, 51, 52], and human reconstruction [10, 21, 22, 53]. Parametric human models, e.g. SMPL [21], rely on human skeleton for shape skinning. Several human reconstruction methods [10, 51, 48] also utilize skeleton estimation as a guidance. However, all these works mainly focus on exploring skeleton joint positions and neglect relations among them. Lee et al. [19] proposed to use LSTM to leverage joint relations, but it only allows to propagate features from parent joints to their children. In contrast, we propose to use graph convolution network (GCN) to propagate features among connected skeleton graph joints, which facilitates the exploration of both parent and children joint features.

Another category of related works is about human pose estimation [37, 38, 9, 29, 30], which focuses on predicting 2D or 3D joint positions. The state-of-the-art 2D pose estimation methods [9] can nicely predict human joints even if multiple person interactions exist, whose success partially owes to the supervision of relations among human joints. Estimation of 3D joint pose is still an unresolved problem, largely due to the difficulty of 3D labeling and the ambiguity in 3D space. Recent works [37, 38, 30] harvest 3D joint prediction in an unsupervised way by exploring multi-view consistency or geometry consistency, but do not explore joint dependency for pose estimation.

Learning-based Point Cloud Analysis: Point cloud analysis has attracted lots of interests in computer vision community, because of its important role in 3D analysis [34, 35, 50, 39]. Pioneering works [34, 35] introduced several important concepts in point clouds analysis, including invariance to point permutations and capture of point interactions. However, the state-of-the-art method, PointNet++ [35], which uses a multi-layer perception (MLP) for single point feature learning and pooling among neighbors to obtain permutation-invariant features, sacrifices important local information. Though Klokov et al. [18] used kd-tree to tackle this problems, but the splitting position in kd-tree may vary abruptly when point clouds are rotated. A good solution to this problem is EdgeConv [46], but they require a global knn graph, which results in $O(N^2)$ complexity in both space and time. In contrast, we use a local KNN graph constructed with very few points to capture point interactions with neighbors and apply the fast graph convolution to extract inter-related point features.

Neural Network on Graph Structure: Learning features on irregular graphs has become popular in many applications such as 3D geometric data analysis [23, 44], social network analysis [24], action recognition [49], and pose estimation [12, 8]. Existing graph convolutional networks (GCNs) can be divided into two mainstreams: spatial-based and spectral-based. Spatial-based methods [25, 41] learn features by directly filtering local neighbors on graph, and only a limited number of neighbors can be considered in each layer because of memory restriction. Spectral-based methods [7, 23] learn features in Fourier domain constructed by the eigen-decomposition of Laplacian matrix. However, the unstable and computationally expensive eigen-decomposition makes it unsuitable to process noisy point data. A compromise is the fast spectral convolution on graphs [36, 17, 11], which uses a k-order Chebyshev polynomial to approximate the spectral convolution and thus avoids eigen-decomposition. In this work, we adopt the fast spectral convolution and apply it on point graph as well as skeleton graph.

3. Preliminary

3.1. Parametric Human Model

Parametric human models, e.g. SCAPE [3] and SMPL [21], offer a compact representation of human shapes by encoding its variations as a function of shape and pose parameters. Particularly, the state-of-the-art representation of SMPL provides many benefits. Firstly, human shape and pose are disentangled, which allows independent analysis or control of shape or pose [31, 16]. Secondly, SMPL avoids the direct modeling of rugged and twisted shapes, which are headaches for neural network based methods [42, 20, 13], by modeling the deformation with a skinning process. Lastly, SMPL is differentiable and thus can be easily integrated with neural networks [31, 16]. In this research, we adopt SMPL as the underlying representation to model 3D human.

In particular, SMPL is composed of shape parameters, pose parameters, and global translation parameters. Shape parameters $\beta \in \mathbb{R}^{10}$ are used for shape blending, and encode the global shape information. Pose parameters are used for pose blending and skinning, and encode local information between adjacent joints with the exception that the pose parameters of the root joint denotes the global rotation of the whole shape. Note that pose parameters in SMPL denote the relative rotation from a joint to its parent. It is different from 2D or 3D human pose estimation [37, 9], where the pose refers to joint locations. An example is shown in Fig. 2. Although the original SMPL model uses axis-angle representation for pose parameters, we choose quaternion

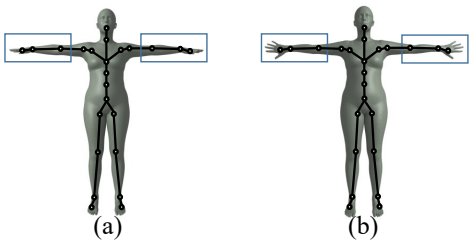

Figure 2. Two different human poses but with same joint positions.

representation since it relieves the ambiguity in axis-angle representation [56]. This leads to pose parameters $\alpha \in \mathbb{R}^{96}$ for 24 joints with each joint represented by quaternion representation with four values. In this work, we do not consider the global translation parameters, as it can be easily inferred once human pose is known or handled by normalizing input point clouds.

3.2. Convolutions on Graph Structures

The previous works on human joint or pose estimation [37, 38, 29] mainly use a multi-layer perceptron to estimate poses. We argue that such design is hard to learn relations between joints and propose to use graph convolution operation to exploit joint relations for feature learning. Specifically, we adopt the fast localized spectral convolution [36, 17, 11] to capture joint dependency. Moreover, we also propose to use the graph convolution to capture point interactions in 3D point clouds by constructing a neighborhood graph formed by linking k-nearest neighbors.

Consider a graph denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{W})$, where $\mathcal{V}$ is a set of n nodes and $\mathcal{W}_{n\times n}$ is the adjacent matrix with element $w_{ij} = 1$ indicating a connection between vertex i and vertex j and $w_{ij} = 0$ for no connection. The graph convolution is defined on the normalized Laplacian matrix $\mathcal{L}$, where $\mathcal{L} = \mathcal{D}^{-\frac{1}{2}}(\mathcal{D} - \mathcal{W})\mathcal{D}^{-\frac{1}{2}}$ with the diagonal matrix $\mathcal{D}$ defined as $d_{ii} = \sum_j^n w_{ij}$. The early spectral graph convolutions require to work on the Fourier space determined by Eigen decomposition of $\mathcal{L}$, which is unstable and computationally expensive. Thus, the fast localized spectral convolution [36, 17, 11] is introduced by using K-order Chebyshev polynomial to approximate the spectral convolution:

$$y_j = \sum_{i=1}^{F_{in}} G_{\theta_{ij}}(\mathcal{L}) \cdot x_i, \tag{1}$$

where $x_i \in \mathcal{R}^n$ and $y_j \in \mathcal{R}^n$ are the i-th input feature map and the j-th output feature map, respectively, and $\theta_{ij} \in \mathcal{R}^K$ denotes learning parameters. The graph convolution filters are calculated by $G_{\theta_{ij}}(\mathcal{L}) = \sum_{k=0}^{K-1} \theta_{ij,k} T_k(\mathcal{L})$, where the k-th order Chebyshev polynomial $T_k(\mathcal{L}) = 2 \cdot \mathcal{L} \cdot T_{k-1}(\mathcal{L}) - T_{k-2}(\mathcal{L})$ with $T_1(\mathcal{L}) = \mathcal{L}$ and $T_0(\mathcal{L})$ being the identity matrix in $R^{n\times n}$.

4. The Proposed Method

Overview. Fig. 1 gives an overview of the proposed network architecture, which takes 3D point cloud $\{\mathbf{p_i}\}^{N_1}$ with N_1 points as input and outputs SMPL shape and pose parameters that is subsequently fed into to the off-the-shelf SMPL model to obtain the reconstructed 3D human mesh. The entire network mainly consists of three modules: a modified PointNet++ module (PointNet++ W/ graph aggregation (GA)) to extract point cloud features, an attention module (AM) to help map unordered point features into ordered skeleton joint features, and a skeleton graph module (SGM) that uses the graph convolution to extract joint features to regress SMPL parameters. Finally, the estimated SMPL parameters are fed into the off-the-shelf SMPL model to obtain the reconstructed 3D human mesh, which is compared with the ground truth for supervised training and compared with the input point cloud during testing as an online tuning. Note that SMPL model is differentiable, and thus backward gradient can be easily obtained by back-propagating through SMPL functions. In the following, we describe the three major modules in detail.

4.1. Feature Learning for Point Clouds

In this step, we adopt PointNet++ [35] as the backbone to extract features defined on N_2 sampled points, which are obtained by furthest point sampling as in PointNet++. Although PointNet++ is a very powerful feature extraction framework for point clouds, its convolutional operation is still performed on each single point (see Fig. 3(a)), which does not properly explore point interactions. Motivated by the great success of convolutional neural network (CNN) on images, which learns features on neighboring pixels by convoluting them with different types of filters, we propose to modify PointNet++ by incorporating the graph convolution to learn local patterns.

In particular, given a set of points, we sample and group local neighbors (S&G module) as PointNet++, but learn features for each point group by a graph aggregation (GA) module as shown in Fig. 3(b), instead of using PointNet. Specifically, GA module constructs a local graph based on Euclidean distance of neighboring points, and only k nearest neighbors ($k = 2$ in our experiments) are kept. Then we calculate Laplacian matrix $\mathcal{L}$, and perform the fast localized spectral convolution as described in Sec. 3.2. This method facilitates learning different weights for point features of neighbors in different hops. In our implementation, S&G module and SA module with different parameters are applied totally three times for different resolutions, and output the final features on N_2 points as shown in Fig. 1, where we set $N_2 = 64$ and use $1024, 256, 64$ points for three S&G modules.

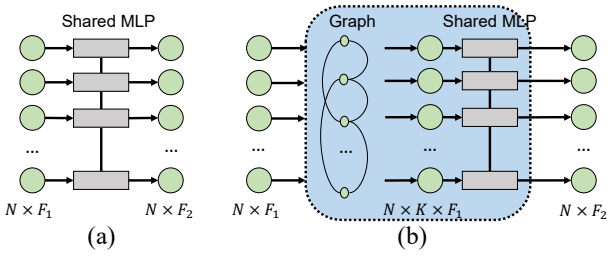

Figure 3. Comparison between PointNet feature learning (a) and the proposed graph aggregation module (b).

4.2. Attention Module

With the extracted N_2 point features, the next step is to map them into the features of N_3 skeleton joints, as shown in Fig. 1. However, the N_2 points are not in any specific order because of the randomness in point clouds. PointNet++[35] uses a pooling operation to aggregate local point features into a global one, as shown in Fig. 4(a), which offers an invariance to permutations of points but results in loss in point features.

In this research, we propose an attention module to preserve the local point features while being invariant to point orders. Particularly, the attention module dynamically learns relative contributions of each point to different skeleton joints according to point features (see Fig. 4(b)). The contribution weights are learned by a multi-layer perception (MLP) network adjusted according to their pertinence to skeleton joints. We take both pooling features and features on each point to predict these relative weights. There are some alternative choices shown in Fig. 4(c,d). The one in Fig. 4(c) directly replicates the pooling feature N_3 times, which will not work well since features on different joints are the same. Fig. 4(d) directly use MLP for the mapping. The weights are fixed once learned, and thus cannot dynamically adapt to permutation or changes in input. In our implementation, N_3 is set to 24 as defined in SMPL. An example of 24 joints is shown in Fig. 2.

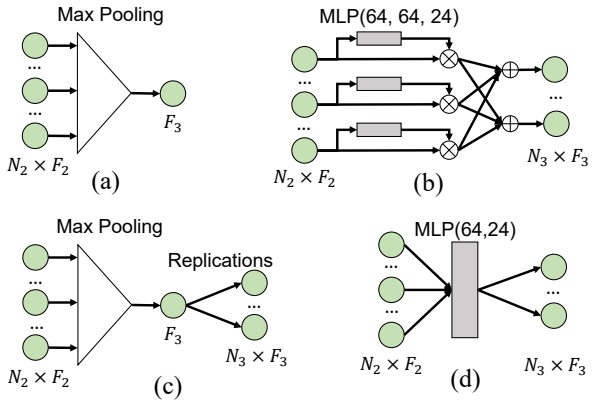

Figure 4. Feature pooling of PointNet++ (a), the proposed attention module (b) and other alternatives (c,d).

4.3. Skeleton Graph for Parameter Estimation

The purpose of this step is to regress the SMPL shape parameters β and pose parameter α. The pervious solutions [31, 16] directly predict SMPL parameters by MLP networks like Fig. 5 (a). However, their studies show that it is very hard to predict SMPL parameters even in fully supervised training. This is because SMPL shape and pose parameters interact in a nonlinear way. Shape parameters are used for joint predictions in the rest pose, which are further coupled with joint transformations derived from pose

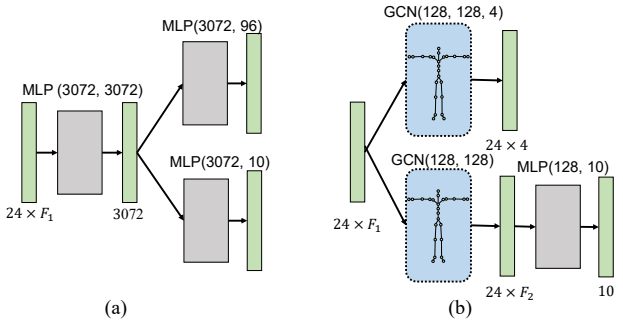

Figure 5. (a) MLP based SMPL parameter regression. (b) Proposed skeleton graph module based on the fast localized spectral convolution layer [17, 11], where GCN stands for graph convolution network.

parameters. Due to the skinning process, errors in pose or shape parameters will lead to large derivations in human shape, though pose parameters play the major role.

Thus, instead of MLP based direct regression, we propose to exploit the skeleton graph to incorporate the domain knowledge for better SMPL parameter regression. Particularly, we develop a skeleton graph module (SGM) (see Fig. 5 (b)), which takes features defined on the 24 skeleton joints as input and replicates them in two different branches for regressing shape and pose parameters, respectively. To capture the dependency among neighboring joints, the graph convolution described in Sec. 3.2 is applied to learn better joint features. The pose branch directly predicts four pose parameters on each joint by four layers of the graph convolution. The shape branch learns features on local joints with three layers of the graph convolution, followed by MLP to predict the 10 shape parameters. Note that Laplacian matrix L in SM is constructed with joints as nodes and $w_{ij} = 1$ for any two connected joints i and j.

4.4. Offline Training and Online Tuning

In order to train the entire network in a fully supervised manner, we need pairs of point clouds and the corresponding 3D ground truth meshes with the same number vertices ($N_4 = 6890$) and topology as SMPL meshes. It is extremely time-consuming and costly to construct such a large-scale training dataset. To avoid this dilemma, we resort to training on synthesized data. Specifically, we sampled a random set of shape and pose parameters as SURREAL [43], which are then fed into SMPL model to generate 3D human meshes. The input point clouds are generated by sampling 3D surface points on SMPL meshes.

With the constructed input and output pairs, we train the network with the following *supervised loss*:

$$L_{sup} = L_v + \lambda_{lap} \cdot L_{lap}, \tag{2}$$

where L_v is the vertex loss measuring the vertex distance, L_{lap} is the common Laplacian term to regularize / smooth

over-bent shapes (same as that defined in 3DCODED [13]), and λ_{lap} is a hyperparameter to balance the two loss terms. The vertex loss can be written as

$$L_v = \frac{1}{N_4}\sum_{i=1}^{N_4} ||\hat{\mathbf{v}}_i - \mathbf{v}_i||_2^2, \quad (3)$$

where $\{\hat{\mathbf{v}}_i\}^{N_4}$ is the reconstructed mesh and $\{\mathbf{v}_i\}^{N_4}$ is the corresponding ground truth mesh.

Although the trained network can perform well on synthesized data, it does not work well on real point clouds due to the domain gap. Thus, we introduce an online tuning phase, where we measure Chamfer distance between the input point cloud $\{\mathbf{p}_j\}^{N_1}$ and the reconstructed mesh $\{\hat{\mathbf{v}}_i\}^{N_4}$ and use it as an *unsupervised loss* to guide the online tuning:

$$L_{ch} = \sum_{i=1}^{N_4} \min_{j\in[1,N_1]} ||\hat{\mathbf{v}}_i - \mathbf{p}_j||_2^2 + \sum_{j=1}^{N_1} \min_{i\in[1,N_4]} ||\mathbf{p}_j - \hat{\mathbf{v}}_i||_2^2. \quad (4)$$

Training Details. During training, the whole network is optimized in a supervised way on synthesized dataset with Adam optimizer and an initial learning rate of 1×10^{-3} scheduled by ReduceOnPlateau method in PyTorch. During online tuning, we use a learning rate of 1×10^{-4} to tune parameters of SGM and 1×10^{-6} for other modules.

5. Experiments

In this part, we evaluated our framework and its individual modules on different datasets.

5.1. Datasets and Evaluation Metrics

Synthesized Dataset: Our network is trained on the synthesized dataset. We make use of SURREAL dataset [43] that provides SMPL shape and pose parameters for models captured in real scenarios, which enables generations of large numbers of human shapes with large variations and reasonable poses. Specifically, we directly sample the shape and pose parameters as SURREAL to generate training data with 5120 examples, validation data with 128 examples, and testing data with 1024 examples.

Dyna Dataset: We also evaluate our algorithm on Dyna dataset [32], which offers registered meshes with SMPL topology. The testing dataset is generated by randomly sampling 6890 points from each original mesh in two complex motion sequences 'jumping jacks' and 'running on spot'.

DFAUST dataset [6] provides raw scans of several persons in different motions. We evaluated our algorithm on all sequences in the dataset.

Berkeley MHAD dataset [26] provides two depth sequences from Kinect with human joint locations. We merged the depth images as one point cloud according to the provided camera parameters, and cropped out human regions by using bounding boxes spanned by human joints. The evaluation is conducted on two motion sequences.

Figure 6. Some test examples. In each separated column, point clouds are shown in left, while the ground truth are given in right. Note that an over-bent shape is shown in the middle.

In Fig. 6, we show some testing examples.

Evaluation Metrics: We consider the cases with and without ground truth meshes. For the synthesized testing dataset and Dyna dataset, ground truth meshes are known, and thus we calculate the average vertex-wise Euclidean distance from prediction to ground truth as in Eq. (3). Note that we use vertex-wise distance rather than vertex-to-surface distance as it can better reflect the distortion of reconstructed results. For DFAUST dataset and MHAD dataset, where ground truth meshes are unknown, we calculate the average point-to-vertex distance D_{p2v} (i.e. the first term in Eq. (4)) and the average vertex-to-point distance D_{v2p} (i.e. the second term in Eq. (4)). Note that we report mean values and maximal values over all test instances to show how averagely and badly an algorithm performs.

5.2. Ablation Study

Table 1 shows the mean and maximal average distances of our method and its variants on the synthesized dataset and Dyna dataset. Note that all methods yield large maximal mean mesh errors in the synthesized testing dataset. This is caused by some over-bent shapes as shown in Fig. 6.

To evaluate the influence of the GA module, we create a baseline named *Ours-GA* by replacing GA with the original PointNet++ module shown in Fig. 3(a). Comparing the results of *Ours-GA* and *Ours* in Table 1, we can conclude that incorporating the graph convolution into PointNet++ to facilitate point interactions is beneficial, boosting the performance by at least 4mm.

To evaluate the effect of the attention module (AM), we construct two baselines: *Ours-AM+POOL* and *Ours-AM+MLP* by replacing AM with the two alternative mod-

Table 1. Results of mean and maximum distances in mm of our method and its variants on the synthesized dataset and Dyna dataset.

	Synthesized		Jumping jacks		Running on spot	
Method	mean	max	mean	max	mean	max
Ours-GA	20.6	**356.4**	31.3	94.9	30.9	88.5
Ours-AM + MLP	36.0	445.0	40.3	81.1	42.7	83.1
Ours-AM + POOL	141.1	403.1	131.6	212.8	132.4	170.1
Ours-SGM +MLP	26.0	446.8	32.1	76.1	38.0	85.7
Ours-AM-SGM+POOL+MLP	29.5	462.4	41.0	107.8	35.7	83.7
Ours	**15.5**	423.1	**26.9**	**60.4**	**22.5**	**45.5**

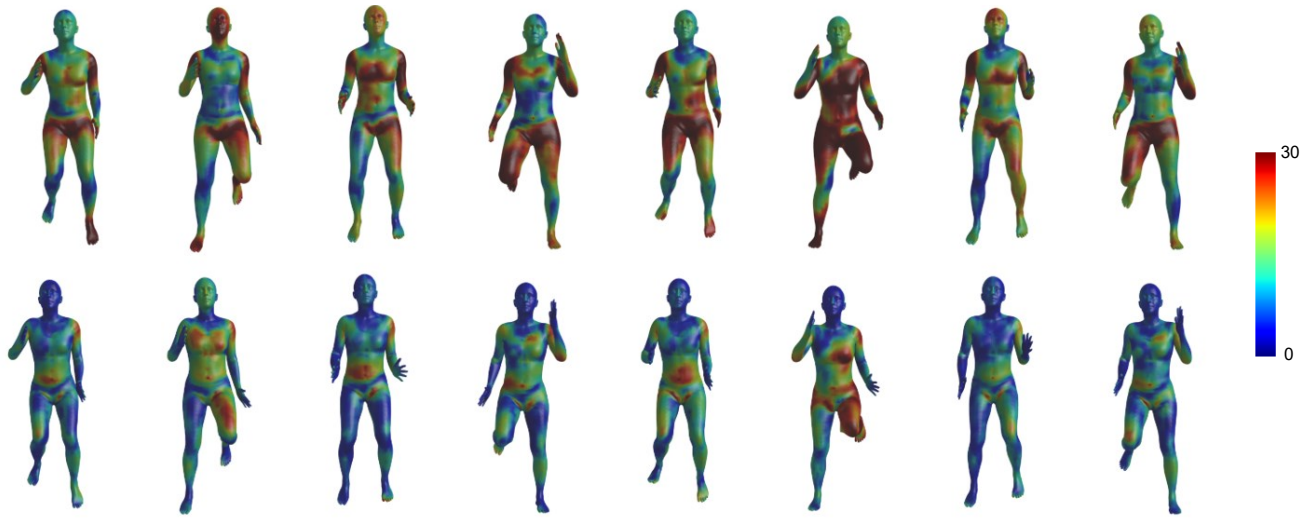

Figure 7. Reconstruction results on the motion sequence of 'running on spot' of Dyna dataset. Top: our results without fine tuning - *Ours (Initial)*. Bottom: our results with fine tuning - *Ours (Final)*. Mesh colors correspond to different reconstruction errors. Best viewed in color.

ules depicted in Fig. 4(c) and (d), respectively. It can be seen from Table 1 that the performance of *Ours-AM+POOL* is poor since the operations of pooling and replication of features give the same features for different joints. With a simple MLP mapping (Fig. 4(d)), *Ours-AM+MLP* achieves reasonable performance, but still has at least 13mm gap compared with *Ours*. This is because the mapping weight matrix of MLP is fixed once training is done, which prevents the network from capturing dynamic mapping relations for different permutations of inputs.

To evaluate the skeleton graph module (SGM), we construct a baseline named *Ours-SGM+MLP* by replacing SGM with a multi-layer perceptron, which is widely used in pose estimations [31, 16]. Comparing the results of *Ours-SGM+MLP* and *Ours*, we can see that SGM can improve the reconstruction accuracy by at least 5mm. Another choice for the system is to directly predict SMPL parameters rather than using an attention module and a skeleton graph module. This experiment is conducted by replacing both AM and SGM with a pooling operation and MLP (denoted as *Ours-AM-SM+POOL+MLP* in Tab. 1). We can see that this leads to worse results (at least 13mm drops).

At last, we evaluated the proposed online tuning scheme in the last row of Table 2. Comparing the results of without and with online fine tuning, denoted respectively as *Ours (Initial)* and *Ours (Final)*, we can see that the online tuning can greatly improve the performance, by adapting to the new data on Dyna dataset, which is different from the training data from the synthesized dataset. Some visual results are provided in Fig. 7, which further demonstrates the combination of offline training and online tuning makes the network capable of adapting to new domain.

5.3. Comparisons to the State-of-the-art

Table 2. Comparisons with the state-of-the-art methods on Dyna dataset. The network prediction results are denoted as *Initial*, and the final tuning or optimized results are denoted as *Final*.

	Jumping jacks				Running on spots			
	Initial		Final		Initial		Final	
Method	mean	max	mean	max	mean	max	mean	max
3DCODED [13]-syn	41.6	89.2	20.7	53.5	38.1	220.1	16.7	228.1
3DCODED [13]-author	32.5	114.8	17.2	109.6	24.7	290.5	11.8	298.2
SMPLify [5]-mesh	-	-	**8.7**	**12.2**	-	-	**10.0**	**14.2**
SMPLify [5]-pcd	-	-	42.5	340.1	-	-	88.8	406.7
Ours	**26.9**	**60.4**	**14.2**	**67.4**	**22.5**	**45.5**	**11.4**	**32.1**

We compared our method with two state-of-the-art approaches, i.e. 3DCODED [13] and SMPLify method [5]. For 3DCODED [13], we directly use the authors' released code for comparison. We evaluate the model trained on our training dataset (denoted as *3DCODED-syn*) and the authors' pretrained model (denoted as *3DCODED-author*), which uses much more data for training (around 200*k*). Note that 3DCODED also uses a trained network to produce an initial reconstruction (*Initial*) and then optimizes its representation to obtain a better reconstruction (*Final*). Due to the difficulty to directly optimize the SMPL parameters using SMPL model, Bogo et al. [5] proposed several important pose priors to prevent over-bent shapes and achieve successful reconstruction. So we compare with the method [5] instead of directly optimizing with SMPL model. The code of [5] is also available but is quite slow. Thus, we re-implemented it on GPU to speed up the process. We consider two versions of the SMPLify method [5]: *SMPLify-mesh* and *SMPLify-pcd*, where the former is to optimize the SMPL parameters so as to make the reconstructed mesh close to a given SMPL ground truth mesh and the latter is to

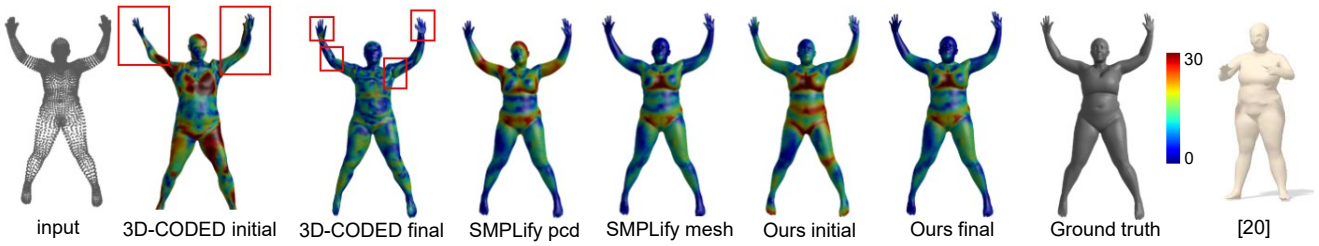

Figure 8. Visual comparisons of different methods. Note that the result of [20] is a figure directly adapted from the original paper, which is not corresponding to the input here. It is mainly to show that the mesh produced by [20] is of limited quality. Best viewed in color.

fit the reconstructed mesh to a given point cloud. Note that *SMPLify-mesh* is in fact the upper bound, indicating the best that SMPL model can produce given the ground-truth mesh with the same topology.

Tab. 2 shows the comparison results on Dyna dataset. We can see that *SMPLify-mesh* can fit the ground-truth mesh quite well, but *SMPLify-pcd* performs poorly in fitting point clouds where no correspondence is given. Our methods outperform 3DCODED in both the initial network output and the final reconstruction. Fig. 8 gives the visual comparisons on one example. 3D-CODED [13] can nicely reconstruct human pose and rough shape, but its results may suffer from rugged or over-bent meshes as shown in Fig. 8. We also show the limited quality of the reconstruction results of [20] by copying a figure from [20].

Other results. We also applied our framework to raw scans for human completion and reconstruction. Comparison results on DFAUST dataset are shown in Table 3, where our method is better than 3DCODED [13] trained with the same data and SMPLify [5], while comparable to 3DCODED [13] trained with more data. We also test our method on point clouds generated from two depth images of Berkeley MHAD dataset and the results are shown in Table 4. A visual comparison in Fig. 9 shows that our method achieves a more desirable reconstruction result than others.

Table 3. Comparisons of the distance results in mm on all sequences of DFAUST dataset. Note that in each cell, the first and second numbers denote the distances D_{p2v} and D_{v2p}, respectively.

	3DCODED [13]-syn	3DCODED [13]-author	SMPLify [5]-pcd	Ours
mean	11.5/16.9	7.0/12.5	25.4/31.9	8.1/12.6
max	617.4/380.9	564.2/215.7	296.2/229.4	127.5/102.1

Table 4. Comparison results on two sequences of Berkeley MHAD dataset.

Method	mean (seq1)	max (seq1)	mean (seq2)	max (seq2)
3DCODED [13]-syn	22.3/26.6	36.0/38.5	18.9/21.9	32.1/32.9
3DCODED [13]-author	18.8/20.3	23.9/27.8	16.6/17.8	22.3/25.2
SMPLify [5]-pcd	31.1/41.1	43.3/58.8	31.3/39.7	48.6/58.4
Ours	21.4/23.5	28.6/34.7	16.9/18.2	21.5/21.2

Limitations. In our experiments, we observed our method has relatively large errors in female shape reconstruction,

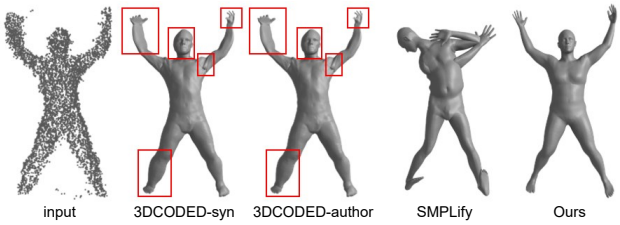

Figure 9. A visual comparison on MHAD.

especially in chest, belly, and hip part (see Fig. 8). We conjecture this is inherited from the SMPL representation, since *SMPLify-mesh* exhibits similar problems. The possible reason may be that SMPL model only uses 10 parameters for shapes, which makes it hard to model body parts with a larger derivation from neutral shapes. Another major limitation is that our method is restricted to SMPL model and can only reconstruct naked human shapes.

6. Conclusion

In this paper, we have presented an end-to-end learning framework for 3D human shape reconstruction from point clouds. The main technical contributions include (1) introducing a graph aggregation module to augment PointNet++ by extracting better point features; (2) proposing an attention module to better map unordered point features into ordered skeleton joint features; and (3) designing a skeleton graph module to extract better joint features for SMPL parameter prediction. The experimental results have demonstrated that the proposed modules can significantly boost the reconstruction accuracy. This work could lead to many other future studies. For example, it is interesting to see whether PointNet++ with GA can perform better in other point cloud tasks such as segmentation and classification. We can also make use of the extracted joint features in SGM for estimating 3D joint locations.

Acknowledgments. HY would like to thank Yanbo Fan and Boyi Jiang for helpful discussions. This research is mainly supported by MoE Tier-2 Grant (2016-T2-2-065) and partially supported by a grant (M4082186) for Joint WASP/NTU and the National Natural Science Foundation of China (61620106003).

References

[1] Thiemo Alldieck, Marcus A. Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. *CoRR*, abs/1903.05885, 2019.

[2] Thiemo Alldieck, Marcus A. Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8387–8397, 2018.

[3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.

[4] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2300–2308, 2015.

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter V. Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: automatic estimation of 3d human pose and shape from a single image. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 561–578, 2016.

[6] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: registering human bodies in motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5573–5582, 2017.

[7] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[8] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *2019 IEEE International Conference on Computer Vision, ICCV 2019*, 2019.

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1302–1310, 2017.

[10] Ke-Li Cheng, Ruofeng Tong, Min Tang, Jing-Ye Qian, and Michel Sarkis. Parametric human body reconstruction based on sparse key points. *IEEE Trans. Vis. Comput. Graph.*, 22(11):2467–2479, 2016.

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

[12] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *2019 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, 2019.

[13] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, pages 235–251, 2018.

[14] Marc Habermann, Weipeng Xu, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt. Reticam: Real-time human performance capture from monocular video. *CoRR*, abs/1810.02648, 2018.

[15] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8320–8329, 2018.

[16] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7122–7131, 2018.

[17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[18] Roman Klokov and Victor S. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 863–872, 2017.

[19] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating LSTM: 3d pose estimation based on joint interdependency. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 123–141, 2018.

[20] Or Litany, Alexander M. Bronstein, Michael M. Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1886–1895, 2018.

[21] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015.

[22] Riccardo Marin, Simone Melzi, Emanuele Rodolà, and Umberto Castellani. FARM: functional automatic registration method for 3d human bodies. *CoRR*, abs/1807.10517, 2018.

[23] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5425–5434, 2017.

[24] Federico Monti, Michael M. Bronstein, and Xavier Bresson. Deep geometric matrix completion: A new way for recommender systems. In *2018 IEEE International Conference*

on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018, pages 6852–6856, 2018.

[25] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2014–2023, 2016.

[26] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley MHAD: A comprehensive multimodal human action database. In *2013 IEEE Workshop on Applications of Computer Vision, WACV 2013, Clearwater Beach, FL, USA, January 15-17, 2013*, pages 53–60, 2013.

[27] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. 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 2018, Verona, Italy, September 5-8, 2018*, pages 484–494, 2018.

[28] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. *CoRR*, abs/1904.05866, 2019.

[29] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1263–1272, 2017.

[30] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1253–1262, 2017.

[31] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 459–468, 2018.

[32] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: a model of dynamic human shape in motion. *ACM Trans. Graph.*, 34(4):120:1–120:14, 2015.

[33] Gerard Pons-Moll, Jonathan Taylor, Jamie Shotton, Aaron Hertzmann, and Andrew W. Fitzgibbon. Metric regression forests for correspondence estimation. *International Journal of Computer Vision*, 113(3):163–175, 2015.

[34] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. pages 77–85, 2017.

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5105–5114, 2017.

[36] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, pages 725–741, 2018.

[37] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 765–782, 2018.

[38] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8437–8446, 2018.

[39] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2530–2539, 2018.

[40] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew W. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 103–110, 2012.

[41] Kiran Koshy Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. Attention-based graph neural network for semi-supervised learning. *CoRR*, abs/1803.03735, 2018.

[42] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 20–38, 2018.

[43] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4627–4635, 2017.

[44] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2598–2606, 2018.

[45] Timo von Marcard, Bodo Rosenhahn, Michael J. Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Comput. Graph. Forum*, 36(2):349–360, 2017.

[46] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *CoRR*, abs/1801.07829, 2018.

[47] Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1544–1553, 2016.

[48] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Trans. Graph.*, 37(2):27:1–27:15, 2018.

[49] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 7444–7452, 2018.

[50] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 206–215, 2018.

[51] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 910–919, 2017.

[52] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7287–7296, 2018.

[53] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. Deep network for the integrated 3d sensing of multiple people in natural images. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 8420–8429, 2018.

[54] Chao Zhang, Sergi Pujades, Michael J. Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5484–5493, 2017.

[55] Zerong Zheng, Tao Yu, Hao Li, Kaiwen Guo, Qionghai Dai, Lu Fang, and Yebin Liu. Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IX*, pages 389–406, 2018.

[56] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *CoRR*, abs/1812.07035, 2018.