Sequential 3D Human Pose and Shape Estimation from Point Clouds

Kangkan Wang*[1,2], Jin Xie[1,2], Guofeng Zhang[3,4], Lei Liu[1] , Jian Yang[1,2]
[1]Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education,
[2]Jiangsu Key Lab of Image and Video Understanding for Social Security,
School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[3]State Key Laboratory of CAD&CG, Zhejiang University, China
[4]ZJU-SenseTime Joint Lab of 3D Vision

Abstract

This work addresses the problem of 3D human pose and shape estimation from a sequence of point clouds. Existing sequential 3D human shape estimation methods mainly focus on the template model fitting from a sequence of depth images or the parametric model regression from a sequence of RGB images. In this paper, we propose a novel sequential 3D human pose and shape estimation framework from a sequence of point clouds. Specifically, the proposed framework can regress 3D coordinates of mesh vertices at different resolutions from the latent features of point clouds. Based on the estimated 3D coordinates and features at the low resolution, we develop a spatial-temporal mesh attention convolution (MAC) to predict the 3D coordinates of mesh vertices at the high resolution. By assigning specific attentional weights to different neighboring points in the spatial and temporal domains, our spatial-temporal MAC can capture structured spatial and temporal features of point clouds. We further generalize our framework to the real data of human bodies with a weakly supervised fine-tuning method. The experimental results on SURREAL, Human3.6M, DFAUST and the real detailed data demonstrate that the proposed approach can accurately recover the 3D body model sequence from a sequence of point clouds.

1. Introduction

Recovering 3D human shapes has numerous real-world applications in robotics, augmented reality (AR) and virtual reality (VR). Particularly, with the recent advancement of depth sensors such as Microsoft Kinect, 3D human shape estimation from depth images has gained popularity in the 3D computer vision community. 3D human shape estimation from depth images aims to recover 3D meshes of human bodies [26, 8, 44, 29]. However, accurately estimating 3D human body shapes from depth images is highly challenging since there are arbitrary deformations and self-occlusions with human bodies. In addition, viewpoint changes of depth cameras and severe random noises on depth images make the problem more difficult.

Most of 3D body shape estimation methods from depth images [16, 12, 11, 5, 41, 42] mainly focus on utilizing temporal information to build point correspondences between consecutive frames and recover the 3D model of each frame with the correspondences. In the case of large discrepancy between the 3D template model and an input depth image, it is difficult to establish correct correspondences using the nearest neighboring search method. Thus, these methods are not effective to recover the 3D body model from a single depth image. In addition, since these methods sequentially recover 3D human body models from depth images, the correspondence errors are accumulated over the sequences. In [38], the local descriptors of 3D human shapes are learned to construct the dense correspondences and the 3D models are then recovered by fitting the template model to input depths. Nonetheless, these works mainly focus on recovering 3D human body models from a sequence of depth images. Few efforts are made on the 3D body model recovery from a sequence of point clouds.

Point clouds can provide more geometry information than depth images. It is desirable to recover the 3D human body models from a sequence of point clouds so that 3D geometric structures of point clouds can be exploited. We aim to directly infer sequential 3D body models by extracting local features of a sequence of point clouds. Thus, we do not need to construct point correspondences between consecutive frames and avoid the error accumulation during the process of sequential 3D body model recovery.

Instead of estimating the SMPL [25] model parameters of human bodies, in this paper, we propose a sequential 3D human shape estimation method from point clouds by predicting the vertex coordinates of multi-resolution 3D body meshes. First, we employ PointNet++ [31] to extract the latent features of point clouds of sequential frames separately.

*Corresponding author: wangkangkan@njust.edu.cn

We then develop a spatial-temporal mesh attention convolution to regress vertex coordinates of the 3D body meshes at different resolutions from the latent features. Based on the generated mesh at the low resolution, we construct a spatial mesh attention convolution (MAC) by dynamically assigning specific attentional weights to the one-ring neighborhoods of the mesh vertices on the current frame. Similarly, we also construct a temporal MAC by assigning attentional weights to the corresponding mesh vertices on the consecutive frames. The spatial MAC can capture local structured features of point clouds in the spatial domain while the temporal MAC can fuse structured features of point clouds on sequential frames to form a temporal representation. In addition, our method can be generalized well to the real data captured by depth sensors with a weakly-supervised fine-tuning method. The experimental results on SURREAL [37], Human3.6M [17], DFAUST [7] and the real data of human bodies demonstrate the effectiveness of the proposed method. In summary, the main contributions of our method are as follows:

- We innovatively formulate the problem of 3D human pose and shape estimation from a sequence of point clouds.
- We propose a spatial-temporal mesh attention convolution to progressively regress the vertex coordinates of the 3D human meshes.
- We propose a weakly-supervised fine-tuning algorithm for the 3D body model recovery of the real human bodies with detailed surfaces.

2. Related Work

3D human body modeling from depth images. The existing methods of 3D human body modeling from depth images can be roughly divided into template-based and template-less methods. The template-based methods utilize template priors for the 3D body model recovery such as embedded skeleton [39, 40], template models [11], or parametric models [5, 44, 29]. These template priors encapsulate much prior knowledge of the template, thus making the 3D body reconstruction robust. For example, Guo et al. [11] deforms a pre-scanned template model to each input depth through a novel L_0 based motion regularizer which effectively reduces the accumulated error in large motions. Template-less methods [26, 16, 9, 12, 8] create the 3D body models without any prior knowledge about the body shape. These methods volumetrically fuse all captured depth maps to reconstruct 3D models in realtime, but they are restricted to slow motions. Recently, some approaches [41, 42, 47] extend to deal with large human motions by incorporating template priors into template-less methods. For both template-based and template-less methods, it is required to build point correspondences for each frame through the closest 3D point searching method. The built correspondences are prone to be inaccurate in the case of a single input depth due to large discrepancy of human poses and shapes between the template and the depth. Point correspondences can be predicted directly for depth images of human bodies through a random forest [30] or by matching learned feature descriptors [38]. Based on the predicted point correspondences, 3D body models are then recovered by deforming the template to depth data. LBS Autoencoder [23] fits articulated mesh models to point clouds by inferring the joint angles and deformation of a LBS template, which is mainly proposed for point clouds with complete 3D shape but not point clouds of depth images.

3D human pose and shape from color images. Most methods of 3D body shape estimation from color images fit a parametric body model [2, 25] or a template model [13] to a set of observations on the input color images, such as keypoints and silhouettes. For example, Bogo et al. [6] first detects 2D body joints and then fits the SMPL model [25] to these 2D joint locations. Many deep learning based methods [35, 27, 22, 36] infers the 3D body shape directly from color images using convolutional networks. DensePose [10] estimates dense human pose by learning dense correspondences between an RGB image and a template body model. Kanazawa et al. [18] infers the SMPL parameters through an iterative 3D regressor from the latent features on a single RGB image. A Graph CNN method [21] first attaches the extracted features from an input color image to 3D vertex coordinates of a template mesh and then predicts the vertex coordinates of 3D body meshes using a convolutional mesh regression. Recent works [49, 48, 1, 33] attempt to recover the surface details of 3D human shapes beyond the parametric model from color images. For example, Zhu et al. [49] proposes a hierarchical mesh deformation framework to restore detailed body shapes by utilizing body joints, silhouettes, and per-pixel shading information.

3D human pose and shape from videos. In recent years, feed-forward convolutional networks are often adopted to encode temporal features from image sequences in 3D human pose and shape estimation [19, 45]. Recent studies [4] suggest that feed-forward convolutional models not only perform more accurately than canonical recurrent architectures [14, 24] on a broad range of sequence modeling tasks, but also are simpler and easier to train. Kanazawa et al. [19] learns to capture 3D human dynamics from video by using a temporal convolutional network which reduces uncertainty and jitter in 3D prediction of single-view approaches [18]. Zhang et al. [45] refines the temporal convolutional network of [19] with a causal structure so that only past temporal context is used to predict 3D human motion from video. Some works [15, 35, 43, 46, 3] recover the sequential 3D models by enforcing temporal consistency across

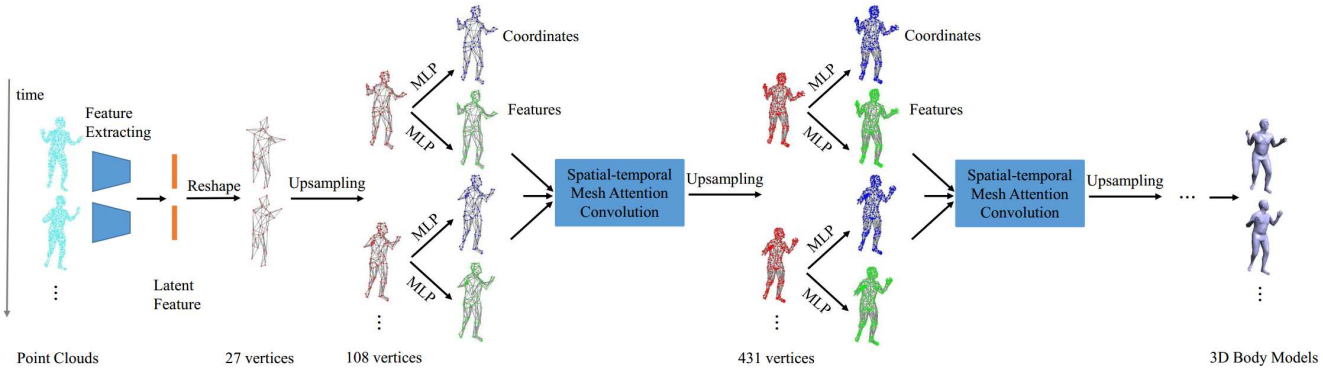

Figure 1. Overview of the proposed framework. Our framework can predict the 3D body model sequence from an input sequence of point clouds of a human body. The proposed spatial-temporal mesh attention convolution regresses 3D coordinates of mesh vertices at different resolutions by capturing structured spatial and temporal features of point clouds. Please refer to Sect. 3 for detailed description.

consecutive frames. Tung et al. [35] maps a color image sequence to a sequence of corresponding 3D meshes based on the consistency constraint that 3D motion of mesh vertices when projected should match 2D optical flow. Zanfir et al. [43] extends the single image method to video by imposing temporally coherent pose and motion reconstructions. The bundle adjustment algorithm proposed in [3] can jointly optimize the per-frame estimations of [18] over the whole video, and exploit temporal consistency of model parameters to resolve ambiguities. In this work, we use feed-forward convolutional networks for 3D body model estimation by proposing a temporal encoder on mesh structure.

3. Proposed Approach

Given a sequence of point clouds of a human body, our goal is to estimate the sequential 3D body models that can fit to the corresponding point clouds in the sequence. The framework of our proposed method is illustrated in Fig. 1. In the framework, we first extract local features of the point clouds of each frame independently. Then, based on the extracted features, a spatial-temporal mesh attention convolution (MAC) network is proposed to predict the 3D body mesh of each frame from coarse to fine. The spatial-temporal MAC can exploit both spatial and temporal features encoded in the coarse meshes to infer the finer meshes at the high resolution. In addition, we generalize our method to real point clouds of human bodies by a weakly-supervised fine-tuning method.

3D human body model. We adopt Skinned Multi-Person Linear model (SMPL) [25] as the 3D human body model. SMPL is a widely used statistical model which can generate various 3D body models with natural human shapes and poses. Based on a set of shape and pose parameters, the SMPL model can output a 3D body model with $N = 6,890$ vertices. Please refer to [25] for more details. In our method, instead of estimating the model parameters, we directly predict the 3D coordinates of the model vertices that can fit to the input point clouds accurately. From the predicted model vertices, we can easily obtain the SMPL parametric models through a model fitting method [25].

3.1. Spatial-temporal mesh attention convolution

Since PointNet++ [31] can characterize geometric structures of point clouds well, we employ PointNet++ to extract the latent features of point clouds of each frame. We then develop a spatial-temporal MAC network to generate multi-resolution 3D body meshes from the extracted features. As shown in Fig. 1, at the high resolution, we first upsample the coarse mesh from the previous low resolution and then employ the spatial-temporal MAC to generate the finer mesh with a fixed topology. The mesh upsampling is performed by left-multiplying the upsampling matrix with the mesh. We pre-compute the upsampling matrices of the SMPL template model at different resolutions as [32]. The proposed spatial-temporal MAC can capture the structured features of point clouds stored on the mesh vertices in both the spatial and temporal domains. The spatial MAC dynamically assigns specific attentional weights to the one-ring neighborhoods of the mesh vertices on a single frame while the temporal MAC assigns attentional weights to the corresponding mesh vertices on the consecutive frames. We apply the spatial-temporal MAC in a coarse-to-fine manner to generate the final 3D human body model.

3.1.1 Spatial mesh attention convolution

At the frame l, we first employ the upsampling operation to generate a high-resolution mesh $\boldsymbol{G}_k$ with q vertices at the k-th resolution. For conciseness, we ignore the frame stamp l in the following derivations of this subsection. We then regress the vertex coordinates of the mesh $\boldsymbol{G}_k$ from the input features of q vertices through a MLP network. We formulate the following 3D coordinate loss to generate the vertex coordinates of the high-resolution mesh:

$$\mathbf{L}_{coord}(k) = \sum_{i=1}^{q} \parallel \boldsymbol{p}_i - \tilde{\boldsymbol{p}}_i \parallel_2^2, \tag{1}$$

where $\boldsymbol{p}_i$ is the coordinates of vertex i on the generated mesh $\boldsymbol{G}_k$, and $\tilde{\boldsymbol{p}}_i$ is the ground truth coordinates of vertex i on $\tilde{\boldsymbol{G}}_k$. The mesh $\tilde{\boldsymbol{G}}_k$ is downsampled from the full ground truth model with the downsampling operation [32]. We also map the input features of mesh vertices into a new set of vertex features $\boldsymbol{h} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_q\} (\boldsymbol{h}_i \in \mathbb{R}^F)$ with a MLP layer, where F is the feature dimension of each vertex.

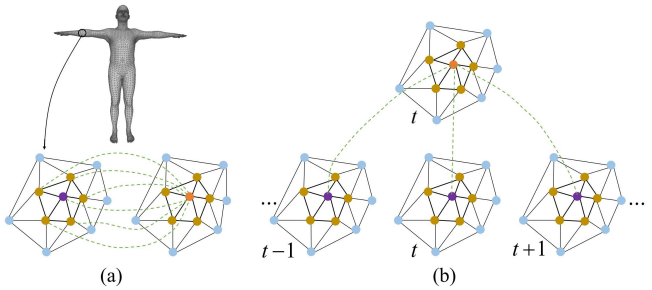

Figure 2. An illustration of our spatial mesh attention convolution (a) and temporal mesh attention convolution (b). Spatial mesh attention convolution is performed on the one-ring neighborhoods of the mesh vertices on a single frame, while temporal mesh attention convolution is performed on the corresponding mesh vertices of consecutive frames.

With the attention mechanism [28], we construct a spatial MAC to capture local structured features of the one-ring neighborhoods of the vertices on the generated mesh $\boldsymbol{G}_k$. In the constructed spatial MAC, different one-ring neighborhoods of the mesh vertices are assigned to specific attentional weights. The attentional weights of each vertex with its neighbors are related to the differences of vertex coordinate and feature vector, which are computed as follows:

$$\boldsymbol{\sigma}_{ij} = \boldsymbol{\sigma}([\Delta \boldsymbol{p}_{ij}, \Delta \boldsymbol{h}_{ij}]), j \in \mathcal{N}(i), \tag{2}$$

where $\Delta \boldsymbol{p}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i$, $\Delta \boldsymbol{h}_{ij} = \boldsymbol{h}_j - \boldsymbol{h}_i$, and $\mathcal{N}(i)$ is the neighbor set of vertex i (including itself). By taking the concatenation of vertex coordinate and feature vector differences as the input, the spatial attentional weights are learned by the attention mechanism $\boldsymbol{\sigma}$, which is a MLP network in our experiments. To handle the neighbors across different vertices and spatial scales, the attentional weights are normalized across all the neighbors of vertex i as follows:

$$\tilde{\boldsymbol{\sigma}}_{ij} = softmax(\boldsymbol{\sigma}_{ij}) = \frac{exp(\boldsymbol{\sigma}_{ij})}{\sum_{s \in \mathcal{N}(i)} exp(\boldsymbol{\sigma}_{is})}, \tag{3}$$

where $\boldsymbol{\sigma}_{ij}$ is the attentional weight vector of vertex j to vertex i. The final output features of vertex i after spatial MAC can be computed by a linear combination of the neighbor features with the normalized attentional weights:

$$\tilde{\boldsymbol{h}}_i = \sum_{j \in \mathcal{N}(i)} \tilde{\boldsymbol{\sigma}}_{ij} \boldsymbol{h}_j + \boldsymbol{b}_i, \tag{4}$$

where $\boldsymbol{b}_i \in \mathbb{R}^F$ is a learnable bias.

3.1.2 Temporal mesh attention convolution

Temporal context provided by the sequential frames can alleviate the problems of occlusion uncertainty and shape ambiguity in single-view approaches. Thus, it is necessary to encode the temporal features and use them in the recovery of sequential 3D models. The mesh vertices store features for vertex coordinate regression which encodes the information of the 3D body shape. These features can provide much useful information for the 3D model estimation of other frames in a sequence. Since the sequential frames have the corresponding mesh topology at the same resolution, we can temporally fuse the features of the same mesh vertex across all the frames. We apply a temporal mesh attention convolution for mesh vertices to exploit useful information from consecutive frames. Specifically, for a vertex i on the mesh $\boldsymbol{G}_k$ of frame l, we compute the attentional weight of frame j to frame l on vertex i as follows:

$$\boldsymbol{\varepsilon}^i_{lj} = \boldsymbol{\varepsilon}(\tilde{\boldsymbol{h}}^i_j - \tilde{\boldsymbol{h}}^i_l), j \in \{1, 2, ...fn\}, \tag{5}$$

where fn is the frame number of a sequence sample, $\tilde{\boldsymbol{h}}^i_j$ and $\tilde{\boldsymbol{h}}^i_l$ are the i-th vertex features after spatial MAC on the mesh $\boldsymbol{G}_k$ of frame j and l, respectively. The temporal attention mechanism $\boldsymbol{\varepsilon}$ maps the feature vector difference to the temporal attentional weights, which is a MLP network in our experiments. The temporal attentional weights are also normalized across all the frames as follows:

$$\tilde{\boldsymbol{\varepsilon}}^i_{lj} = \frac{exp(\boldsymbol{\varepsilon}^i_{lj})}{\sum_{t=1}^{fn} exp(\boldsymbol{\varepsilon}^i_{lt})}. \tag{6}$$

The final output features of vertex i in frame l after temporal MAC can be computed as follows:

$$\hat{\boldsymbol{h}}^i_l = \sum_{j=1}^{fn} \tilde{\boldsymbol{\varepsilon}}^i_{lj} \tilde{\boldsymbol{h}}^i_j + \boldsymbol{b}^i_l, \tag{7}$$

where $\boldsymbol{b}^i_l \in \mathbb{R}^F$ is a learnable bias. Through $K = 4$ different resolutions shown in Fig. 1, the mesh is upsampled to $1,723$ vertices. We finally apply another spatial MAC to map the vertex features to 3D vertex coordinates. The mesh regression loss is defined as follows:

$$\mathbf{L}_{mesh} = \sum_{i=1}^{N} \parallel \boldsymbol{v}_i - \tilde{\boldsymbol{v}}_i \parallel_2^2, \tag{8}$$

where $\tilde{\boldsymbol{v}}_i$ is the coordinates for vertex i of frame l on the ground truth model. To avoid the high vertex redundancy and reduce the training time, we predict the 3D models with $1,723$ vertices that is a factor of 4 downsampled on the original SMPL vertices. The 3D models of original scale can be easily obtained through a mesh upsampling [32] on the

predicted 3D models. Then, the overall loss function of our method is defined as:

$$\mathbf{L} = \sum_{l=1}^{fn}(\mathbf{L}_{mesh} + \lambda \sum_{k=1}^{K} \mathbf{L}_{coord}(k)), \quad (9)$$

where λ is the regularization parameter.

3.2. Weakly-supervised fine-tuning for real detailed data

Since there are no ground truth 3D models for real bodies with details such as clothes, we fine-tune the network on this kind of real data in a weakly-supervised manner. By testing on real point clouds using the pre-trained models, we can obtain 3D models $\mathcal{V}$ which are roughly consistent with the input point clouds in poses and shapes. Although the predicted 3D models $\mathcal{V}$ does not fit the input point clouds well, we can use them to supervise the fine-tuning on real clothed data. We define the mesh regression loss in the fine-tuning network as follows:

$$\boldsymbol{L}'_{mesh} = \boldsymbol{L}_{3D} + \beta \boldsymbol{L}_{Laplacian} + \gamma \boldsymbol{L}_{edge}, \quad (10)$$

where $\boldsymbol{L}_{3D}$ is the 3D correspondence loss, $\boldsymbol{L}_{Laplacian}$ is the Laplacian loss, $\boldsymbol{L}_{edge}$ is the edge loss, β and γ are the regularization parameters. The 3D correspondence loss forces the vertices of the estimated models to align to the corresponding points on the point clouds, defined as follows:

$$\mathbf{L}_{3D} = \frac{1}{N_c} \sum_{i=1}^{N} m_i \parallel \boldsymbol{v}_i - \boldsymbol{p}_i \parallel_2^2, \quad (11)$$

where $\boldsymbol{v}_i$ is the i-th vertex on the estimated 3D models, $\boldsymbol{p}_i$ is the corresponding point of $\boldsymbol{v}_i$ on the input point clouds, N_c is the number of valid correspondences, and m_i is 0 or 1 (if it is a valid correspondence, $m_i = 1$; otherwise, $m_i = 0$). The point correspondences are initially built based on $\mathcal{V}$ and updated iteratively during the fine-tuning process. Since the 3D correspondence loss only constrains the visible vertices of the body models, the occluded body parts are prone to be recovered with unnatural shapes. Thus, we introduce the Laplacian loss [34] to preserve the surface smoothness:

$$\mathbf{L}_{Laplacian} = \sum_{i=1}^{N} \parallel \boldsymbol{\delta}_i - \hat{\boldsymbol{\delta}}_i \parallel_2^2, \quad (12)$$

where $\boldsymbol{\delta}_i = \boldsymbol{v}_i - \frac{1}{N_i} \sum_{j \in \mathcal{N}(i)} \boldsymbol{v}_j$ is the Laplacian coordinate of vertex i on the estimated 3D models, $\mathcal{N}(i)$ is the neighbor set of vertex i, N_i is the number of vertices in the set $\mathcal{N}(i)$, and $\hat{\boldsymbol{\delta}}_i$ is the Laplacian coordinate of vertex i on $\mathcal{V}$. In addition, we apply the edge loss [2] to penalize unnatural edges and enforce edge length consistency of 3D models:

$$\mathbf{L}_{edge} = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} (\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 - \|\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{v}}_j\|_2^2)^2, \quad (13)$$

where $\boldsymbol{v}_i$ and $\hat{\boldsymbol{v}}_i$ denote vertex i on the estimate models and $\mathcal{V}$, respectively. We formulate the 3D coordinate loss $\mathbf{L}'_{coord}(k)$ of the estimated meshes at different resolutions using the same three losses defined in Eq. 10 with different mesh topologies. By optimizing the total fine-tuning objective $\mathbf{L}' = \sum_{l=1}^{fn}(\mathbf{L}'_{mesh} + \lambda \sum_{k=1}^{K} \mathbf{L}'_{coord}(k))$, the estimated 3D models can be registered to the input point clouds.

4. Experiments

In this section, we first elaborate our implementation details and then evaluate our method by comparing with the state-of-the-art methods. Finally, we perform ablative analysis on our method and test it on the real data.

4.1. Implementation details

Datasets. We conduct experiments on the SURREAL [37], Human3.6M [17], DFAUST [7], and real data. The training dataset of SURREAL contains $55,001$ clips of 3D body models, each clip with mostly 100 frames long. We uniformly sample $10,000$ clips to generate $200,000$ subsequences with 5 frames long both for male and female as the training data. DFAUST dataset [7] contains more than $40,000$ registered scans of real undressed bodies, which have shapes and motions beyond the SMPL model. Since there are the corresponding SMPL models for DFAUST dataset, we generate $50,000$ subsequences for both men and women as the training data. We render the 3D models to depth images from different views and make the resolution of the rendered images nearly the same as real data to simulate real depths. The rendered depths are finally converted to point clouds for the training. For SURREAL, Human3.6M and DFAUST, we uniformly sample 100 sequences with about 100 frames long for both male and female as the test data. Note that the test data does not include any same subject in the training data. We have captured a small real dataset with a Kinect V2 sensor which contains more than 8100 frames with different subjects under various motions. Our network is fine-tuned with Eq. 10 on samples from SURREAL, DFAUST and the real captured data.

Architecture and experimental settings. The raw point clouds of depth images are uniformly downsampled to $L = 2,500$ points in our experiments. We use the original PointNet++ [31] to extract local features on the point clouds. The extracted $1,024$-dim feature vector is first transformed to $(27 * 256)$-dim vector with a fully-connected layer and then reshaped to 27 vertices with 256-dim feature vector. Our spatial-temporal MAC networks consist of four different mesh resolutions with upsampling factors of $\{4, 4, 2, 2\}$, finally outputting 3D coordinates of $1,723$ vertices. The feature channels at different resolutions are $\{256, 128, 64, 32\}$, respectively. We use $mlp\{256, 256, 3\}$ for the vertex coordinate regressor. For both spatial and temporal attention mechanism, we first learn the attentional weights with

$mlp\{16, 16\}$ and then map them to the weights with the same size as the input features using a followed MLP layer. Each mesh convolution is followed by a ReLu layer except the last one of regressing the 3D coordinates. The learning rate is set to 1×10^{-4}. The length of each training sample is 5 frames long. We try to use training samples with larger frame length, but the training takes much longer time. By using the length of 5 frames, our method can yield a good accuracy of reconstruction. We use Adam optimizer [20] with the batch size of 8. We empirically set $\lambda = 1$, $\beta = 60$, and $\gamma = 100$. The running time for a test sample is about 24.7ms on average with a NVIDIA 2080 Ti GPU.

Error metrics. The compared methods are evaluated through both quantitative and qualitative experiments. We quantify reconstruction error with the Mean Average Vertex Error (MAVE) over all vertices of all recovered 3D models in millimeter (mm):

$$\epsilon = \frac{1}{N_f} \sum_{k=1}^{N_f} \frac{1}{N} \sum_{i=1}^{N} \sqrt{\|(v_i - \hat{v}_i)\|_2^2}, \tag{14}$$

where N_f is the number of test samples, v_i is the i-th vertex on the recovered 3D model, $\hat{v}_i$ is the corresponding vertex of v_i on the ground truth model, and N is the vertex number.

4.2. Comparison to state-of-the-art methods

We first compare our 3D body model estimation method with three kinds of model fitting methods from depth images. Pure model fitting method [25] deforms the SMPL template to the depths using the searched point correspondences between the template and input depths. Bogo et al. [6] first detects 2D body joints and then fits the SMPL template to the detected joints. Wei et al. [38] builds the point correspondences by matching the learned feature descriptors for depth images of human bodies. The 3D models are then generated by fitting the SMPL template to point correspondences found using [38]. We deform the estimated models of both [6] and [38] to the input depths further using searched point correspondences. The reconstruction errors with different methods are listed in Table 1. The comparison results on the DFAUST data using different methods are shown in Fig. 3. Please refer to the supplementary video for comparison results on depth sequences. The pure model fitting method has higher recovery error due to large discrepancy between the template and the input depths. The performance of [6] and [38] highly relies on the estimation of detected joints and learned point correspondences, respectively. Inaccurate joints and point correspondences might cause large reconstruction error using these methods. In contrast, our method predicts the 3D body models directly from point clouds without building point correspondences, thus leading to much higher recovery accuracy.

Since there is no deep learning method of 3D body model estimation from depth images, we extend RGB-based networks for the comparison by adding a 3D correspondence loss defined as Eq. 11. We train the regression network [18] on depth images and compare it with our method. We also compare our method with the recent method [21] by extracting features on depth images and regressing 3D meshes through Graph CNN. Kanazawa et al. [19] proposes a temporal encoder method for recovering the SMPL models from videos. We compare the temporal encoder of [19] with our method by employing their method on depth images. As shown in Table 1 and Fig. 3, the comparison results demonstrate that our method can outperform the state-of-the-art methods in recovering sequential 3D body models from a sequence of depth images. Rather than estimating the SMPL parameters as [18] and [19], our method adopts a spatial-temporal MAC network to predict 3D coordinates of mesh vertices in a coarse-to-fine manner, thus resulting in higher fitting accuracy to the point clouds. In Graph CNN of [21], the spatial neighbors of vertices are convoluted by a neighborhood averaging operation. Different from [21], our spatial MAC method learns the neighboring relationship for each vertex based on the differences of feature vectors and spatial positions, which can capture more discriminative features from neighboring vertices. The comparison results with temporal encoder of [19] demonstrate that our spatial-temporal MAC can successfully exploit both spatial and temporal features on the mesh vertices across sequential frames. Especially, by fitting the SMPL model [25] to our regressed mesh vertices, the estimated parametric models have the similar accuracy to that of our regressed 3D mesh-

Methods	SURREAL	Human3.6M	DFAUST
Pure model fitting [25]	140.6	148.3	110.1
Bogo et al. [6]	56.1	60.5	57.5
Wei et al. [38]	58.6	64.1	62.2
Kanazawa et al. [18]	54.3	59.8	58.1
Kanazawa et al. [19]	52.7	57.3	56.1
Kolotouros et al. [21]	49.5	54.3	52.2
Our method (Non parametric)	**18.2**	**21.4**	**19.7**
Our method (Parametric)	19.4	22.8	20.3

Table 1. Reconstruction errors (mm) with different methods tested on sequences from the three public datasets.

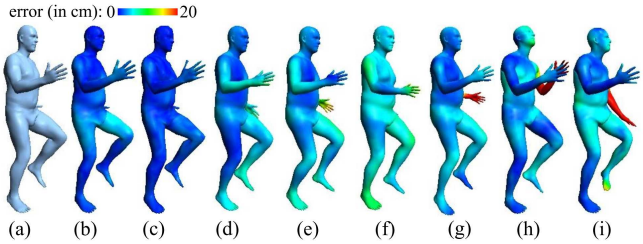

Figure 3. The visualization of reconstruction accuracies using different methods on the DFAUST data. (a) The input scan. (b) The fitted parametric result of our method. (c) The predicted non-parametric result of our method. (d) Kolotouros et al. [21]. (e) Kanazawa et al. [19]. (f) Kanazawa et al. [18]. (g) Wei et al. [38]. (h) Bogo et al. [6]. (i) The result of pure model fitting.

es, demonstrating it can accurately recover the parametric SMPL models from the non-parametric prediction.

4.3. Ablative analysis

Spatial mesh attention convolution. We first evaluate the effectiveness of our spatial mesh attention convolution (SMAC) by comparing our method with and without SMAC. For our method without SMAC, we do not apply the linear combination of neighboring vertex features with the learned attentional weights after the feature mapping. In addition, we test our method by replacing the attention model in the SMAC with a simply neighborhood averaging method [21]. The reconstruction errors using different methods are listed in Table 2. Our method with SMAC can achieve much lower errors than other methods, demonstrating the effectiveness of our SMAC. Compared to the simply averaging method, our SMAC can capture local geometry structure better in the coarse-to-fine regression framework by dynamically assigning the weights for the neighboring vertices.

Temporal mesh attention convolution. We also evaluate the effectiveness of the proposed temporal mesh attention convolution (TMAC) by comparing our method with and without TMAC. The comparison results shown in Table 2 demonstrate our method with TMAC can improve estimation accuracy of mesh vertices by encouraging the network to exploit discriminative temporal features. In addition, using the data of sequential frames can mitigate shape uncertainty and ambiguity. From the example shown in Fig. 4, the occluded hand fails to be recovered accurately without TMAC. In contrast, our method with TMAC faithfully estimates the 3D hand shape with a higher accuracy by utilizing the hand features observed from the sequential views.

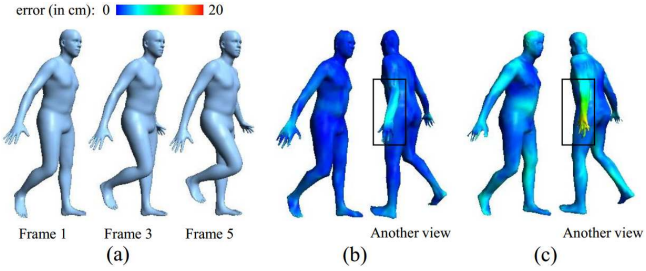

Figure 4. Reconstruction accuracies with and without temporal mesh attention convolution (TMAC). (a) The three input depths from consecutive five frames. (b, c) The results of frame 1 with and without TMAC shown from two views, respectively. By exploiting the hand features from the neighboring frames through TMAC, our method can recover 3D shape of the occluded hand more accurately (surrounded in rectangle).

Spatial-temporal mesh attention convolution. We further evaluate our spatial-temporal MAC by comparing it with a 3D regression method [18] of estimating SMPL models. In the 3D regression method, we extract features from point clouds using PointNet++ [31] and replace the 2D joint loss in [18] with the 3D correspondence loss. As shown in Table 2, the recovery accuracy using the 3D regression method is worse than our method, showing that it is difficult to accurately estimate SMPL model parameters from point clouds. In contrast, our method can accurately estimate 3D coordinates of mesh vertices from coarse to fine by leveraging the mesh topology through the spatial-temporal MAC.

Methods	SURREAL	Human3.6M	DFAUST
our method	**18.2**	**21.4**	**19.7**
Without SMAC	45.3	49.1	47.6
Simply averaging SMAC	25.7	26.5	25.9
Without TMAC	21.2	22.9	22.2
3D regression	65.8	68.4	65.3

Table 2. Reconstruction errors (mm) of our method, our method without spatial mesh attention convolution (SMAC), our method with simply averaging SMAC, our method without temporal mesh attention convolution (TMAC), and the 3D regression method.

Point Number	SURREAL	Human3.6M	DFAUST
2,500	18.2	21.4	19.7
5,000	18.0	20.8	19.3
7,500	17.9	20.7	19.1

Table 3. Reconstruction errors (mm) with different number of sampled points using our method.

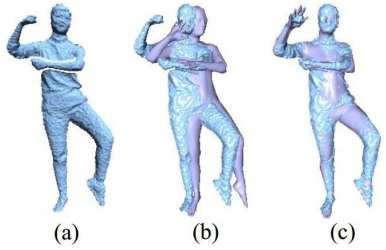

Figure 5. An example of weakly-supervised fine-tuning on "Girl 2" data. (a) The input depth. (b, c) The results before and after weakly-supervised fine-tuning, respectively. The overlay with alignment is shown between the 3D model and the raw depth.

Weakly-supervised fine-tuning on real data. To evaluate the effectiveness of our weakly-supervised fine-tuning on real detailed data, we compare the estimation results before and after the weakly-supervised fine-tuning. An example is shown in Fig. 5. Due to the lack of real detailed samples in training data, the predicted 3D models before the fine-tuning cannot fit to the input data well. Although there is relatively large recovery error for the predicted 3D models, they align with the point clouds roughly. With the supervision of the initially predicted 3D models, our fine-tuning network can generate more accurate 3D models that have consistent shapes and poses with the input real data.

The number of sampled points. We also investigate the influence of the number of sampled points L on the reconstruction accuracy using $2,500$, $5,000$, and $7,500$ sampled points. Table 3 shows the reconstruction errors using different number of sampled points. We observe that there is a slight improvement of the recovery accuracy with an in-

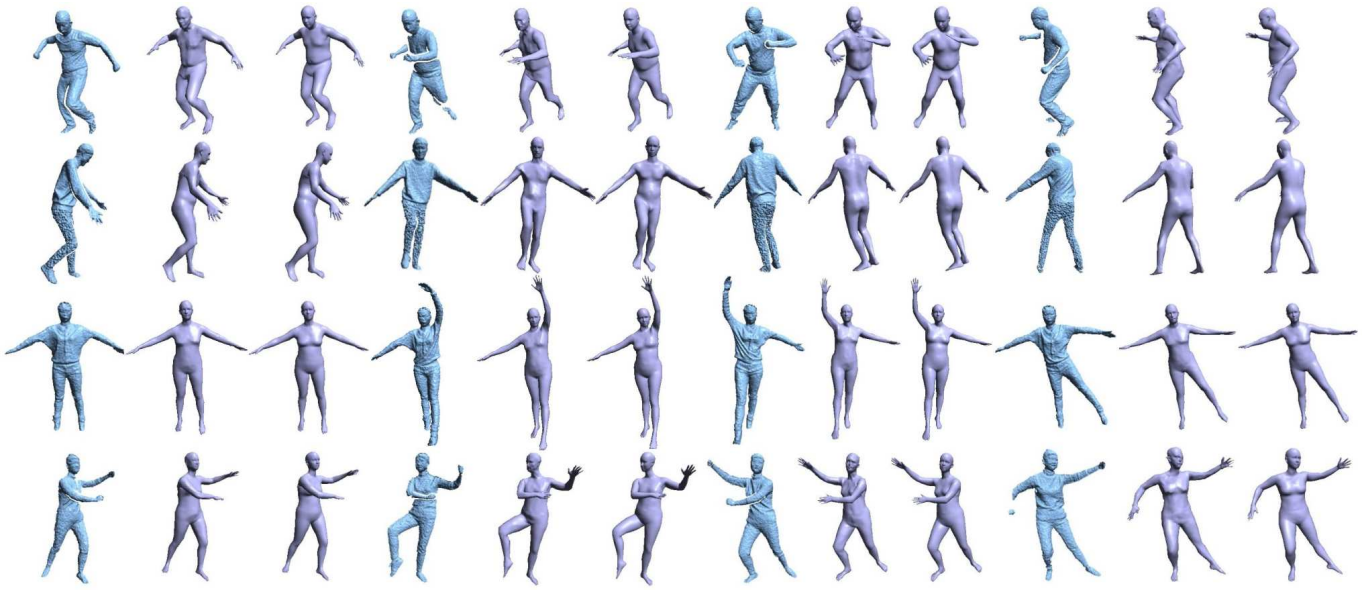

Figure 6. Some recovered 3D models using our method on real data. For each result, we show the extracted raw depth scan, the predicted non-parametric mesh and the fitted parametric model. From top to bottom: "Kungfu" data from [11], "Boy" data, "Girl" data from [11], and "Girl 2" data. Note that we show the raw depth scan instead of sampled points here for better visualization. Please see the supplementary video for reconstruction results of the entire sequences.

creasing number of points. However, the training process also takes much longer. To strike a balance between estimation accuracy and running efficiency, we choose $L = 2,500$ in our experiments. This experiment verifies that our approach is robust to a small number of sampled points since we can obtain a good accuracy with $L = 2,500$ points.

4.4. Test on real data

We test our method on real data of clothed human bodies with a variety of shapes and poses captured by a Kinect V2 sensor. "Kungfu" data and "Girl" data are from [11], and "Boy" data and "Girl 2" data are captured by ourselves. Fig. 6 shows some reconstruction results in the sequences. Please refer to the supplementary video for reconstruction results of the complete sequences. The input to our method is a sequence of 2500 uniformly sampled points from raw point clouds. We generate the 3D body models for five consecutive frames each time. Since our method predicts the 3D models from the point clouds directly without building point correspondences, there is no problem of error accumulation when handling the entire sequences. Although there are serious self-occlusions and arbitrary deformations on the real data, our method still can robustly and accurately estimate the 3D body shapes that fit to the input point clouds well. Through the proposed weakly-supervised fine-tuning, our method can generalize reliably to real point clouds of clothed bodies. Our method may fail in the cases of extremely large poses and loose clothes like long skirts. By applying our method on more real detailed data, we can generate a large dataset of 3D body models aligned with the real data for the needs of this kind of data in the community. The human motions are always tracked by searching point correspondences on successive depth frames in traditional approaches, while our method reconstructs the sequential 3D models directly from point clouds, which is a novel way to track human motions in a sequence.

5. Conclusion

In this paper, we addressed the problem of sequential 3D human pose and shape estimation from a sequence of point clouds. Instead of estimating the parametric models, we propose a spatial-temporal mesh attention convolution to accurately predict vertex coordinates of 3D meshes at different resolutions in a coarse-to-fine manner from latent features of point clouds. By dynamically assigning attentional weights to neighboring points in the spatial and temporal domains, the proposed spatial-temporal mesh attention convolution can exploit both local structured features of point clouds of a single frame and temporal structured features of point clouds of consecutive frames, which improves the recovery accuracy of sequential 3D body models. In addition, our method is successfully generalized to real detailed data captured by depth sensors through a weakly-supervised fine-tuning method. The experimental results on SURREAL, Human3.6M, DFAUST and real detailed data demonstrate the effectiveness of the proposed method.

Acknowledgments This work was partially supported by the Natural Science Foundation of China under Grant Nos.61602444, 61822310, U1713208, and Program for Changjiang Scholars. This work was also sponsored by SenseTime Research Fund.

References

[1] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019.

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics*, 24:408–416, July 2005.

[3] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3D human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019.

[4] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

[5] Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578, 2016.

[7] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[8] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4D: Real-time performance capture of challenging scenes. In *ACM SIGGRAPH*, 2016.

[9] Mingsong Dou, Jonathan Taylor, Henry Fuchs, Andrew Fitzgibbon, and Shahram Izadi. 3D scanning deformable objects with a single RGBD sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[10] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018.

[11] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using L0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[12] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo and motion reconstruction using a single rgbd camera. *ACM Transactions on Graphics*, 2017.

[13] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. LiveCap: Real-time human performance capture from monocular video. *ACM Transactions on Graphics*, 2019.

[14] Mir Rayat Imtiaz Hossain and James J. Little. Exploiting temporal information for 3D human pose estimation. In *European Conference on Computer Vision*, 2018.

[15] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V. Gehler, Javier Romero, Ijaz Akhter, and Michael J. Black. Towards accurate marker-less human shape and pose estimation over time. In *International Conference on 3D Vision (3DV)*, 2017.

[16] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. VolumeDeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*, pages 362–379, 2016.

[17] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), July 2014.

[18] 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*, pages 7122–7131, 2018.

[19] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015.

[21] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[22] 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*, pages 6050–6059, 2017.

[23] Chun-Liang Li, Tomas Simon, Jason Saragih, Barnabás Póczos, and Yaser Sheikh. LBS Autoencoder: Self-supervised fitting of articulated meshes to point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[24] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Chen. Recurrent 3D pose sequence machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248, 2015.

[26] Richard Newcombe, Dieter Fox, and Steve Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[27] 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*, pages 459–468, 2018.

[28] Veličković Petar, Guillem Cucurull, Casanova Arantxa, Romero Adriana, Liò Pietro, and Bengio Yoshua. Graph attention networks. *International Conference on Learning Representations*, 2018.

[29] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics*, 36(4), July 2017.

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

[31] 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*, pages 5099–5108, 2017.

[32] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision*, pages 725–741, 2018.

[33] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[34] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and Hans-Peter Seidel. Laplacian surface editing. In *Proceedings of the EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 179–188, 2004.

[35] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems*, pages 5236–5246, 2017.

[36] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *European Conference on Computer Vision*, 2018.

[37] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017.

[38] Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[39] Genzhi Ye, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt. Performance capture of interacting characters with handheld kinects. In *European Conference on Computer Vision*, pages 828–841, October 2012.

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

[41] 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 *Proceedings of the IEEE International Conference on Computer Vision*, October 2017.

[42] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[43] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3D pose and shape estimation of multiple people in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[44] Chao Zhang, Sergi Pujades, Michael Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[45] Jason Y. Zhang, Panna Felsen, Angjoo Kanazawa, Panna Felsen, and Jitendra Malik. Predicting 3D human dynamics from video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[46] Xiuming Zhang, Tali Dekel, Tianfan Xue, Andrew Owens, Qiurui He, Jiajun Wu, Stefanie Mueller, and William T. Freeman. MoSculp: Interactive visualization of shape and time. *arXiv:1809.05491*, 2018.

[47] 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 *European Conference on Computer Vision*, pages 384–400, 2018.

[48] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. DeepHuman: 3D human reconstruction from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[49] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.