Learning Dense Correspondences for Non-Rigid Point Clouds With Two-Stage Regression

Kangkan Wang, *Member, IEEE*, Guofeng Zhang, *Member, IEEE*, Huayu Zheng, and Jian Yang, *Member, IEEE*

***Abstract*— We propose a novel deep learning method to predict dense correspondences for partial point clouds of non-rigidly deformable targets. Dense correspondences are learned in the form of vertex displacements of a template mesh towards the point clouds. A two-stage regression framework is proposed to estimate accurate displacement vectors, including the global and local regression networks. Specifically, the global regression network estimates global displacements from the global features of the template mesh and point clouds through a graph CNN based hierarchical encoder-decoder network. Based on the initial displacements, a mesh can be generated that fits to the point clouds roughly. In the local regression network, a local feature embedding layer fuses local features of point clouds with graph features on the generated mesh through an attention mechanism. Consequently, the embedded local features are employed to refine the correspondences in local regions of the targets by predicting the increments of vertex displacements. Our method is further generalized to correspondence estimation on unseen real data with a robust fine-tuning method. The experimental results on diverse datasets of various deformable subjects (e.g., human bodies, animals, and hands) demonstrate that the proposed approach can accurately and robustly estimate dense correspondences from non-rigid point clouds.**

***Index Terms*— Dense correspondences, deep learning, non-rigid point clouds, weak supervision.**

Manuscript received September 14, 2020; revised May 28, 2021; accepted September 9, 2021. Date of publication October 6, 2021; date of current version October 12, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 61602444, Grant 61906089, and Grant U1713208; and in part by the Program for Changjiang Scholars; in part by the Fundamental Research Funds for the Central Universities under Grant NJ2020023; in part by the Open Project Program of the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, under Grant VRLAB2021C03, and in part by the Open Project Program of the State Key Laboratory of CAD&CG, Zhejiang University, under Grant A2106. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aswin Sankaranarayanan. *(Corresponding author: Kangkan Wang.)*

Kangkan Wang is with the Key Lab of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China, also with Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: wangkangkan@njust.edu.cn).

Guofeng Zhang is with the State Key Lab of CAD&CG, Zhejiang University, Zijingang Campus, Hangzhou 310058, China (e-mail: zhangguofeng@cad.zju.edu.cn).

Huayu Zheng and Jian Yang are with the Key Lab of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China, and also with Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: fox16789@163.com; csjyang@njust.edu.cn).

Digital Object Identifier 10.1109/TIP.2021.3116786

I. Introduction

FINDING dense correspondences [10], [28], [33], [35] for 3D shapes is one of the fundamental problems in computer vision and computer graphics, which is a significant building block in many promising applications such as games, robotics, and virtual reality. With the increasing availability of 3D point clouds, dense correspondence estimation from non-rigid point clouds has become paramount in many research topics such as multi-view stereo [53], object retrieval [19], 3D reconstruction [26], and motion tracking [5], [17], [52]. However, due to various factors such as variations of deformable subjects, incomplete 3D data and viewpoint changes of cameras, estimating dense correspondences from point clouds still remains challenging.

Most methods [5], [17], [54], [55] obtain dense point correspondences by registering a template model to input point clouds through non-rigid deformation techniques. These methods work on a sequence of frames rather than a single frame as the nearest neighboring searching method for point correspondences is only effective for small motions between neighboring frames. In the cases of large discrepancy between a single frame of point clouds and the template model, the found correspondences are prone to errors. The model fitting methods [6], [16], [30] first detect 2D joint locations and then fit a statistical model to the detected joints for the correspondences. Since the model fitting methods highly rely on joint estimation, it is difficult to handle the cases with strong occlusions or large poses, leading to the high correspondence errors. The deep learning method [52] directly predicts dense correspondences between 3D shapes of human bodies by converting the correspondence problem into a classification problem for multiple segmentations of the 3D human body. Nonetheless, due to the inconsistency between multiple segmentations, the predicted correspondences might be inaccurate from a single depth image.

Descriptor-based methods [2], [9], [28], [43] can establish correspondences between 3D shapes by matching local geometry descriptors. However, low resolution, random noise, and partial data pose significant challenge for computing local geometric features, which requires tedious manual efforts to remove outliers. By taking the descriptors [2], [43] as the inputs, non-rigid 3D shapes can be matched by estimating the functional maps [18], [27], [36], [41]. These methods generalize poorly across datasets due to reliance on the hand-crafted input descriptors. The recent method [14] learns geometry features from raw 3D shapes which improves the generalization

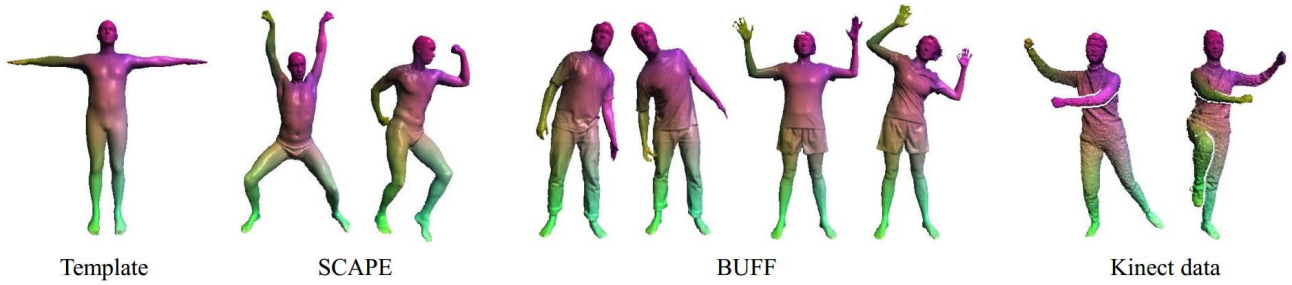


Fig. 1. Dense correspondences predicted using our method on SCAPE [1], BUFF [57], and Kinect data. Correspondence is visualized by the colors mapped from a template model. The input to our method is the sampled point clouds from a single depth image.

power, but it requires ground truth correspondences for real dataset. Affected by the input descriptors, the performance of these methods is sensitive to arbitrary deformations of modeling targets and data occlusion from a single view.

Recently, 3D-CODED [15] estimates the 3D correspondences by predicting the shape deformation of a template into the input shape. Since the network of 3D-CODED is trained on synthetic data and domain gap exists between real data and synthetic data, the accuracy and robustness are limited when handling real point clouds. Compared to 3D-CODED, our method can predict dense correspondences for deformable subjects more robustly and accurately especially on the unseen point clouds. Other methods [3], [29], [51] find the correspondences by estimating the 3D flow between two consecutive frames. However, these methods mainly focus on handling small or rigid motions.

In this paper, we propose a novel deep learning framework to predict dense correspondences for partial point clouds of non-rigidly deformable targets. The correspondences are estimated in the form of dense 3D displacement vectors of template vertices to the point clouds. The proposed framework consists of two stages, i.e. the global regression network and local regression network. The global regression network predicts the global displacements with global features of the template model and point clouds, while the local regression network estimates the displacement increments by exploiting local features of point clouds using an attention-based mechanism. To handle unseen real data, we propose a weakly-supervised model estimation method to create pseudo-ground truth correspondences which are provided to the two-stage network as full 3D supervision. With the initially predicted correspondences, the proposed model estimation method can robustly recover accurate 3D models that are consistent with the real data. The correspondences from the recovered 3D models are then used to fine-tune the networks, making our method generalize well to correspondence estimation on real data. Fig. 1 shows some correspondence results of our approach on various real dataset of human bodies. The experimental results on diverse datasets of different subjects (e.g., human bodies, animals, and hands) demonstrate the effectiveness of the proposed dense correspondence method for non-rigid point clouds. In summary, the main contributions of our method are as follows:

- We propose a novel deep learning method to predict dense correspondences of non-rigidly deformable targets by learning the vertex displacements of a template mesh to the partial point clouds. Our method can achieve accurate estimation of dense correspondences and outperform the state-of-the-arts. We will release our source code and trained models to benefit the community.
- We propose a two-stage regression framework to estimate dense displacements. The global regression network learns geometry features of the template mesh and point clouds to predict global displacements; the local regression network refines the displacements locally with the embedded local features between the point clouds and the mesh generated from the initial displacements.
- A local feature embedding technique is proposed with an attention mechanism to fuse local deep features of point clouds with geometry features of the graphs which effectively improves the correspondence accuracy in the local regions of point clouds.
- A robust fine-tuning method is proposed to generalize our method to handle real data. The pseudo-ground truth correspondences are generated for real data with a weakly-supervised model estimation method and are then supplied to the two-stage regression network as the full supervision.

II. Related Work

A. Dense Correspondences From 3D Model Estimation

3D model estimation aims to recover the 3D models from the input data. Dense correspondences can then be established from the registration results between the recovered 3D models and the input data. Most methods of 3D model estimation from a single color image match a parametric model [1], [6], [16], [30] to a set of observations from the input image, such as keypoints and silhouettes. For example, SMPLify [6] estimates a full 3D body mesh from a single color image by first predicting the 2D body joint locations and then fitting the statistical SMPL model [30] to the 2D joints. Deep learning based methods [21], [23], [37], [45], [56] regress the parametric model directly from a single color image. Kanazawa *et al.* [21] first extracts features on the color image and then infers the SMPL parameters of human bodies from the features through a 3D regression network. A mesh convolution method [23] first attaches the extracted features from the input color image to 3D vertex coordinates of a template mesh and then predicts coordinates of mesh vertices using a convolutional mesh regression.

Learning based methods are proposed to build dense correspondences for a single depth image. Metric regression

forest is adopted in [38] to directly predict correspondences for human bodies from depth images. Wei *et al.* [52] trains smoothly changed feature descriptors over depth images of human bodies and builds dense correspondences by matching the learned feature descriptors. LBS Autoencoder [25] matches articulated mesh models to point clouds by learning the linear-blend skinning deformation, which is mainly proposed for point clouds with complete 3D shape but not partial data. Optimization based models [54], [55] primarily handle a sequence of depth images since they rely on information of neighboring frames to build point correspondences for each frame. Our recent work [47] can predict the 3D body model sequence from an input sequence of partial point clouds of a human body with a spatial-temporal mesh attention convolution network.

B. Correspondence Estimation Based on Feature Descriptors

Discriminative feature descriptors play a fundamental role in building correspondences for deformable shapes. The spectral descriptors [2], [9], [28], [43] are derived from the eigenvalues and eigenvectors of the Laplace-Beltrami operator [4], thus allowing to deal with any shape representation, e.g. meshes, point clouds, or graphs. Geometric deep learning methods [9], [10], [33], [35] generalizes deep learning techniques to non-Euclidean structured data such as graphs and manifolds which can be applied for 3D shape correspondence estimation. For example, MoNet [35] learns task-specific features on non-Euclidean domains with mixture model networks which formulates previously proposed Geodesic CNN [33] and Anisotropic CNN [10] as particular instances of [35]. These methods can obtain robust descriptors for isometric and non-isometric deformations, but the performance degrades significantly when handling partial point clouds due to irregular structure and data missing.

Based on the pre-computed features [2], [43], pointwise correspondences can be found among non-rigid shapes by computing functional maps [36]. The functional map estimation methods are fundamentally error prone due to the initial choice of descriptor functions. To alleviate this dependence, FMNet [27] learns a transformation of initial descriptors to compute more optimal functional map, but requiring ground truth point-wise maps. Following the similar functional map framework, the unsupervised versions are proposed by enforcing structural properties on the computed map (SURFMNet [41]) or replacing supervision of pointwise correspondences with standard geometric quantities (Unsup FMNet [18]). The recent method [14] learns the features directly from raw shape geometry and has better generalization power than current descriptor-based learning methods. Since these methods need compute the feature descriptors on the partial point clouds, the performance of these methods is very sensitive to arbitrarily non-rigid deformations and partial data under view occlusion.

C. 3D Correspondence Estimation for Point Clouds

Recent works estimate the correspondences by learning 3D flow from point clouds using deep neural networks. Scene flow can be predicted in [3], [29] from unstructured point clouds by learning geometric similarities and spatial relations of points, but these methods are limited to handle small motions between neighboring frames. The other kinds of methods [31], [50], [51] build 3D correspondences by finding the transformation of registering one point cloud into another. These registration estimation methods are mainly proposed for dealing with rigid point clouds. To the best of our knowledge, there are few methods that directly learn dense 3D flow for non-rigidly deformable point clouds.

3D-CODED [15] builds dense 3D correspondences for non-rigidly deformable point clouds with an encoder-decoder architecture. From the extracted global features, this method estimates the deformations of a template model into the point clouds by regressing coordinates of model vertices. When testing on real partial point clouds, we find in the experiments that the predicted models of their network are prone to be unreasonably distorted and significantly inconsistent with the input shapes due to the lack of real training data. In 3D-CODED, the initially predicted models are refined through an additional optimization step by minimizing the Chamfer distance between the input and transformed template. However, as stated in 3D-CODED, the quality of the initialization (the prediction of their network) is crucial for the deformation optimization. That is the reason why the deformation optimization is very likely to fail when Chamfer distances are computed incorrectly from the unreliable initial models. In this paper, we sufficiently exploit local geometry features on the point clouds with an attention strategy to improve the correspondence accuracy and successfully handle unseen real point clouds with a robust fine-tuning method.

III. Proposed Approach

Given a template model and the partial point clouds of deformable targets, our method accurately predicts the corresponding point for each template vertex in the point clouds by estimating 3D displacement vectors of each vertex. We choose the human body as the deformable target to present our approach which is a popular modeling subject in the areas of computer vision such as 3D human pose and shape estimation [21], [47], [55]. It is noted that our approach can be applied to handle general non-rigidly deformable targets. In the experiments, we demonstrate the effectiveness of our method on other targets such as animals and hands. We use the Skinned Multi-Person Linear model (SMPL) [30] to represent the template model of human bodies. The input point clouds can be human bodies with arbitrary non-rigid deformations. Fig. 2 illustrates the framework of the proposed method. Our framework consists of two stages, i.e., the global regression network and local regression network. We adopt a graph CNN based hierarchical encoder-decoder structure for both two regression networks. The global regression network estimates the global displacements based on the concatenated global features extracted on the template model and point clouds. Based on the initial displacements, the template is registered to the point clouds globally. Consequently, with the registered mesh, the local regression network further refines the displacements

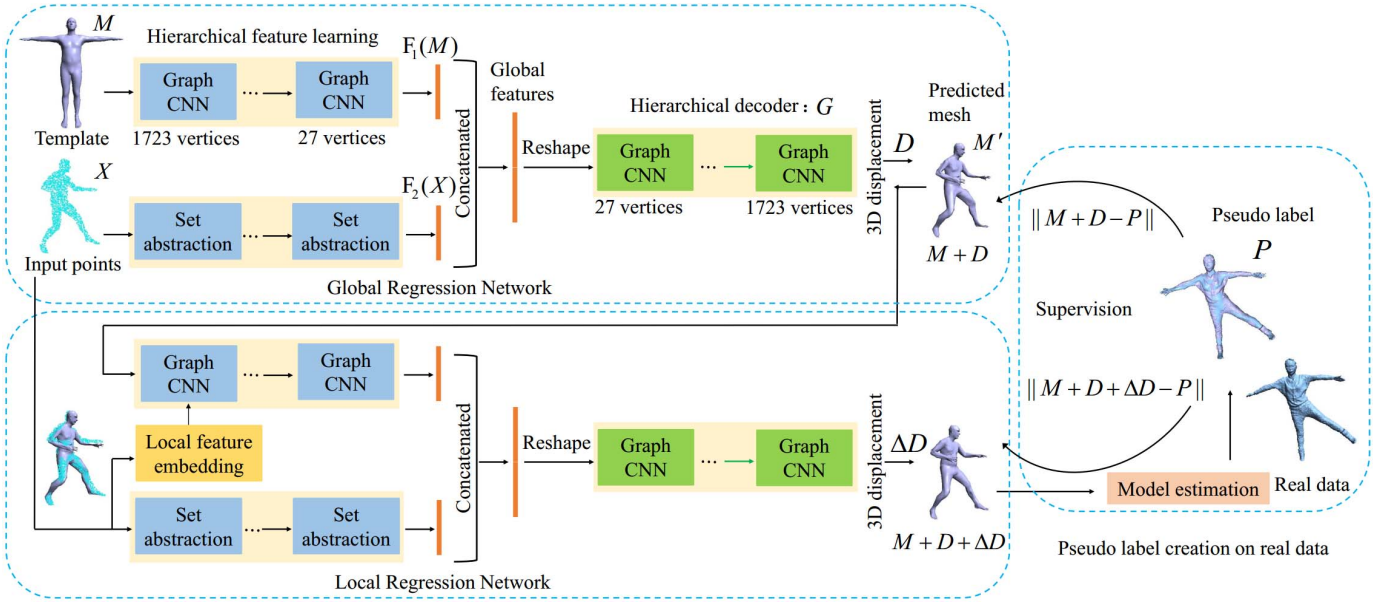


Fig. 2. Overview of the proposed framework. Our two-stage regression method can predict dense correspondences in the form of 3D displacements between vertices of a template model and the input points. Please refer to Sect. III for detailed description. The global and local regression networks are firstly trained on synthetic data; and then fine-tuned on real dataset without 3D ground truth correspondences by introducing pseudo-ground truth correspondences through model estimation. Initialized from the predicted mesh on real data, the model estimation recovers the parametric model that fits to the input points. The recovered parametric model is then used as the pseudo-ground truth correspondences to fine-tune both global and local regression networks in a fully-supervised manner.

in the local regions of point clouds. In the local regression network, the local feature embedding layer fuses the local features of point clouds with that of the registered mesh in an attention strategy to estimate incremental displacements for template vertices. To further effectively handle real data, pseudo-ground truth correspondences are first generated using a robust model estimation method and then utilized as the supervision to fine-tune the two-stage network.

A. Geometric Feature Learning

1) Spectral Graph CNN: We adopt the Chebyshev spectral Graph CNN [13] to capture geometry features on the mesh and estimate 3D displacements from the extracted features. Compared to the multi-layer perceptrons of estimating coordinates of model vertices [15], graph CNN can regress more smooth 3D shapes by implicitly utilizing mesh topology, as demonstrated in the recent works [23], [47]. The 3D template mesh is defined as $\mathcal{M} = (\mathcal{V}, A)$, where $\mathcal{V} = \{\boldsymbol{v}_i\}_{i=1}^{n}$ is a set of n vertices, $A \in \mathbb{R}^{n\times n}$ is the adjacency matrix ($A_{ij} = 1$, an edge connecting vertices i and j; $A_{ij} = 0$, otherwise). The normalized graph Laplacian is computed as $L = I_n - D^{-1/2}AD^{-1/2}$, where $D = diag(\Sigma_j A_{ij})$ is the diagonal degree matrix and I_n is the identity matrix. Given $\boldsymbol{f} = (\boldsymbol{f}_1, \cdots, \boldsymbol{f}_n)^T \in \mathbb{R}^{n\times p}$ represent the input features of n vertices with p-dimension of feature. The mesh convolution is formulated as a recursive Chebyshev polynomial of order K which captures geometry features of the K-ring neighbors for each vertex. The learned features $\boldsymbol{f}'$ are defined as:

$$\boldsymbol{f}' = \sum_{k=0}^{K-1} T_k(\tilde{\boldsymbol{L}}) \cdot \boldsymbol{f} \cdot \boldsymbol{\theta}_k, \tag{1}$$

where $T_k \in \mathbb{R}^{n\times n}$ is the Chebyshev polynomial of order k that is computed recursively as $T_k(\boldsymbol{x}) = 2\boldsymbol{x}T_{k-1}(\boldsymbol{x}) - T_{k-2}(\boldsymbol{x})$ with $T_0 = \mathbf{1}$ and $T_1 = \boldsymbol{x}$, $\tilde{\boldsymbol{L}} = 2\boldsymbol{L}/\lambda_{max} - I_n$ is the rescaled Laplacian, λ_{max} is the maximum eigenvalue of $\boldsymbol{L}$, and $\boldsymbol{f}' \in \mathbb{R}^{n\times q}$ is the output q-dimensional feature. Chebyshev coefficients $\boldsymbol{\theta}_k \in \mathbb{R}^{p\times q}$ are trainable parameters in the graph convolutional layer. We test the performance of our method with different values of the order K. The average correspondence errors (in mm) are 19.4, 18.1, 18.3, 29.7 for $K = \{1, 2, 4, 6\}$ on SURREAL dataset [46], respectively. We can see that the best accuracy is obtained when $K = 2$. High value of K involves much redundant feature mapping at a coarse graph (e.g. 27 vertices), which might lead to larger errors. Thus, we set $K = 2$ in our experiments.

2) Feature Extraction on Point Clouds: We use a similar framework of PointNet++ [39] to extract geometry features of point clouds which is composed by several set abstraction levels. The inputs are a set of points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ and the corresponding point features $\boldsymbol{f} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_n\}$. In each set abstraction level, farthest point sampling is firstly performed to sample m centroid points $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m\}$ which cover the entire point set. For each centroid point $\boldsymbol{y}_i$, we then find k neighboring points among all points within a radius r. The k neighboring points of each centroid point form point sets in a local region. Both k and r are constant values for all the centroid points in the same set abstraction level. In the higher set abstraction level, we enlarge r and increase k accordingly. From m local regions of all centroid points, we use a multi-layer perceptron network to extract features for each point. Finally, a max pooling operator is applied in each local region to aggregate information from all the neighboring points to extract local features for point $\boldsymbol{y}_i$.

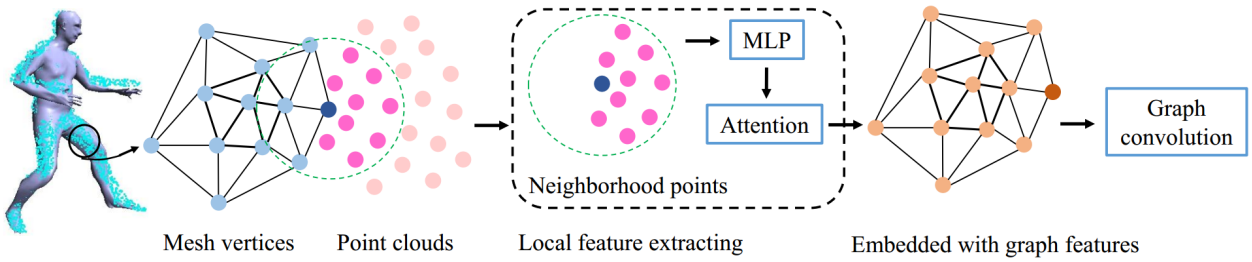


Fig. 3. Illustration of local feature embedding. We extract local features in the neighborhood points of each vertex on the registered mesh through an attention mechanism. The local features are then embedded with graph features which are taken as the input to feature learning of graph convolution.

After several successive set abstraction levels, a global feature vector can be extracted over the whole point clouds.

B. Global Regression Network

The global regression predicts global 3D displacement vectors for template vertices with a graph CNN based hierarchical encoder-decoder network. The encoder network extracts the global features on the template model $\boldsymbol{M}$ and the point clouds $\boldsymbol{X}$. In order to capture both global and local geometry context, we apply graph CNN on multi-resolutions of the template mesh hierarchically. We adopt the mesh sampling method of [40] to generate meshes of different resolutions. The mesh downsampling and upsampling operations are performed by left-multiplying the downsampling and upsampling matrices with the mesh, respectively. The mesh at each resolution has a fixed topology, and downsampling and upsampling matrices are precomputed at different resolutions using the sampling method proposed in [40]. By performing the graph convolution and downsampling iteratively, the graph features are finally mapped to a global feature vector $\boldsymbol{F}_1(\boldsymbol{M})$, where $\boldsymbol{F}_1$ is a feature mapping function. We apply successive set abstraction levels in PointNet++ to iteratively downsample the point clouds and capture local features in local regions of points. Through the max pooling operator over the features of all points in the final level, a global feature vector $\boldsymbol{F}_2(\boldsymbol{X})$ is obtained from the point clouds $\boldsymbol{X}$, where $\boldsymbol{F}_2$ is a feature mapping function.

From the concatenated global features $[\boldsymbol{F}_1(\boldsymbol{M}); \boldsymbol{F}_2(\boldsymbol{X})]$, the decoder network estimates the 3D displacement vectors between the template vertices and corresponding points:

$$\boldsymbol{G} : [\boldsymbol{F}_1(\boldsymbol{M}); \boldsymbol{F}_2(\boldsymbol{X})] \rightarrow \boldsymbol{D}, \tag{2}$$

where $\boldsymbol{G}$ is the mapping function of the decoder network, $\boldsymbol{D} \in \mathbb{R}^{n\times 3}$ denotes the 3D displacement vectors of all n vertices. The decoder network estimates the 3D displacements with a hierarchical graph CNN in a coarse-to-fine manner. The concatenated features are firstly mapped to graph features of the coarsest mesh using a fully connected layer. Graph CNN and mesh upsampling are then applied iteratively to regress the displacement vectors using the following loss:

$$L_{global} = \sum_{i=1}^{n} \parallel \hat{\boldsymbol{v}}_i + \boldsymbol{d}_i - \boldsymbol{p}_i \parallel_2^2, \tag{3}$$

where $\hat{\boldsymbol{v}}_i$ is the 3D coordinates of the i-th vertex on the template, $\boldsymbol{d}_i$ is the regressed 3D displacement vector to the corresponding point of the i-th vertex, and $\boldsymbol{p}_i$ is the ground truth 3D coordinates of the corresponding point.

C. Local Regression Network

After the global regression, the initial corresponding points are predicted for all template vertices. A new mesh $\boldsymbol{M}'$ can be generated with the same topology as the template $\boldsymbol{M}$ which is registered to the point clouds $\boldsymbol{X}$ roughly. In the global regression, we only utilize the global geometric features extracted on the entire point clouds, but ignore many local geometric features in local parts of point clouds. Thus, correspondences are likely to be inaccurate in the local regions. Based on the predicted mesh $\boldsymbol{M}'$, the local regression network aims to estimate more accurate vertex displacements.

We propose a local feature embedding layer to fuse local features of point clouds $\boldsymbol{X}$ with graph features of mesh $\boldsymbol{M}'$. Fig. 3 illustrates the process of our local feature embedding. With the registered mesh $\boldsymbol{M}'$, we can exploit local features of the input point clouds around the vertices of mesh $\boldsymbol{M}'$. For each vertex $\boldsymbol{v}_i$ on mesh $\boldsymbol{M}'$, we search the neighboring point set $\boldsymbol{X}_i$ in the point clouds within a radius r around $\boldsymbol{v}_i$. We then apply a MLP to encode deep features for each point in the neighborhood $\boldsymbol{X}_i$ as:

$$\boldsymbol{f}'_j = MLP(\boldsymbol{f}_j, \boldsymbol{x}_j - \boldsymbol{v}_i), \quad \{\boldsymbol{x}_j \mid \boldsymbol{x}_j \in \boldsymbol{X}_i\}, \tag{4}$$

where $\boldsymbol{f}_j$ is the input feature of point $\boldsymbol{x}_j$ in $\boldsymbol{X}_i$, and $\boldsymbol{f}'_j$ is the output feature of $\boldsymbol{x}_j$.

To capture distinctive features in the neighborhoods, we adopt an attention mechanism to learn the relationship of different points in $\boldsymbol{X}_i$ with $\boldsymbol{v}_i$. The attentional weights of $\boldsymbol{v}_i$ with its neighboring points $\boldsymbol{x}_j$ are computed as follows:

$$\boldsymbol{\alpha}_{ij} = \boldsymbol{\alpha}([\boldsymbol{f}'_j - \boldsymbol{g}_i, \boldsymbol{x}_j - \boldsymbol{v}_i, \boldsymbol{r}_j \boldsymbol{n}_i]), \quad j \in \mathcal{N}_i, \tag{5}$$

where $\boldsymbol{g}_i$ is the local geometry feature of the vertex $\boldsymbol{v}_i$, $\mathcal{N}_i$ is the neighbor set of vertex $\boldsymbol{v}_i$, and $\boldsymbol{r}_j$ and $\boldsymbol{n}_i$ are the normals of $\boldsymbol{x}_j$ and $\boldsymbol{v}_i$, respectively. Besides the feature vector differences, we also consider the position distances and normal angles between two points to estimate the point relationship, which is similar to the idea of point correspondence finding. Then, feature vector differences, position differences, and dot products of normals are concatenated as the input features, and the attentional weights of vertex $\boldsymbol{v}_i$ are regressed by the attention mechanism $\boldsymbol{\alpha}$, which is a MLP in our experiments.

To handle the neighbors across different vertices and spatial scales, the attentional weights are normalized across all the

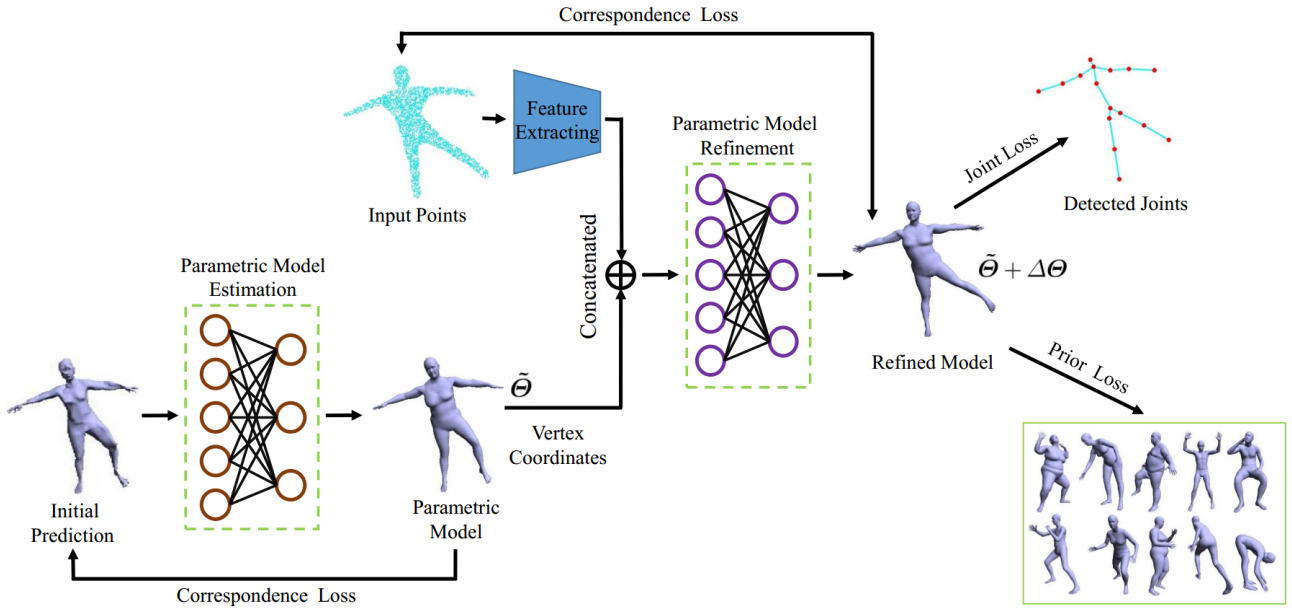


Fig. 4. An illustration of model estimation that creates pseudo labels for real data. Based on the predictions of the two-stage regression, we estimate the 3D models that fit to the input points through initial estimation and refinement of parametric models. The vertex coordinates of 3D models can then be used as pseudo-ground truth correspondences to fine-tune the two-stage regression network.

neighbors of vertex $\boldsymbol{v}_i$ as follows:

$$\tilde{\boldsymbol{\alpha}}_{ij} = softmax_j(\boldsymbol{\alpha}_{ij}) = \frac{exp(\boldsymbol{\alpha}_{ij})}{\sum_{s\in\mathcal{N}_i} exp(\boldsymbol{\alpha}_{is})}, \tag{6}$$

where $\boldsymbol{\alpha}_{ij}$ is the attentional weight vector of point $\boldsymbol{x}_j$ in neighborhood $\boldsymbol{X}_i$ relative to vertex $\boldsymbol{v}_i$. The local features of $\boldsymbol{X}_i$ are computed by a linear combination of the neighboring point features with the normalized attentional weights:

$$\tilde{\boldsymbol{h}}_i = \sum_{j\in\mathcal{N}_i} \tilde{\boldsymbol{\alpha}}_{ij} \boldsymbol{f}'_j + \boldsymbol{b}_i, \tag{7}$$

where $\boldsymbol{b}_i \in \mathbb{R}^F$ is a learnable bias. The extracted local features $\tilde{\boldsymbol{h}}_i$ in neighborhood $\boldsymbol{X}_i$ are then concatenated with the geometry feature $\boldsymbol{g}_i$ of vertex $\boldsymbol{v}_i$. The fused local features are taken as the input graph features to learn deep features on the mesh $\boldsymbol{M}'$ using hierarchical graph CNN. In ablation studies, we perform the local feature embedding layer at different mesh resolutions and find that it is more effective to apply it after the graph convolution layer on the full mesh. In this way, local features of the point clouds can be exploited more sufficiently.

In the local regression network, we employ a graph CNN based hierarchical encoder-decoder structure similar to the global regression. The global features of point clouds are extracted and concatenated with the learned deep features of $\boldsymbol{M}'$. A hierarchical graph CNN is then applied to estimate the increments of 3D displacement vectors from the newly concatenated features with the following loss:

$$L_{local} = \sum_{i=1}^{n} \parallel \boldsymbol{v}_i + \Delta\boldsymbol{d}_i - \boldsymbol{p}_i \parallel_2^2, \tag{8}$$

where $\boldsymbol{v}_i$ is the 3D coordinates of the i-th vertex on the initially predicted mesh $\boldsymbol{M}'$, $\Delta\boldsymbol{d}_i$ is the regressed increment of 3D displacement vector to the corresponding point of $\boldsymbol{v}_i$, and $\boldsymbol{p}_i$ denotes the ground truth correspondences as Eq. (3).

D. Fine-Tuning on Real Data

Due to the domain gap between the real data and the synthetic training data, the trained two-stage network may fail to work well on the real data (e.g. depths captured by Kinect). In this section, we generalize our method to real data with a robust fine-tuning method. Based on the initially predicted correspondences $\tilde{\boldsymbol{P}}$ for the input point clouds, we propose a model estimation method to recover the parametric models that fit to the point clouds robustly and accurately. Then, by supplying the 3D models of these real data to the network as full 3D supervision, we fine-tune the two-stage regression to improve its performance. An illustration of model estimation is shown in Fig. 4 which creates the pseudo-ground truth correspondences for the real data. The parametric SMPL model [30] of human bodies is represented using a 85 dimensional vector $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{R}, \boldsymbol{t})$, where $\boldsymbol{\beta} \in \mathbb{R}^{10}$, $\boldsymbol{\theta} \in \mathbb{R}^{69}$ and $(\boldsymbol{R}, \boldsymbol{t})$ denote the shape parameters, pose parameters and the global rigid motion of the human body, respectively. For other deformable targets, we can adopt the corresponding parametric models like SMAL [59] of animals and MANO [24] of hands.

From the initial correspondences $\tilde{\boldsymbol{P}}$, we employ fully connected networks to learn the initial parametric model $\boldsymbol{V}(\tilde{\boldsymbol{\Theta}})$ using the following loss:

$$E_{init}(\tilde{\boldsymbol{\Theta}}) = \frac{1}{n}\sum_{i}^{n} \|\boldsymbol{V}(\tilde{\boldsymbol{\Theta}})_i - \tilde{\boldsymbol{p}}_i\|_2^2 + \delta L_{prior}, \tag{9}$$

where $\boldsymbol{V}(\tilde{\boldsymbol{\Theta}})_i$ is the i-th vertex of the initial parametric model $\boldsymbol{V}(\tilde{\boldsymbol{\Theta}})$, $\tilde{\boldsymbol{p}}_i$ is the predicted corresponding point of the i-th vertex, and δ is a regularization parameter. L_{prior} is an adversarial prior [21] to prevent producing implausible 3D models. For animals and hands, we utilize the prior loss defined in [59] and [24], respectively. Due to the domain gap, unnatural meshes are easily predicted on real data which

have an abnormal shape of human bodies. This problem is not solved in the related works like [15]. In 3D-CODED [15], the optimization step requires a reasonable initialization and may fail to correct the abnormal shape. In our model estimation, the normal body shape can be recovered effectively that fit to the unreasonable predictions. As shown in Fig. 9 (b), the models generated from the predicted correspondences are distorted seriously. The parametric models can successfully rectify the unnatural deformations, as shown in Fig. 9 (c). It should be noted that we only use the predicted correspondences as the supervised signals during the initial model estimation.

By aligning the initial model $\boldsymbol{V}(\tilde{\boldsymbol{\Theta}})$ with the correspondences in the input point clouds, we can estimate the parametric model accurately that is consistent with the point clouds. Since the input point clouds are partial, the nearest neighboring point found by the Chamfer distance [15] may be incorrect. To solve this problem, we define a distance measure F between the input point p and the model vertex v as [49]:

$$F(p, v) = \max(1 - \frac{\| \boldsymbol{x}_p - \boldsymbol{x}_v \|}{\theta_x}, 0) \cdot \max(\boldsymbol{n}_p \boldsymbol{n}_v, 0), \quad (10)$$

where $\boldsymbol{x}_p$ and $\boldsymbol{n}_p$ denote the position and normal of point p respectively, and $\boldsymbol{x}_v$ and $\boldsymbol{n}_v$ denote the position and normal of model vertex v, respectively. The angles of point normals can effectively prune erroneous correspondences under partial point clouds. For each point p in the input points, we can find the vertex $\tilde{v}$ from the estimated model to maximize F. If $F(p, \tilde{v}) > 0$, the correspondence $(p, \tilde{v})$ is correct, otherwise it is removed. For each vertex on the models, we also find the corresponding point in the input point clouds in the same way.

Since the initial parametric model has a close shape or pose to the input points, the model estimation can become easier by taking the initial parametric model as the input. Thus, we concatenate vertex coordinates of the initial model with the features extracted from the point clouds to estimate the refined parametric model. We also employ fully connected networks to learn the updates of model parameters by minimizing the following function:

$$\begin{aligned} E(\Delta\boldsymbol{\Theta}) = & \frac{1}{\sum_k \sigma_k} \sum_i \|\sigma_i (\boldsymbol{V}(\boldsymbol{\Theta})_i - \boldsymbol{x}_{\boldsymbol{\varepsilon}_i})\|_2^2 \\ & + \frac{1}{\sum_k \tau_k} \sum_j \|\tau_j (\boldsymbol{x}_j - \boldsymbol{V}(\boldsymbol{\Theta})_{\boldsymbol{\epsilon}_j})\|_2^2 \\ & + \gamma \sum_m \|\boldsymbol{\pi}_c (\boldsymbol{J}_{3D}(\boldsymbol{V}(\boldsymbol{\Theta}))_m) - \hat{\boldsymbol{J}}_m\|_1 + \delta L_{prior}, \end{aligned} \quad (11)$$

where $\boldsymbol{\Theta} = \tilde{\boldsymbol{\Theta}} + \Delta\boldsymbol{\Theta}$ denotes the refined parameters by adding the updates $\Delta\boldsymbol{\Theta}$, $\boldsymbol{V}(\boldsymbol{\Theta})_i$ is the i-th vertex of the refined parametric model $\boldsymbol{V}(\boldsymbol{\Theta})$, $\boldsymbol{\varepsilon}_i$ is the correspondence index in the point clouds of the i-th vertex, $\boldsymbol{x}_j$ is the j-th point of the point clouds, $\boldsymbol{\epsilon}_j$ is the correspondence index of $\boldsymbol{x}_j$ in the model vertices, $\sigma_i(\tau_j) \in \{0, 1\}$ is a mask (if the correspondence is correct, $\sigma_i = 1$; otherwise, $\sigma_i = 0$), $\sum_k \sigma_k$ ($\sum_k \tau_k$) is the number of correct correspondences, γ and δ are regularization parameters, and $\hat{\boldsymbol{J}}_m$ are the ground truth 2D positions of the m-th joint. The first two terms of the loss function measure the correspondence distances between the estimated model and the input points. The 3D joint locations $\boldsymbol{J}_{3D}(\boldsymbol{V}(\boldsymbol{\Theta}))_m$ are obtained with linear regression [30] from the mesh vertices. The 3D joints are then projected into 2D image space with the camera parameters $\boldsymbol{\pi}_c$ to form estimated 2D joints. We adopt the joint loss for human bodies and detect the ground truth 2D joints using OpenPose [12]. By incorporating the joints, our method can handle large deformations that fail to be recovered from the initial predictions. Some results of the refined parametric models are shown in Fig. 9 (d). In our experiments, we set $\gamma = 50$ and $\delta = 10$ empirically.

The vertex coordinates of the refined 3D models are leveraged as the pseudo-ground truth correspondences to supervise the fine-tuning of the two-stage regression network. Trained on synthetic data, our two-stage regressor cannot work well on real data due to the domain gap. By providing pseudo-ground truth correspondences as full supervision to the two-stage regressor, the performance of our method can be improved remarkably on the real data. In this way, the two-stage regressor can produce more accurate results on real data as more examples are supplied to the network as supervision. The fine-tuning on real data has been done in related works like 3D-CODED [15], but the problem and goal of our method are different from the related works. The goal of our method is to refine the unreasonable predictions of the networks and utilize the refined results in return to boost the performance of the two-stage regression. In addition, the idea of our model estimation can be applied in the optimization-based model fitting methods to obtain the parametric models for real data. However, the direct optimization is rather sensitive to the initialization of parameters and highly relies on joint detection. When tested on samples with strong occlusions or large poses, the optimization easily gets trapped in local minima and the model fitting will fail. Benefitting from solving the fitting problem in a learning framework, our model estimation can perform more robustly and accurately under large poses or strong occlusions and also achieve faster running efficiency. In this work, a complete deep learning framework is developed which integrates the supervised learning and the pseudo label creation so that they can collaborate with each other efficiently.

IV. Experiments

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

A. Implementation Details

1) Datasets: We conduct experiments on the SURREAL [46], Human3.6M [20], DFAUST [8], and various real data. Each training model in the SURREAL dataset is rendered from 15 different viewpoints arranged evenly in a cycle around the 3D model to create depth images, which simulates real depths captured by a depth sensor. From the rendered depths, we uniformly sample $100,000$ depth images for both men and women as the training data. DFAUST dataset [8] contains real registered scans of human bodies, which have shapes and

motions beyond the SMPL model. We generate 80, 000 depth images for both male and female as the training data from DFAUST dataset. The depths are converted to point sets for the training. Each test model is also rendered from 15 different viewpoints to generate depth images. We uniformly sample 10, 000 frames with different poses and shapes as the test data for the three public datasets, respectively. The test data do not have the same human bodies as the training data. For the real data, we use point clouds captured by Kinect and publicly available datasets such as SCAPE [1], BUFF [57], FAUST [7], and CAPE dataset [32]. We also test our method on the data of deformable subjects like animals and hands.

2) Architecture and Experimental Settings: The raw point clouds of depth images are uniformly downsampled to $n = 3000$ in our experiments. We use a template mesh with 1723 vertices downsampled on the original SMPL mesh [30] to avoid spatial redundancy of the vertices. The template meshes of other subjects are handled in the same way and the downsampled vertex number can be changed according to the topology of the template. The feature extraction on the meshes applies four successive graph CNN [13] on different resolutions of graphs with downsampling factors of $\{2, 2, 4, 4\}$ and feature channels of $\{32, 64, 128, 256\}$, and finally outputs a 1024-dim feature vector with a full-connected layer. The global features extracted on the point clouds compose of four successive set abstraction levels of PointNet++ [39] with downsampled point numbers of $\{1500, 500, 150, 50\}$ and finally generates a 1024-dim feature vector. In the four successive set abstraction levels, the neighbor point number and the searching radius are set to $\{16, 32, 64, 64\}$ and $\{0.05, 0.1, 0.2, 0.4\}$, respectively. For the local feature embedding layer, we search 256 neighboring points within a radius $r = 0.2m$ around the vertices. We use $mlp\{32, 32, 64\}$ to extract local features of point clouds and $mlp\{32, 64\}$ for the attention mechanism. The concatenated 2048-dim feature vector is transformed to a (27×256)-dim vector with a full-connected layer and reshaped to 27 vertices with 256-dim feature vector. Four successive graph CNN are then applied with upsampling factors of $\{4, 4, 2, 2\}$ and feature channels of $\{256, 128, 64, 32\}$, finally outputting the 3D displacement vectors of 1723 vertices. Each graph convolution is followed by a ReLU layer except the last one of regressing the 3D displacements. Both two regression networks in the weakly-supervised model estimation consist of two fully-connected layers with 1024 neurons each with a dropout layer, followed by a final layer of 85-dim neurons. There are 3 iterations in the regressor and the network of the adversarial prior is the same as that in [21]. The learning rate is set to 1×10^{-3}. We use Adam optimizer [22] with the batch size of 8. The average running time for a test sample is about 16.1ms for the global regression, 29.3ms for the local regression, and 14.5ms for the model estimation with a NVIDIA 2080 Ti GPU.

B. Comparison to State-of-the-Art Methods of 3D Model Estimation

Dense correspondences can be established by fitting 3D template models to depth images. We first compare our correspondence method with three kinds of model fitting methods for a single depth image. Pure model fitting method [30] deforms the SMPL template to the input depths directly. Bogo *et al.* [6] first detects 2D body joints and then fits the SMPL template to the point clouds based on detected joints. Wei *et al.* [52] builds the point correspondences by matching the learned feature descriptors for depth images of human bodies. We register the template model to the point clouds based on the predicted correspondences [52] using their released codes. The correspondence errors are computed between vertices of the fitted 3D models and the ground truth corresponding points as in [47]. The correspondence errors with different methods are listed in Table I. The comparison results on the SURREAL data using different methods are shown in Fig. 5. The pure model fitting method has much higher error because there is large discrepancy between the template model and the input depth and correspondences fail to be searched correctly. The performance of Bogo *et al.* [6] highly relies on the joint estimation which has difficulties in handling the cases with strong occlusions or large poses. With inaccurately detected joints, searched correspondences are prone to be unreliable. Also, erroneous deformations may be generated during the fitting by merely using joint information. These reasons make the optimization get trapped in local

TABLE I

CORRESPONDENCES ERRORS (mm) WITH DIFFERENT METHODS TESTED ON SURREAL, HUMAN3.6M, AND DFAUST DATASETS

Methods	SURREAL	Human3.6M	DFAUST
Pure model fitting [30]	208.7	212.5	210.1
Bogo et al. [6]	65.5	72.3	65.1
Wei et al. [52]	63.3	68.2	63.1
Kanazawa et al. [21]	50.1	57.6	56.3
Kolotouros et al. [23]	47.1	52.4	51.8
3D-CODED [15]	31.9	32.6	34.3
Wang et al. [47]	19.4	20.6	20.1
Our method	**18.1**	**18.3**	**18.7**

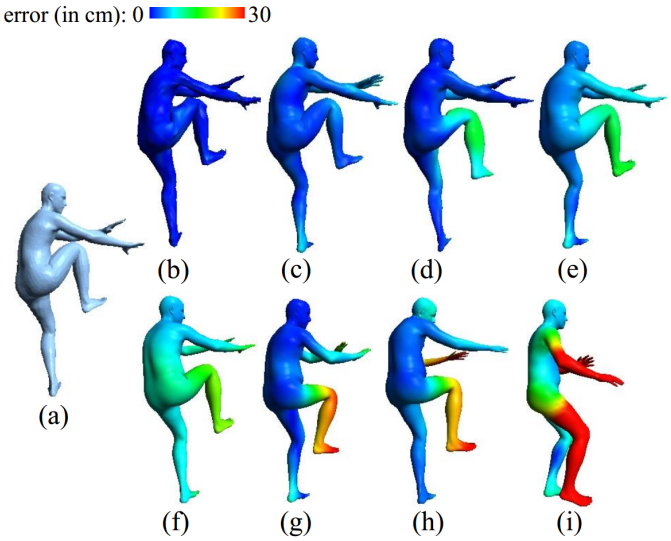


Fig. 5. The visualization of correspondence errors using different methods on the SURREAL data. (a) The input scan. (b) Our method. (c) Wang *et al.* [47]. (d) 3D-CODED [15]. (e) Kolotouros *et al.* [23]. (f) Kanazawa *et al.* [21]. (g) Wei *et al.* [52]. (h) Bogo *et al.* [6]. (i) The result of pure model fitting.

minima and cause large correspondence error in certain areas (e.g. right legs in Fig. 5). The predicted correspondences using [52] might be inaccurate by matching learned features because neighboring points probably are not classified into the same segmentation and multiple segmentations might fail to enforce smoothness of feature embedding, leading to large errors in certain areas of Fig. 5. In contrast, our method can predict more accurate correspondences for the point clouds.

We also compare our method to our recent work [47] which estimates the 3D human shapes from point clouds. Many deep learning methods of estimating 3D body models are proposed for a single color image. As in [47], we extend color images based methods for the comparison by adding a 3D correspondence loss defined as Eq. (11) to register the estimated models with the input depths. Kanazawa *et al.* [21] predicts the parametric SMPL models [30] from extracted features on a single color image. Kolotouros *et al.* [23] first attaches the extracted features to each vertex of a template mesh and then regresses the 3D meshes through Graph CNN. We re-train the network of [21] and [23] using depth images. The correspondence errors with different methods are listed in Table I. The comparison results on the SURREAL data using different methods are shown in Fig. 5. The comparison results demonstrate our method can obtain dense correspondences with higher accuracy than these model estimation methods. Since color images contain sufficient texture information of human bodies, the CNN can capture discriminative features of 2D joints to regress the body models. However, depth images only provide depth discrepancy over the body and the extracted features on depth images might not be discriminative as color images. Thus, there are relatively larger errors for the color images based methods [21] and [23]. Particularly, the correspondence error of our method is lower than that of our recent work [47] since the attention mechanism in the local feature embedding technique effectively improves the correspondence accuracy in the local regions of point clouds. This work mainly focuses on learning correspondence, while the recent work [47] recovers the 3D models.

C. Comparison to State-of-the-Art Methods of Non-Rigid 3D Shape Matching

We apply our method on non-rigid 3D shape matching and compare to several state-of-the-art methods. Here, we choose some representative methods for the comparisons which obtain state-of-the-art performance. FMNet [27] is a learning based functional map estimation method which requires ground truth point-wise maps. The unsupervised version, SURFMNet [41], enforces structural properties on the computed map. Unlike the approaches [27], [41], Deep Geometric Functional Maps (GeomFmaps) [14] learn the features directly from the geometry of the shapes and compute a robust functional map with a novel regularized estimation layer, which is a supervised method. We also compare to these methods with post-processing refinements (please refer to [14] for details of refinement methods). In addition, we compare our method to MoNet [35] which learns feature descriptors on graphs and manifolds and matches 3D shapes based on the descriptors. These methods are implemented with their released codes.

We use FAUST [7] and SCAPE [1] as the training and testing dataset which provide a variety of shape structures and connectivity. For all compared methods, we split the dataset into training and test sets containing 80 and 20 shapes for FAUST, and 51 and 20 shapes for SCAPE as in [14]. The correspondence accuracy is measured by geodesic error between the predicted and ground truth correspondences. We perform the experiments in two settings, i.e., complete shapes and partial shapes from a single view. It is noted that handling partial point clouds is more challenging than handling the complete shapes due to occlusions and data missing. Since all shapes are matched to a reference shape for MoNet [35], 3D-CODED [15] and our method in the experiment of complete shapes, we build correspondences between any two shapes using matched results with the reference shape as [15]. For the experiment of partial shapes, we match partial data to a complete reference shape and compute geodesic error on the reference shape. To generate partial data, we render each shape from 10 different views scattered evenly in a cycle around it. We follow the same data preprocessing and experimental settings in the original papers. The raw data are re-meshed for SURFMNet [41] and GeomFmaps [14] except FMNet [27] since the raw meshes are directly used in [27]. We do not perform data augmentation as FMNet [27] as the sampling method is not provided clearly in [27]. In original SURFMNet [41] and FMNet [27], the network is trained on all the data and the trained model is applied on the same test set. In MoNet [35], geodesic errors are computed between the input shape and the reference shape, while ours are computed between any two input shapes in the complete shape matching. The above factors lead to that some errors computed in our experiments are a bit different from that reported in the original papers.

Quantitative comparison of the different methods are shown in Fig. 6 on complete and partial data, respectively. The comparison results on complete data demonstrate our method can achieve much lower average error than these functional map based methods. These methods primarily operate in the spectral domain while our approach directly learns the correspondences in the spatial domain. The refinements, e.g. ZoomOut (ZO) refinement [34], are not adapted to our method. Without any post-processing refinement, our method still outperforms or at least is comparable to these compared methods with the refinements. MoNet [35] casts correspondence as a labeling problem, which tries to label each vertex on the test shapes with a vertex index on the reference shape. Rather than the nearest neighbor searching in other methods, most vertices have zero error, but the rest ones may have much higher error using MoNet [35]. Particularly, our method is more effective to handle partial point clouds than other methods. Two examples of matching partial data are shown in Fig. 7. Partial shapes pose challenges for computing the Laplace-Beltrami operators and SHOT features. The noisy initial descriptors lead to large error for functional map estimation methods.

More accurate correspondences can be obtained through non-rigid fitting based on the initially estimated correspondences of different methods. We perform the non-rigid fitting using Laplacian deformation proposed in [42]. Laplacian

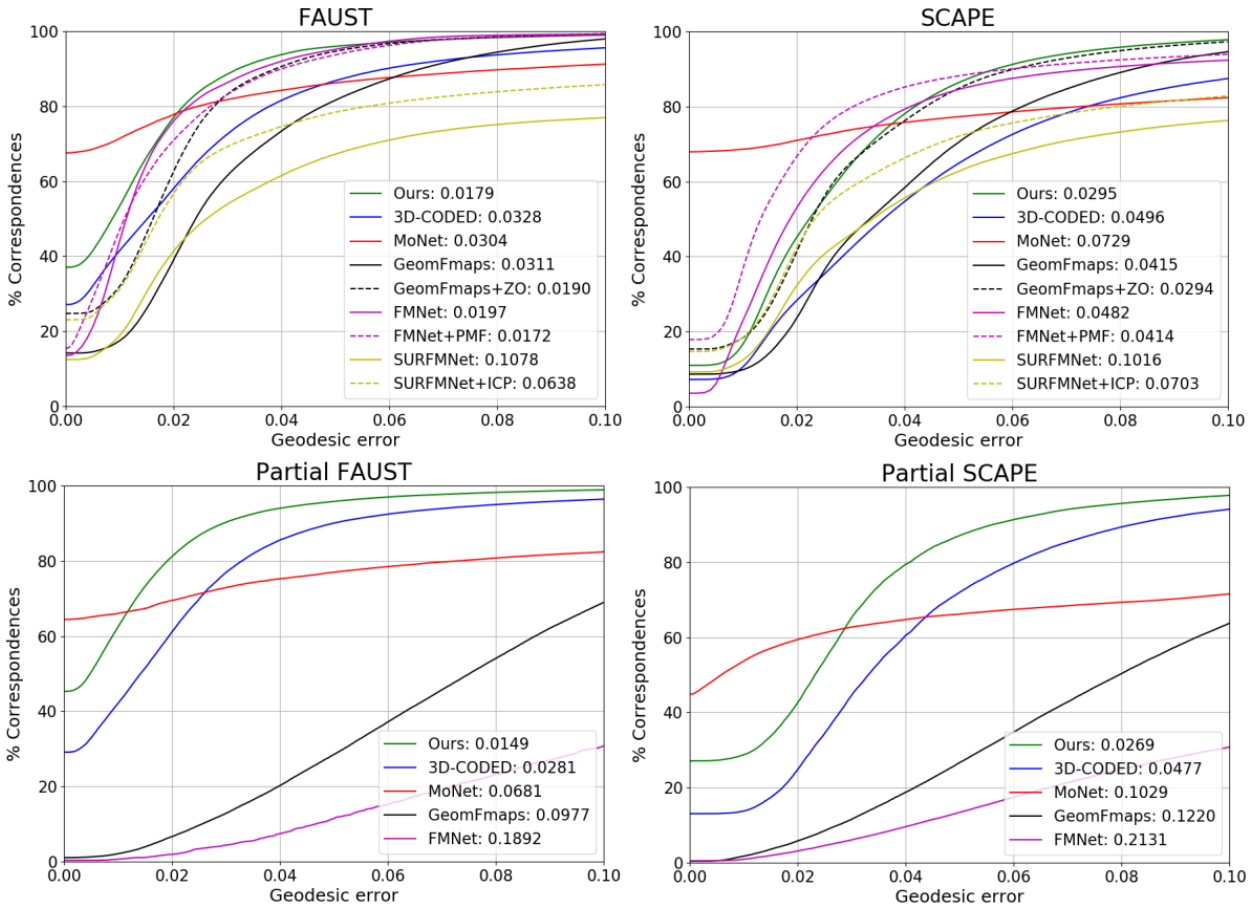


Fig. 6. Quantitative comparison of correspondence accuracy using different methods on complete and partial data, respectively. For the experiment on partial data, we only compare our method to the state-of-the-art methods that can handle partial data.

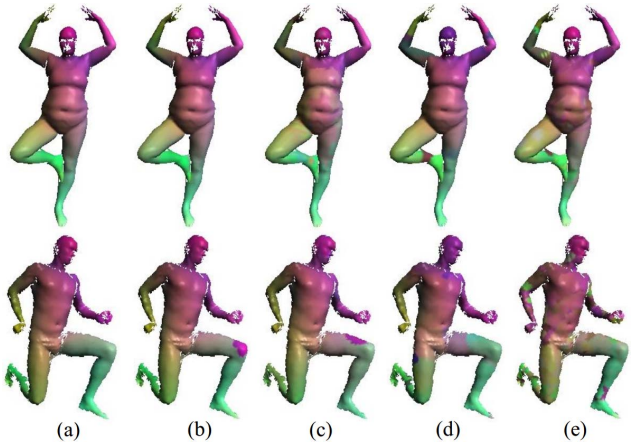


Fig. 7. Comparison examples of different methods on matching partial data of FAUST (top row) and SCAPE (bottom row). (a) Our method. (b) 3D-CODED [15]. (c) MoNet [35]. (d) GeomFmaps [14]. (e) FMNet [27]. Correspondence is visualized by the colors mapped from the template model shown in Fig. 1. The colorization of our method is more consistent with the template.

deformation [42] is formulated as minimizing the energy function consisting of a Laplacian smooth term and a correspondence term. With the initially estimated correspondences between the template model and the input point clouds, the SMPL template is deformed to the point clouds and a 3D model can be recovered that fits to the point clouds. The new correspondences are then built between the recovered model and the point clouds using the correspondence searching method described in Section III-D. After applying non-rigid fitting on the complete SCAPE data, the average geodesic errors decrease to 0.0085, 0.0102, 0.0286, 0.0082, 0.0093, 0.0234 for our method, 3D-CODED, MoNet, GeomFmaps+ZO, FMNet+PMF, and SURFMNet+ICP, respectively. The results show that the non-rigid fitting can result in better correspondence accuracy by taking more accurate initial correspondences as the input.

D. Comparison to 3D-CODED [15]

3D-CODED [15] presents a deep learning approach for predicting correspondences between the input shape and the template. Different from their method, which requires a relatively complete 3D shape, our method can handle the partial data captured from a single view with serious occlusion and data missing which is more challenging. Their method improves the correspondences by an additional optimization step, while we innovatively exploit local features on the point clouds and fuse them with graph features using an attention mechanism, largely improving the correspondence accuracy. In addition, compared to 3D-CODED, our model estimation method can handle the unseen real data more robustly and accurately. Our method can refine the unreasonable predictions of the networks by incorporating the parametric

TABLE II

CORRESPONDENCE ERRORS (mm) AT THE BEGIN AND END OF THE GLOBAL AND LOCAL REGRESSION ON THE SURREAL, HUMAN3.6M, AND DFAUST DATASET. WE ALSO LIST THE ERRORS OF APPLYING ANOTHER LOCAL REGRESSION

Two stages	SURREAL		Human3.6M		DFAUST	
	Begin	End	Begin	End	Begin	End
Global reg.	58.9	27.7	55.4	28.3	70.4	31.4
Local reg.	26.7	18.1	27.2	18.3	30.3	18.7
2×Local reg.	18.0	17.8	18.1	17.9	18.4	18.1

models and handle large deformation with joint information. In 3D-CODED, the optimization step is an optimization-based method and independent of the network in their first step. In contrast, all components of our methods are fulfilled with networks and composed in an unified framework, which achieves the efficient collaboration between the two-stage network fine-tuning and the pseudo label creation. We compare our method with 3D-CODED in the experiments of model estimation (Section IV-B) and shape matching (Section IV-C). The quantitative comparisons on our testing dataset are listed in Table I and that of shape matching are shown in Fig. 6, demonstrating that our method can achieve higher correspondence accuracy than 3D-CODED. Qualitative comparisons are shown in Fig. 5, Fig. 7 and Fig. 9. Particularly, 3D-CODED has large error on SCAPE dataset since human poses vary tremendously in the data. In contrast, our method can estimate more accurate correspondences for these data with large poses.

E. Ablative Analysis

1) The Two-Stage Regression Framework: We first evaluate the effectiveness of our two-stage regression framework by comparing the correspondence accuracy in the global and local regression network. The correspondence errors at the begin and end of each stage are listed in Table II on different datasets. The error in the begin of each stage is computed from the trained model after the first epoch. It can be seen that the correspondence error is reduced largely in each stage. We also observe that there is a noticeable improvement of correspondence accuracy with the local regression network compared to the global regression network. The qualitative comparisons of two stages are shown in Fig. 8 on DFAUST and SURREAL data. The results imply that our two-stage regression framework successfully achieves very accurate dense correspondences. After the local regression, we try to perform another one by decreasing the searching radius of neighboring points. Since the correspondence error declines little as shown in the third row of Table II, we perform the local regression only once in the experiments.

2) Local Feature Embedding: To investigate the effectiveness of local feature embedding (LFE), we report correspondence errors of three embedding methods that add the LFE layer on different graph resolutions, i.e., graphs with 1723, 431, and 27 vertices. These methods correlate local features of point clouds with the graph features on the finest, medium-resolution, and coarsest graph, respectively. In addition, we compare our method to the max pooling operation over the neighboring point features (defined in Eq. (4)) and another LFE by replacing the attention mechanism with a simply average operation. We also compare the results with and without LFE. Table III lists the correspondence errors of different methods on three test datasets. The correspondence accuracy is a bit higher by performing LFE on the finest graph (graph with 1723 vertices) since the local features are captured and exploited more sufficiently. The comparison results between our method with and without LFE show the correspondence accuracy is improved largely with LFE. Two comparison examples with and without LFE are shown in Fig. 8 on DFAUST and SURREAL data. Especially, compared to the max pooling and the simply averaging on the neighboring point features, the attention model can capture the local geometry structure better in point clouds.

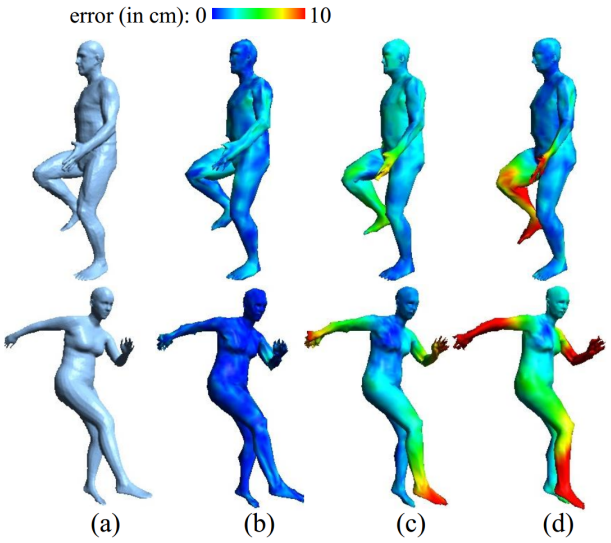


Fig. 8. The visualization of correspondence errors on DFAUST and SURREAL data using different methods in the ablation analysis. (a) The input depths. (b) The results of our method. (c) The results of our method without local feature embedding (LFE). (d) The results of the global regression. Our method with LFE can obtain more accurate correspondences in local regions than that without LFE.

TABLE III

COMPARISON OF APPLYING LOCAL FEATURE EMBEDDING (LFE) ON THE FINEST, MEDIUM-RESOLUTION AND COARSEST GRAPH, LFE WITH MAX POOLING AND SIMPLY AVERAGING, AND OUR METHOD WITHOUT LFE. NUMBERS ARE CORRESPONDENCE ERRORS IN mm

Methods	SURREAL	Human3.6M	DFAUST
Finest resolution	**18.1**	**18.3**	**18.7**
Medium resolution	21.3	20.8	21.7
Coarsest resolution	25.4	26.2	26.3
Max pooling LFE	21.0	21.6	22.1
Simply averaging LFE	22.7	23.5	23.2
Without LFE	24.9	26.8	27.8

3) The Number of Sampled Points: We also report the correspondence errors of our method on different numbers of sampled points $n = \{1000, 3000, 5000, 7000\}$. Table IV lists the correspondence errors by applying our method on different numbers of sampled points. The results demonstrate that our method can work well for different numbers of sampled points. There is a slight improvement on the performance with an increasing number of points. To balance between correspondence accuracy and training efficiency, we choose $n = 3000$ in our experiments. This experiment also verifies that our approach is robust to a small number of sampled

TABLE IV
CORRESPONDENCES ERRORS (mm) ON DIFFERENT NUMBER OF SAMPLED POINTS USING OUR METHOD

Point Number	SURREAL	Human3.6M	DFAUST
1000	20.9	21.4	21.7
3000	18.1	18.3	18.7
5000	18.0	18.1	18.3
7000	17.9	18.0	18.2

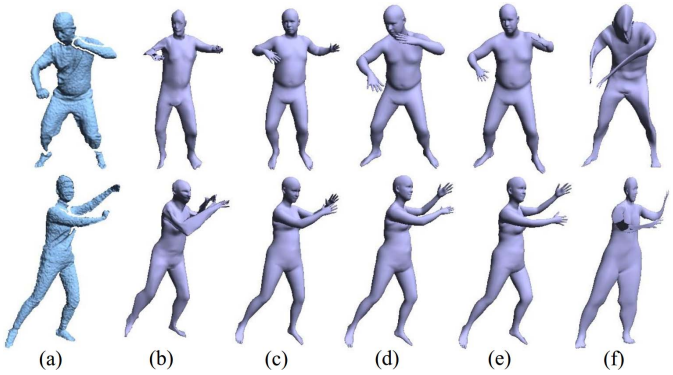


Fig. 9. The recovered results using our model estimation method on "Kungfu" data from [17] and "Girl" data. We also show the results of 3D-CODED [15] for comparison. (a) The input depths. (b) The resulting models using our two-stage regression. (c) The initial parametric models fitted to (b). (d) The refined parametric models fitted to (a) with both the correspondences and joints. (e) The refined models with the correspondences but without joints. (f) Predicted models using 3D-CODED based on Chamfer distance.

points since the correspondence accuracy does not reduce a lot with a decrease of point numbers.

4) Model Estimation on Real Data: In this test, we demonstrate the effectiveness of the proposed weakly-supervised model estimation method on real data. The model estimation method consists of two steps, i.e., initial estimation of parametric models and model refinement. Tested on real data, the proposed two-stage network can easily lead to unsatisfying results due to the lack of real training data. The initial estimation of parametric models (shown in Fig. 9 (c)) can correct the distorted models (shown in Fig. 9 (b)) predicted by the two-stage network. The model refinement can accurately align the initial estimation to the input data by searching accurate correspondences and utilizing joint information (shown in Fig. 9 (d)). Since minimizing the correspondence distance can only achieve local deformation, alignment failure still occurs under large shape discrepancy between the initial models and input data. Fortunately, the joint information can effectively alleviate the problem by handling large deformation. Fig. 9 (d) and Fig. 9 (e) show two examples of our method with and without joint information. Under serious occlusion and data missing in the partial data, we compute the correspondence distance by considering both point positions and normals, resulting in more accurate alignment compared to the Chamfer distance of 3D-CODED [15]. Fig. 9 (d) and Fig. 9 (f) show two examples of the final results using our method and 3D-CODED [15], respectively.

5) Model Estimation vs. Optimization-Based Model Fitting: In this experiment, we compare the proposed model estimation with the optimization-based model fitting in terms of reconstruction accuracy. Based on the state-of-the-art method [6], we perform model fitting with the same two steps in our model

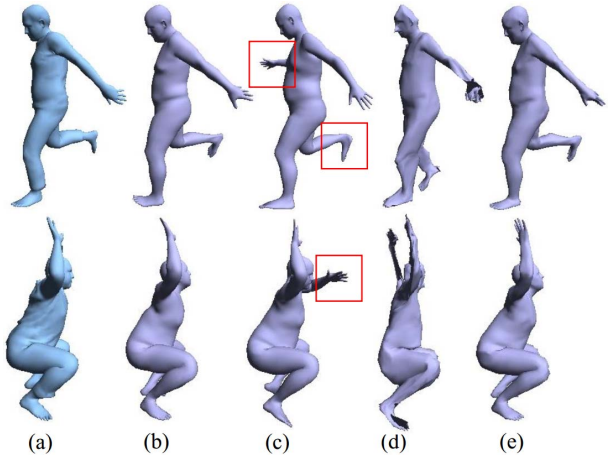


Fig. 10. Comparison examples between our model estimation and the optimization-based model fitting [6] on CAPE data. (a) The input depths. (b) The recovered models using our model estimation. (c) The recovered models using [6]. The resulting models of our two-stage regression before (d) and after (e) the fine-tuning, respectively. Some body parts cannot be recovered correctly using [6] (surrounded in rectangle) due to strong occlusions or large poses. In contrast, our method can perform more accurately and robustly.

estimation, i.e., initial estimation and model refinement. The CAPE dataset [32] are used in this test. The CAPE data is more challenging because the humans are dressed with various clothes. We select 8 subjects from the total 10 male subjects and uniformly sample 10, 000 models. Using the same way of dataset generation in Section IV-A, totally 150, 000 samples are rendered as the training data. We generate 5, 000 samples from the other 2 subjects as the testing data. The pre-trained two-stage regressor is applied on the CAPE test data to predict the initial correspondences. Based on the initial correspondences, the parametric models are then recovered by our model estimation method and the model fitting method [6], respectively. Since there are the corresponding ground truth SMPL models, we compute the reconstruction errors as in [47]. The average reconstruction errors are 27.7mm for our model estimation and 32.8mm for the model fitting method [6]. The model fitting method can work well for the data with relatively small poses. Nonetheless, the reconstruction error becomes larger when tested on samples with strong occlusions or large poses. The comparison results show that our model estimation can achieve higher reconstruction accuracy.

The optimization-based model fitting merely relies on the 2D joints and correspondences with the point clouds. In the cases with strong occlusions or large poses, human joints might be detected inaccurately and searched correspondences are prone to be unreliable. Also, the direct optimization is rather sensitive to the initialization of parameters. The inaccurate joints or improper initializations make direct optimization easily get trapped in local minima and lead to large fitting errors. As shown in Fig. 10 (c), the recovered models using direct optimization cannot fit to the input data well and even the body shapes are distorted. In contrast, deep neural networks have the stronger capacity of representing any complicated function, and modern techniques allow the deep networks to be successfully trained. By learning from sufficient training samples and leveraging deep features of

Fig. 11. Correspondence results using our method on real data. For each result, we show correspondence colorization on the depth image, the generated mesh with predicted 3D displacements, and the fitted parametric model to the input point clouds. From top to bottom: "Kungfu" [17], SCAPE [1], BUFF [57], and "Girl" data. Note that the input to our method is the sampled 3000 points from the depth image.

point clouds in predicting the 3D model, our method can robustly predict reasonable results even if the input joints are inaccurate due to strong occlusion or large poses. Thus, our model estimation can recover the 3D models more accurately and robustly. In addition, compared to computationally exhaustive optimization of model fitting methods, our method has considerably faster running efficiency. The inference time of our model estimation (14.5ms per frame) is hundreds of times faster than the model fitting method (11.35s per frame).

With the supervision of pseudo-ground truth correspondence produced in model estimation, we fine-tune the two-stage regression network on the CAPE training data. Computed on the test data, the average correspondence errors of our two-stage regression are 46.1mm and 28.4mm before and after the supervised fine-tuning, respectively. Two examples before and after the fine-tuning are shown in Fig. 10 (d) and Fig. 10 (e), respectively. The comparison results demonstrate that fine-tuning with the pseudo label can remarkably improve the performance of our two-stage regression on unseen real data.

F. Evaluation on Real Datasets

We test our method on various datasets of real human bodies. "Kungfu" data [17] and "Girl" data are captured by a Kinect V2 sensor. We also test on SCAPE [1] and BUFF dataset [57] to verify the performance of our method. Fig. 11 shows some correspondence results using our method on these real data. The input to our method is 3000 uniformly sampled points from the raw or rendered depth images. Our method directly predicts the 3D displacement vectors of template vertices to the corresponding points. A mesh can be generated

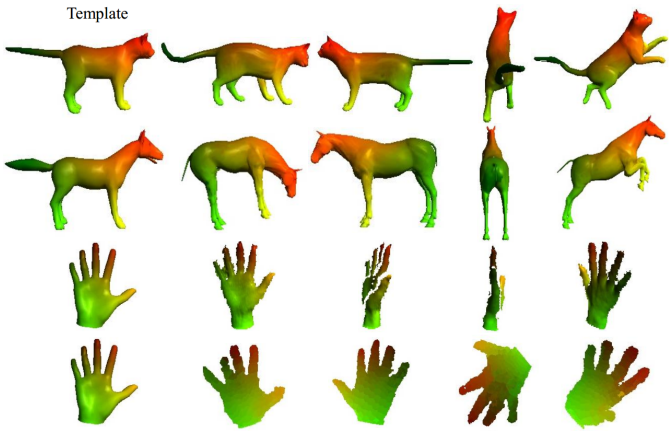


Fig. 12. Correspondence results using our method on animals and hands. From top to bottom: "Cat" from [11], "Horse" from [11], "hand" from [48] and [44], respectively. Correspondence colorization of each test sample is consistent with the template shown in the far left.

by adding the predicted displacement vectors to coordinates of the corresponding vertices. The dense correspondences for each depth are built by finding the nearest neighboring points on the generated mesh. Correspondences are visualized by the colors mapped from the template model. We also show the recovered parametric models that fit to the input point clouds for fine-tuning our network. Although there are strong self-occlusions and arbitrary deformations on the point clouds, our method still can robustly and accurately estimate accurate correspondences for deformable point clouds. The colorizations of predicted correspondences change smoothly over the body and are consistent between different point clouds. Through the proposed fine-tuning method, our method can generalize well to real point clouds. Our method may fail

on human bodies with loose clothes like long skirts. In the future, we plan to adopt an elaborate surface deformation technique to handle the real data with loose clothes.

Our method is also tested on other deformable subjects such as animals and hands. For the animals, we train on the synthetic data from SMAL [59] and test on TOSCA dataset [11]. For hands, we train on the synthetic data of FreiHAND [58] and test on the "Hand" data scanned using a Artec Scanner [48] and captured by a RGB-D sensor [44]. The "Hand" data from [44] have severer noise than that from [48]. Some examples of colorized correspondences are shown in Fig. 12. The results demonstrate that our method can generalize to various non-rigidly deformable targets effectively.

V. Conclusion

In this paper, we addressed the problem of estimating dense correspondences from non-rigid point clouds using deep neural networks. The correspondences are predicted in the form of 3D vertex displacements between vertices of a template mesh and the point clouds. We propose a two-stage regression framework which consists of the global regression and local regression network to estimate 3D displacements of template vertices using a graph CNN based hierarchical encoder-decoder network. The global regression network infers the global displacements from the concatenated global features of the template and the point clouds. In the local regression network, with the resulting mesh from initial displacements, local features of point clouds are captured and fused with graph features through an attention mechanism. The displacements can then be refined locally with the fused local features in the hierarchical encoder-decoder network. In addition, our method is generalized to handle real point clouds through a robust fine-tuning method. The experimental results on diverse datasets of various deformable subjects (e.g., human bodies, animals, and hands) demonstrate that the proposed method can achieve the state-of-the-art performance on the correspondence estimation of non-rigid point clouds.

References

[1] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: Shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, pp. 408–416, Jul. 2005.

[2] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 1626–1633.

[3] A. Behl, D. Paschalidou, S. Donne, and A. Geiger, "PointFlowNet: Learning representations for rigid motion estimation from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7962–7971.

[4] M. Belkin, J. Sun, and Y. Wang, "Constructing Laplace operator from point clouds in $\mathbb{R}^d$," in *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2009, pp. 1031–1040.

[5] F. Bogo, M. J. Black, M. Loper, and J. Romero, "Detailed full-body reconstructions of moving people from monocular RGB-D sequences," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. .

[6] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 561–578.

[7] F. Bogo, J. Romero, M. Loper, and M. J. Black, "FAUST: Dataset and evaluation for 3D mesh registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3794–3801.

[8] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black, "Dynamic FAUST: Registering human bodies in motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6233–6242.

[9] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Comput. Graph. Forum*, vol. 34, no. 5, pp. 13–23, Aug. 2015.

[10] D. Boscaini, J. Masci, E. Rodola, and M. M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3189–3197.

[11] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008.

[12] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[14] N. Donati, A. Sharma, and M. Ovsjanikov, "Deep geometric functional maps: Robust feature learning for shape correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8592–8601.

[15] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry, "3D-CODED: 3D correspondences by deep deformation," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 230–246.

[16] P. Guan, A. Weiss, A. O. Balan, and M. J. Black, "Estimating human shape and pose from a single image," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1381–1388.

[17] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai, "Robust non-rigid motion tracking and surface reconstruction using l0 regularization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3083–3091.

[18] O. Halimi, O. Litany, E. R. Rodola, A. M. Bronstein, and R. Kimmel, "Unsupervised learning of dense shape correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4370–4379.

[19] X. He, T. Huang, S. Bai, and X. Bai, "View N-Gram network for 3D object retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7515–7524.

[20] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[21] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7122–7131.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[23] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4501–4510.

[24] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4990–5000.

[25] C.-L. Li, T. Simon, J. Saragih, B. Poczos, and Y. Sheikh, "LBS autoencoder: Self-supervised fitting of articulated meshes to point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11967–11976.

[26] Z. Li, T. Yu, C. Pan, Z. Zheng, and Y. Liu, "Robust 3D self-portraits in seconds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1344–1353.

[27] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5660–5668.

[28] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 171–180, Jan. 2014.

[29] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 529–537.

[30] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, Nov. 2015.

[31] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepVCP: An end-to-end deep neural network for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 12–21.

[32] Q. Ma *et al.*, "Learning to dress 3D people in generative clothing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6469–6478.

[33] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," 2015, *arXiv:1501.06297*. [Online]. Available: http://arxiv.org/abs/1501.06297

[34] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov, "ZoomOut: Spectral upsampling for efficient shape correspondence," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–14, Nov. 2019.

[35] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5115–5124.

[36] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: A flexible representation of maps between shapes," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–11, Aug. 2012.

[37] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3D human pose and shape from a single color image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 459–468.

[38] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon, "Metric regression forests for correspondence estimation," *Int. J. Comput. Vis.*, vol. 113, no. 3, pp. 163–175, Jul. 2015.

[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[40] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 725–741.

[41] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov, "Unsupervised deep learning for structured shape matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1617–1627.

[42] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proc. Eurograph./ACM SIGGRAPH Symp. Geometry Process. (SGP)*, 2004, pp. 179–188.

[43] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, Jul. 2009.

[44] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall, "Capturing hands in action using discriminative salient points and physics simulation," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 172–193, Jun. 2016.

[45] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, and I. S. C. Laptev, "BodyNet: Volumetric inference of 3D human body shapes," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 20–36.

[46] G. Varol *et al.*, "Learning from synthetic humans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 109–117.

[47] K. Wang, J. Xie, G. Zhang, L. Liu, and J. Yang, "Sequential 3D human pose and shape estimation from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7275–7284.

[48] K. Wang, G. Zhang, and S. Xia, "Templateless non-rigid reconstruction and motion tracking with a single RGB-D camera," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5966–5979, Dec. 2017.

[49] K. Wang, G. Zhang, J. Yang, and H. Bao, "Dynamic human body reconstruction and motion tracking with low-cost depth cameras," *Vis. Comput.*, vol. 37, no. 3, pp. 603–618, 2020).

[50] Y. Wang and J. Solomon, "PRNet: Self-supervised learning for partial-to-partial registration," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–13.

[51] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3523–3532.

[52] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, "Dense human body correspondences using convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1544–1553.

[53] Y. Yao *et al.*, "BlendedMVS: A large-scale dataset for generalized multi-view stereo networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1709–1799.

[54] T. Yu *et al.*, "BodyFusion: Real-time capture of human motion and surface geometry using a single depth camera," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 910–919.

[55] T. Yu *et al.*, "DoubleFusion: Real-time capture of human performances with inner body shapes from a single depth sensor," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7287–7296.

[56] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, "3D human mesh regression with dense correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7054–7063.

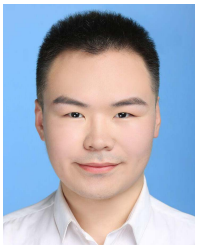
[57] C. Zhang, S. Pujades, M. Black, and G. Pons-Moll, "Detailed, accurate, human shape estimation from clothed 3D scan sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4191–4200.

[58] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, and T. Brox, "FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 813–822.

[59] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3D menagerie: Modeling the 3D shape and pose of animals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6365–6373.

Kangkan Wang (Member, IEEE) received the B.S. degree in computer science from Northwestern Polytechnical University in 2009 and the Ph.D. degree from the State Key Lab of CAD&CG, Zhejiang University, in 2015. He is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. Before that, he was an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include human body reconstruction, 3D reconstruction, non-rigid reconstruction, and motion tracking.

Guofeng Zhang (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, in 2003 and 2009, respectively. He is currently a Full Professor with the State Key Laboratory of CAD&CG, Zhejiang University. His research interests include structure-from-motion, SLAM, 3D reconstruction, augmented reality, video segmentation, and editing. He was a recipient of the National Excellent Doctoral Dissertation Award and the Excellent Doctoral Dissertation Award of the China Computer Federation.



Huayu Zheng received the B.S. degree from Shandong University in 2019. He is currently pursuing the master's degree with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include 3D reconstruction, human pose estimation, and points cloud processing.

Jian Yang (Member, IEEE) received the Ph.D. degree from Nanjing University of Science and Technology (NUST) in 2002. In 2003, he was a Postdoctoral Researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at the Biometrics Centre, The Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at the Department of Computer Science, New Jersey Institute of Technology. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He is the author of more than 200 scientific articles in pattern recognition and computer vision. His papers have been cited more than 6000 times in the Web of Science and 15000 times in the Scholar Google. His research interests include pattern recognition, computer vision, and machine learning. He is a fellow of IAPR.