OPEN: Order-preserving Pointcloud Encoder Decoder Network for Body Shape Refinement

Haidong Zhu*, Ye Yuan†, Yiheng Zhu†, Xiao Yang† and Ram Nevatia*
*Department of Computer Science, University of Southern Califormia, Los Angeles, California 90089
†Bytedance Inc., Mountain View, California 94041
haidongz@usc.edu

***Abstract*—Image-based 3-D human body shape estimation and reconstruction have shown significant improvement by using deep neural networks. Compared with reconstructing from a single image, reconstructing 3-D human body shapes from video or image sequences requires high precision and dense correspondences between the keypoints of the reconstructed shape sequence. Existing methods cannot achieve both high accuracy and keep the dense correspondence between different shapes after reconstruction. In this paper, we propose a method named Order-preserving Point cloud Encoder-decoder Network to refine the reconstructed human body shape from SMPL with the assistance of RGB images while preserving its original dense correspondence. We further introduce using 2-D RGB images as weak supervision when 3-D labels are not available. We assess our methods on the public dataset and show improved results compared with the baseline methods.**

I. Introduction

Reconstructing the 3-D human body shapes from a single image has many applications such as animation, action recognition and human ID. For a video sequence, it is also useful to make correspondence among the 3D points for pose tracking. There has been outstanding progress in methods of reconstructing 3D, but in previous work, there has been a conflict between high reconstruction accuracy for single pose and high correspondence-preserving for varying poses. In this paper, we present a technique that achieves both goals.

To reconstruct the whole human body shape for 2-D images, the representation of the human body should precisely describe the shape. Existing approaches can be split into two categories: parametric methods (SMPLify [1] and SPIN [2]) which describe the human body shape and pose with pre-defined human body reconstructed shapes, and non-parametric methods (such as PIFu [3] and PIFuHD [4]) with implicit functions for reconstruction.

For parametric methods, we take SPIN[2] as an example. It reconstructs human body shape from the 2-D image into the format of SMPL [5] which is a pre-defined human body template. With the position of the joints and the shape parameters, SMPL constructs a shape with 6,890 vertices and 13,776 faces. Each vertex has its specific location on the human surface and k-th point on different shapes represents the same body position if k is the same. Since SMPL only considers the surface of the human, meshes generated by SPIN fail to describe the precise shape with clothes. Since we need to warp the appearance from the 2-D image, if the reconstructed shape cannot align with the border of the clothes, the sampled pixel is inaccurate for the appearance.

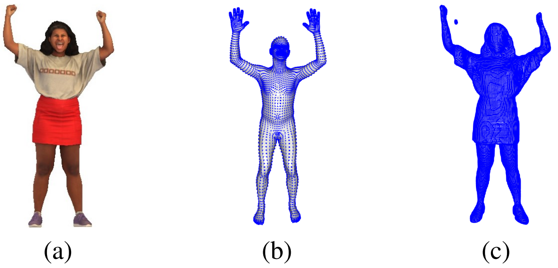

Fig. 1. Example of the (a) original image and reconstructed shape of (b) SPIN and (c) PIFu. Vertices of the shape are represented as blue points.

Compared with SPIN, PIFu [3] and PIFuHD [4] better describe the outer body shape with single RGB images as input. These methods reconstruct the whole human body shape by inferring if points sampled in 3-D space locate inside or outside the human body; a Marching Cubes [6] algorithm is then applied to inside points to reconstruct the boundary of the 3-D mesh. Compared with parametric representations, these methods can have a much more fine-grained shape for the human. Examples of both are shown in Figure 1. The result from SPIN includes every component in the reconstructed human body shape and each vertex has its specific corresponding location, while the reconstruction of PIFu includes many more vertices and can better describe the shape in a 2-D image.

Different from directly reconstructing meshes from a single 2-D image, reconstructing 3-D human body shape sequence from an image sequence or video also requires dense correspondence between the points on different shapes at different timestamps. Although PIFu [3] and PIFuHD [4] can reconstruct more accurate human body shapes, the points do not have a direct correspondence. In contrast, SMPL [5] representation preserves the dense correspondence between different reconstructed shapes.

We propose a network named OPEN, which is the abbreviation of **O**rder-preserving **P**oint cloud **E**ncoder-decoder **N**etwork. With the assistance of the RGB image, we treat the human body shape reconstructed by SMPL as an ordered point cloud and improve its accuracy without disturbing its original order and relative position. Since every vertex in SMPL [5] has its specific representation, by keeping its original order, dense correspondence across views is preserved. In addition, besides using 3-D labels, we also introduce using 2-D images for weak supervision on aligning the reconstructed shapes with the human body segmentation masks on the *xy* plain. Acquiring 3-D labels is expensive. By aligning with 2-D RGB images, we can use the 2-D RGB images which do not have any 3-D shape labels to increase the variety of the available data and improve its performance. The extra depth information can be dropped since if we add or subtract the same value to the depths for all the points, the overall shape still precisely describes the person in the image. We test our method on Renderpeople [7], which is a standard 3-D reconstruction dataset. We show better results than state-of-the-art methods that can generate dense correspondence for the shapes and comparable results with the best outer shape reconstruction method.

In summary, our contributions are as follows: 1) we introduce using OPEN to generate fine-grained ordered point clouds without disturbing the dense correspondence, and 2) we introduce using 2-D RGB images for weak supervision on reconstructing the ordered point clouds for improving the reconstruction accuracy.

II. Related Work

In this section, we describe the latest research of 3-D shape reconstruction for objects as well as human body shapes.

3-D Object Reconstruction: Reconstruct the 3-D object shape is to establish a representation of the 3-D shape for the object. Such reconstruction is useful for tasks such as understanding, segmentation and detection. Researchers have different ways of representations for reconstructing the 3-D object shape, such as point cloud representation [8], [9], mesh representation [10], [11], [12] and voxel-based represnetation[13], [14], [15], [16], [17], [18], [19]. Point cloud representation records the position of some sampled points on the surface of the object. Mesh representation records the triangles of the surface of the object. The three angles for every triangle are the vertices on the surface of the object and the triangle itself is part of the surface. Voxel-based representation records whether, in the cubic space, the voxel is occupied or not with binary representation. All these methods record the position of the voxels or the surface in the representation. If we require a higher resolution of the object, all these methods require more space to store more points or voxel information introduced by the higher resolution. Recent methods [20], [2], [21], [22], [23], [24] use the implicit functions for representing the 3-D object shape. Instead of storing the locations of the points or surface of the object, implicit functions store the object shape in the format of a latent vector or a latent code. To reconstruct the shape, authors use a pretrained model as the decoder function $f(\cdot)$ to estimate whether the queried points are inside or outside of the object or their distance to the surface, followed by the surface reconstruction methods, such as Marching Cubes [6], to reconstruct the object surface. Compared with traditional representations, implicit functions do not need extra storage space for higher resolution of the object reconstruction.

3-D Human Body Reconstruction: Unlike 3-D object reconstruction, estimating the 3-D human body shape focuses on reconstructing the details of human body shape with prior knowledge since most human components are similar to others. Thus researchers use pre-defined models to model the whole human body shape [5], [25], [1], [26], [2]. These methods consider the whole human body shape and generate the human body shape by defining parametric locations of important joints. SMPL [5] records the rotation of 23 joints and a 10-D vector for the body shape. With these parameters, SMPL is able to reconstruct a mesh with 6,890 vertices where each vertex has its specific locations. SMPL-X [26] and STAR [25] include more points for more detailed descriptions and decouple the points from other parts on the body shape, and the movement of one part will not have an effect on others. Based on the SMPL template, SPIN [27] uses an automatic system for fitting the human body into the SMPL shape. There are some latest non parametric methods, such as PIFu [3] and PIFuHD [4]. Similar to implicit representations, PIFu uses the latent space for deciding whether the point in the space is inside or outside the body mesh. Different from parametric methods, PIFu [3] and PIFuHD [4] apply the implementation representation for more finegrained descriptions, while the vertices on its surface do not have specific meaning and cannot find corresponding points on other shapes, since it doesn't have prior knowledge for human shape in reconstruction.

III. Methods

To improve the quality of the reconstructed human body mesh without disturbing the dense correspondence between points of different shapes, we introduce refining the original SMPL [5] mesh for a more accurate description. For every 2-D RGB image $\{v_i\}_{i=1}^{N}$, where i represent the i-th image in the sequence, we take the reconstructed mesh from SMPL [5] model, which is annotated as s_i, as input. The mesh s_i consists of two parts: N_s vertices $\{e_i^s\}_{s=1}^{N_s}$ and the faces f_i connecting these vertices, which are represented as the triangles. Each triangle records three vertice IDs as three angles and forms a patch of the object's surface. Since the SMPL [5] representation is a parametric method, each vertex e_i^s locates at a specific point on the human body. Thus the IDs of the vertices used to construct the triangles for f_i are the same for all reconstructed human body mesh. Instead of directly modifying the mesh, we can use the point cloud representation and improve the point position for improving the mesh accuracy.

A. *Order-preserving Encoder-decoder Network*

Unlike the existing point cloud representations, where the points do not have a specific order, we design an order-preserving encoder-decoder network for pointwise modification without disturbing the original order of the input point clouds. The encoder-decoder aims to take the ordered points $\{e_i^s\}$ as input and generate the corresponding new location $\{e_i^s\}'$ in the sequence of the original input. We show our proposed encoder-decoder in Figure 2.

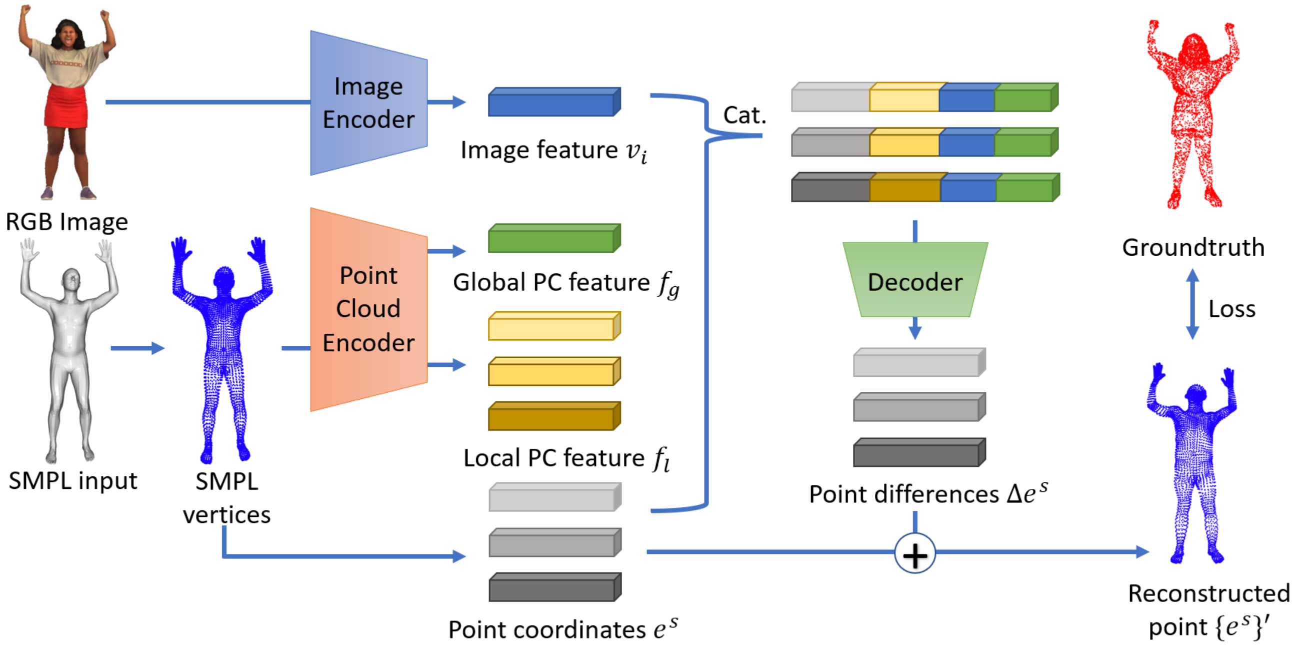

Fig. 2. Our proposed architecture for order-preserved point cloud encoder-decoder network. We take the SMPL model input and RGB image as input and generate the corresponding point difference Δe^s for every corresponding input vertices e^s. 'Cat.' is the abbreviation of concatenation. Feature in the same color are the same feature vector.

1) Encoders: OPEN has two encoders: a point cloud encoder and an image encoder. We followe the FoldingNet [28] encoder for building the point cloud encoder. For each shape, the point cloud encoder consumes the N_s-by-3 point cloud matrix e_i^s and generate two different features, a N_s-by-1024 local feature f_l for every point and a 1-by-512 global feature f_g for the global point cloud representations. We stack two multi-layer perceptrons (MLP) for feature encoding, where the first MLP consumes the e_i^s with its covariance. The shared three-layer perceptron encodes the input feature with batch normalization and ReLU to obtain N_s point-wise features. After that, we apply two graph layers to get more local structural information. The global feature f_g is extracted with a max-pooling layer. For the image encoder, we use a pretrained ResNet-18 [29] model to extract a 1-by-512 vector, which is the last feature layer of the ResNet-18 network, as the image-level feature representation f_i for each input 2-D RGB image v_i.

2) Decoder: The decoder network is to generate the modified ordered locations of the point sequence $\{e_i^s\}'$ corresponding to the original point locations $\{e_i^s\}$. We first concatenate the 1-by-512 global point feature f_g with the 1-by-512 RGB image feature v_i and construct a 1-by-1024 feature as the global representation. We replicate this feature vector by N_s times and concatenate it with the N_s-by-3 point locations e_i^s and N_s-by-1024 local feature f_l and construct a N_s-by-2051 feature matrix for decoding $\{e_i^s\}'$. In this way, features used to decode each vertex of $\{e_i^s\}'$ consists both local features from the corresponding input point $\{e_i^s\}$ as well as the global bottleneck feature f_i and f_g, which helps to decode the point while preserving its original input order. We use a two-layer MLP for decoding the difference of vertexs $\Delta\{e_i^s\}$ between the output and the input point clouds. Instead of directly generating the location of point cloud $\{e_i^s\}'$, $\Delta\{e_i^s\}$ is relatively small and easier to learn. We add $\Delta\{e_i^s\}$ and $\{e_i^s\}$ to generate the final locations $\{e_i^s\}'$ of the modified point clouds.

B. Objectives

Due to the lack of 3-D mesh labels, we use 3-D point clouds from labeled mesh for supervision and apply the 2-D point clouds sampled from RGB images for weak supervision in addition. We have two different splits in the training dataset: (I) images with point cloud annotations and (II) images without point cloud annotations. To train an order-preserving autoencoder network, we calculate the distance between the reconstructed point cloud $\{e_i^s\}'$ and the points in the corresponding groundtruth mesh $\{v_{gt}\}$ for as our objective function. We have both 2-D and 3-D supervision for the point cloud reconstruction.

1) 3-D supervision: For the images with labeled 3-D point cloud annotations, we use the Chamfer Distance [30] between the reconstructed point cloud $\{e_i^s\}'$ and groundtruth $\{v_{gt}\}$ to calculate the similarity between two sets of point clouds. In addition to the Chamfer distance, we also use a L_2 distance between the corresponding points of the reconstructed points $\{e_i^s\}'$ and original points $\{e_i^s\}$. L_2 distance is to minimize the absolute values of the $\Delta\{e_i^s\}$ to ensure the order between different points is preserved. We introduce a hyperparameter λ as the weight of the L_2 loss, which gradually decays to 0 along with the whole training process.

2) 2-D supervision: In addition to the 3-D supervision, we introduce the 2-D supervision for applying the images without

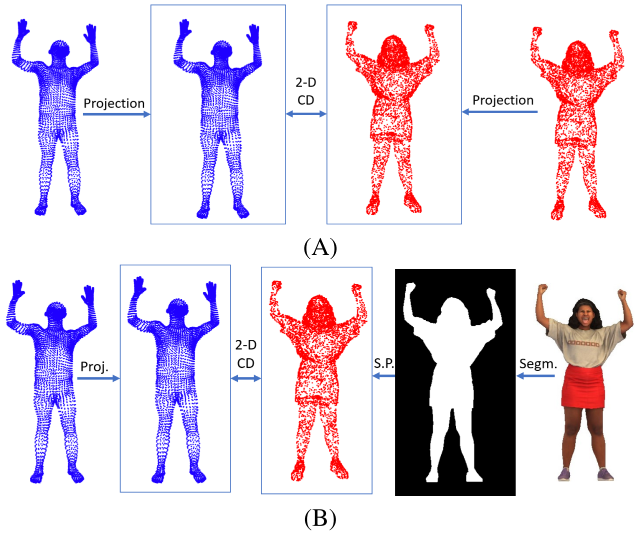

Fig. 3. Generating 2-D projection data for (A) examples with 3-D groundtruth point cloud and (B) examples without 3-D point cloud labels. Shapes consist of blue points are reconstructed points, while the red points are original 3-D labels or generated 2-D points from RGB images. 2D CD, proj, segm, S.P. are the abbreviations of 2-D chamfer Distance, projection, segmentation, and sampling points respectively. If the 3-D annotation is available in the original data, we directly project the groundtruth on a 2-D image; if it is not available, we randomly sampled points from the segmentation mask to generate the synthetic projection of shape. Points enclosed by a blue box are the 2-D points with only *xy* axis available.

labeled 3-D point cloud groundtruth for training. To align the data with and without point cloud annotations, we use the 2-D supervision for both two splits. For the images with point cloud annotations, we project both generated point cloud $\{e_i^s\}'$ and groundtruth points $\{v_{gt}\}$ onto the 2-D space and calculate the 2-D Chamfer Distances between two clusters following Figure 3 (A). For the images without 2-D point cloud annotations, we show the pipeline in Figure 3 (B). We first extract the segmentation mask for the human from the image and then sample points inside the segmentation mask as the projected 2-D points. We then calculate the 2-D Chamfer Distance between the project points of generated mesh $\{e_i^s\}'$ and the sampled points.

In summary, our loss function is as follows

$$\begin{aligned} L_{overall} = & L_{CD,2}(\{e_i^s\}_p, \{v_{gt}\}_p^*) \\ & + \delta L_{CD,3}(\{e_i^s\}, \{v_{gt}\}) + \lambda L_2(\Delta e_i^s) \end{aligned}$$

where $L_{CD,k}$ is the chamfer distance on the k dimension, λ is a hyperparameter, which we set to 0.1 in the experiment. δ is an activation function, which is 1 if 3-D labels are available and 0 if not.

IV. Experiments

A. Dataset

In our experiments, we use three different splits from two datasets for training: Rigged and Posed subsets of the RenderPeople dataset [7] and TT-500 dataset, and evaluate on the test split of RenderPeople [7] Posed set.

Renderpeople Posed dataset [7] is a dataset which consists of commercially available 500 high-resolution photogrammetry scans. It contains a front-view RGB image as well as the rendered 3-D human body mesh for every existing image. Each shape is a posed data where the appearance and pose are different from other data in the dataset. We split it into a 400 training set and 100 test set. Since all 3-D human body shape annotations are available during training, we use this split for supervised 3-D supervision. We evaluate all our experiments on the test set on their front view.

Renderpeople Rigged dataset [7] is another sub-dataset of the RenderPeople dataset. It also consists of 500 high-resolution scans for 500 different humans. All these people share the same pose while preserving their different appearances. Due to the lack of pose information in the original dataset, we only use this dataset as the extension training dataset for synthetic projected 2-D supervision and apply all 500 examples for weak supervision in addition to Renderpeople Posed Dataset.

TT-500 dataset is the abbreviation of the Tiktok Dancing 500 dataset, which is a dataset with 500 dancing videos on TikTok whose length ranges from 6 seconds to 50 seconds. Each video consists of a front view of a dancing video sequence. We sample the from every 100 frames and extract the image dataset with 2,245 images. Note that the images in this dataset do not have 3-D labeled mesh. We use these images as the extension of the RenderPeople Posed Dataset for weak supervision for increasing the variety of the data available during training.

B. Experiment setups

In this subsection, we mainly discuss the details of the experiments. We also discuss the criteria and baselines we compared for the experiments. We present our implementation details in the supplementary material.

1) 3-D reconstruction from 2-D images: To generate the 3-D input human body mesh s_i from the single 2-D image, we use the SMPLify [1] to fit the 3-D human body shape from the 2-D image. The output of the SMPLify [1] is an SMPL model, with 6,890 vertices and 13776 faces containing the vertices. We sample 6,890 vertices from the SMPL model and recenter it to the center of a cubic. We resize it to make sure the high of the model is precisely two and resize the other two dims in the same ratio accordingly.

2) 2-D supervision: For images with 3-D annotations as groundtruth, we directly project the 3-D point cloud onto the 2-D *xy* plain and set all numbers on z dim to 0. To sample points for 2-D Chamfer Distances calculation without *xy*, we first use a pretrained HR-Net [31] to extract the segmentation mask of a human body. After that, we sample all the pixels that are covered by the segmentation mask. Since the pixel is always represented as integers, we first erode the mask for a pixel and then randomly generate a float number between -0.5 and 0.5. After that, we normalize the points to zero means and normalize the height of the image to make its length as 2, with other dimensions resized accordingly. We then shuffle all the points and sampled 8,960 points from the point polls. Since we align both of the projected 3-D points and synthetic 2-D points, these two points are already in the same domain, and we do not need external alignment between the projected and sampled 2-D points.

3) Metrics: We evaluate our methods on two different metrics between the reconstructed shape and the groundtruth shape: Chamfer Distance (CD) for evaluating the similarity between two groups of points and mesh accuracy for evaluating the similarity between two meshes.

For two sets of point clouds, S_1 and S_2, Chamfer distance is calculated as

$$\begin{aligned} ChamferDist(S_1, S_2) = &\sum_{x \in S_1} \min_{y \in S_2} ||x-y||^2 \\ &+ \sum_{x \in S_2} \min_{y \in S_1} ||x-y||^2 \end{aligned}$$

This is to find the nearest point in S_1 from S_2 and in S_2 from S_1 and sum their distances up to find the similarity between two different sets of points in the point clouds. $||\cdot||^k$ is the k-th norm of the value. Mesh accuracy is a distance where 90% of the gaps between sampled points and mesh is smaller than this value, which is defined as

$$MeshAcc(S_1, S_2) = d, s.t. : \frac{N_{p, dist(p, S_2) < d}}{N_p} \geq 0.9, p \in S_1$$

where S_1 is the groundtruth shape and S_2 is the reconstructed shape. N_x is the number of points of the set x and $dist(\cdot)$ is the L2 distance between the sampled points from the reconstructed mesh and the groundtruth mesh.

4) Baseline methods: In our experiments, we make a comparison between three other methods. We use the SMPL model reconstructed by SPIN and the SMPL-X model reconstructed by SMPLify-x for comparisons for the methods that preserve the dense correspondence. SMPL-X model is also a parametric representation, while it contains more points for further description of the shape. We also make a comparison with the finegrained methods PIFu [3], which doesn't preserve the original dense correspondence of the human body shape between different images. We don't compare with PIFuHD [4] since its training code is not publicly available.

C. Results and Analysis

In this subsection, we show both numerical and visualization results. We will discuss some ablation studies in the supplementary material.

1) Quantitative Results: We show our results in Table I. We have two different settings for our OPEN model (supervised training *v.s.* supervised w/ weakly supervised training on external datasets). We have the following observations:

i) **Compared with baselines:** Compared with existing state-of-the-art methods with 3-D dense correspondence like SPIN [5] and SMPLify-X [1], we notice that our methods have better performance on both of the experiment criteria. For example, on our Renderpeople test set, the Chamfer distance for SMPLify and SMPLify-X are 2.29 and 2.17. After refined with OPEN, the Chamfer Distance of the generated mesh becomes 0.60, which is much better than two state-of-the-art methods with dense correspondence. Our model even outperforms the reconstruction results from PIFu on Mesh Acc and shows similar performance on Chamfer Distance after we retrain the PIFu model on our data split. Although OPEN doesn't show big improvement compared to PIFu on general shapes, the prior knowledge of body shape from the input SMPL model helps construct some detailed structures, such as the fingers, which PIFu fails to reconstruct. With the assistant of the human shape knowledge from the input SMPL shape, our network is able to reconstruct much more finegrained and accurate shapes.

TABLE I

Results for RenderPeople [7] compared with existing state-of-the-art methods. Since the training split for the training set is different, we retrain PIFu on our data split. Rigged is the rigged posed data used for weak supervision, and TT-500 dataset. We time both of the criteria with 10^2 for better comparison. ($\downarrow$) indicates the smaller value, the better performance a model has.

Method	CD ($\downarrow$)	Mesh Acc ($\downarrow$)
SPIN [2]	2.29	0.59
SMPLify-X [1]	2.17	0.53
PIFu [3]	0.57	0.37
OPEN	0.60	0.33
OPEN w/ Rigged	0.53	0.29
OPEN w/ TT-500	0.55	0.21

ii) **2-D supervision:** Instead of only training the network with labeled 3-D data on the Renderpeople dataset, we also show improvement on both Chamfer Distance and Mesh Accuracy when using unlabeled RenderPeople Rigged dataset or TT-500 dataset for 2-D weakly supervision. Since the rigged data for RenderPeople share the same pose with different appearances, we show that both including more variation of body shape (comparing OPEN with OPEN w/ Rigged) and more human body pose (comparing OPEN with TT-500 dataset) are helpful for improving the performance of the final reconstructed models with only projected 2-D synthetic supervision. The use of external 2-D weak supervision can increase the variety of the data used during training and improve the ability of generalization.

2) Qualitative Results: We show three reconstructed visualization results of the 3-D reconstructed human body shapes from 2-D RGB images in Figure 4 on the examples of the Renderpeople Posed dataset test split. We compare our method OPEN with two different state-of-the-art methods: PIFu and SMPL reconstructed by SMPLify. In addition, we plot the 6,890 points of the input and the output for the OPEN model with different colors to visualize whether the order of the points or the correspondence between points from different shapes has changed after reconstruction.

Compared with PIFu, SMPLify can reconstruct some de-

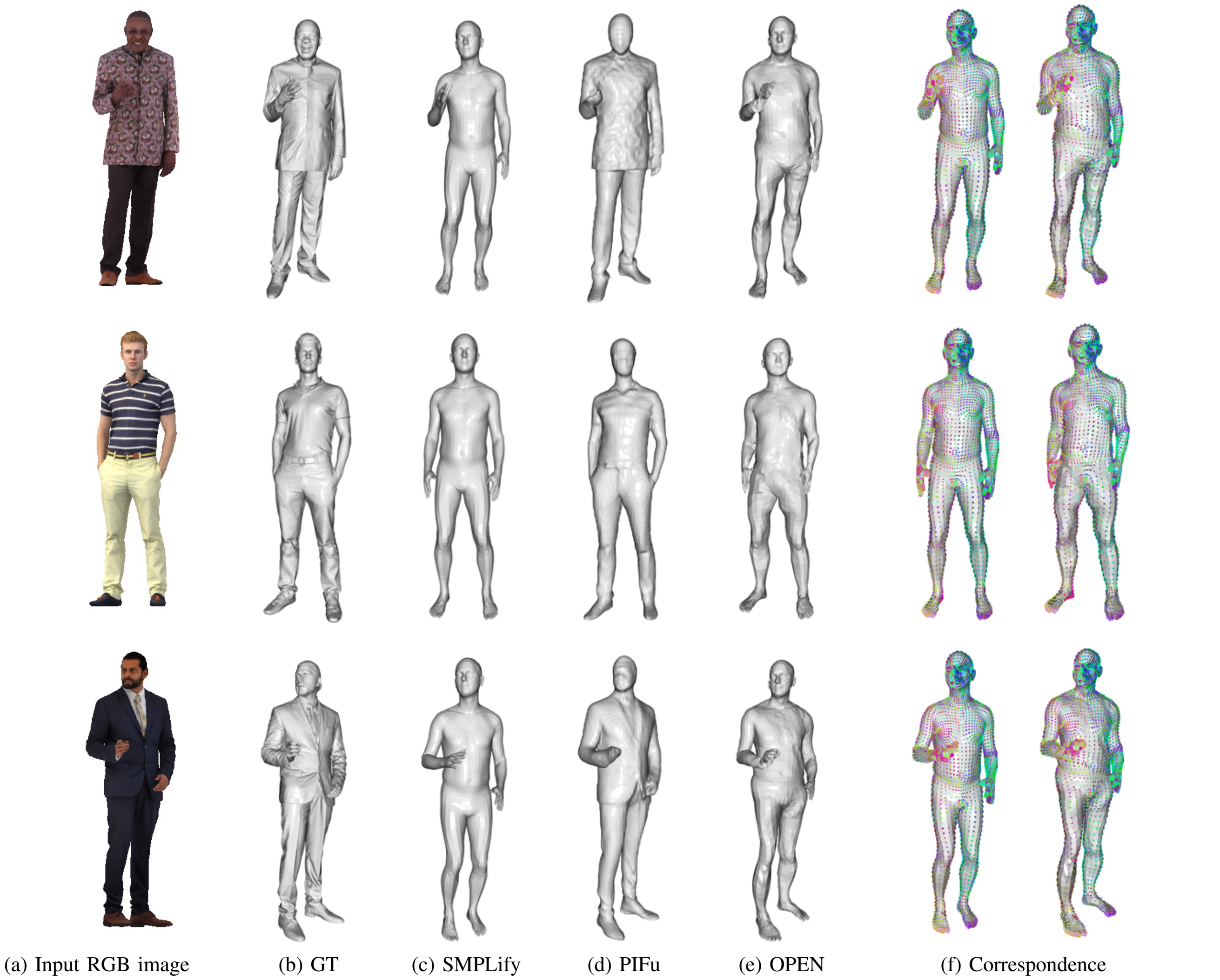

Fig. 4. Visualizations for reconstructed human body shape for the RGB images in (a). Meshes in (b) are the groundtruth mesh of the given image, and results in (c), (d), and (e) are the reconstructed mesh from SMPLify, PIFu, and OPEN respectively. The two meshes in (f) are the dense correspondences comparing the output and input point cloud of the OPEN. Points of the same color are at the same position of the SMPL vertices list.

tails, like fingers, and keep the meaningful correspondence, but the overall shape is less accurate than PIFu. For example, SMPLify fails to fit the shape of the waist most of the time. After being refined by OPEN, the mesh is more accurate and the overall outer shape is as good as PIFu, while OPEN can reconstruct the detailed structures, such as mouth, eyes, and fingers. In addition to refining the appearance of the body shape, OPEN can also correct the pose errors of the input shape. For example, for the results in the third row, the angle between two feet refined by OPEN is larger than the original input SMPL shape, which is a more accurate body shape description than the groundtruth mesh. For the dense correspondence between different shapes, by comparing the results between the input and output order of the point clouds in Figure 4 (e), we note that the order of the points is the same between the input and output, indicating that the dense correspondence is well preserved during processing.

V. Conclusion

In this paper, we propose a method named OPEN for improving the 3-D reconstructed human body shapes without disturbing the current dense correspondence between the meshes. We propose using the vertices extracted from the SMPL representation as the ordered point clouds, and generating the adjustment for each corresponding point with the guidance of the RGB image. In addition to the network, we also introduce using the external RGB images without 3-D supervision for shape reconstruction. Instead of getting expensive 3-D point cloud labels, we can use 2-D images for weak supervision with few 3-D labels and improve the network's performance. We test our methods on the public data, Renderpeople [7], and show improvement for all different settings.

REFERENCES

[1] 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 *ECCV*, 2016, pp. 561–578.

[2] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *CVPR*, 2019, pp. 5939–5948.

[3] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *ICCV*, 2019, pp. 2304–2314.

[4] S. Saito, T. Simon, J. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," in *CVPR*, 2020, pp. 84–93.

[5] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *TOG*, vol. 34, no. 6, pp. 1–16, 2015.

[6] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM Siggraph Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[7] "Renderpeople," https://renderpeople.com/3d-people, 2018.

[8] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *CVPR*, 2017, pp. 605–613.

[9] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *ICML*, 2018, pp. 40–49.

[10] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *ECCV*, 2018, pp. 52–67.

[11] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *ECCV)*, 2018, pp. 371–386.

[12] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3d faces using convolutional mesh autoencoders," in *ECCV*, 2018, pp. 704–720.

[13] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," *arXiv preprint arXiv:1608.04236*, 2016.

[14] M. Gadelha, S. Maji, and R. Wang, "3d shape induction from 2d views of multiple objects," in *3DV*, 2017, pp. 402–411.

[15] Y. Liao, S. Donne, and A. Geiger, "Deep marching cubes: Learning explicit surface representations," in *CVPR*, 2018, pp. 2916–2925.

[16] D. Jimenez Rezende, S. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3d structure from images," *NIPS*, vol. 29, pp. 4996–5004, 2016.

[17] D. Stutz and A. Geiger, "Learning 3d shape completion from laser scan data with weak supervision," in *CVPR*, 2018, pp. 1955–1964.

[18] A. O. Ulusoy, A. Geiger, and M. J. Black, "Towards probabilistic volumetric reconstruction using ray potentials," in *3DV*, 2015, pp. 10–18.

[19] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *NIPS*, 2016, pp. 82–90.

[20] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *CVPR*, 2019, pp. 4460–4470.

[21] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *CVPR*, 2019, pp. 165–174.

[22] R. Chabra, J. E. Lenssen, E. Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. Newcombe, "Deep local shapes: Learning local sdf priors for detailed 3d reconstruction," in *ECCV*, 2020, pp. 608–625.

[23] Y. Duan, H. Zhu, H. Wang, L. Yi, R. Nevatia, and L. J. Guibas, "Curriculum deepsdf," in *ECCV*, 2020, pp. 51–67.

[24] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *ECCV*, 2020, pp. 523–540.

[25] A. A. Osman, T. Bolkart, and M. J. Black, "Star: Sparse trained articulated human body regressor," in *ECCV*, 2020, pp. 598–613.

[26] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *TOG*, vol. 36, no. 6, pp. 1–17, 2017.

[27] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," in *ICCV*, 2019, pp. 2252–2261.

[28] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *CVPR*, 2018, pp. 206–215.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[30] A. Hajdu, L. Hajdu, and R. Tijdeman, "Approximations of the euclidean distance by chamfer distances," *arXiv preprint arXiv:1201.0876*, 2012.

[31] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2020.