Monocular Human Shape and Pose with Dense Mesh-borne Local Image Features

Shubhendu Jena, Franck Multon, Adnane Boukhayma

Monocular Human Shape and Pose with Dense Mesh-borne Local Image Features

Shubhendu Jena, Franck Multon, Adnane Boukhayma
Inria, Univ. Rennes, CNRS, IRISA, M2S, France

***Abstract*— We propose to improve on graph convolution based approaches for human shape and pose estimation from monocular input, using pixel-aligned local image features. Given a single input color image, existing graph convolutional network (GCN) based techniques for human shape and pose estimation (e.g. [19]) use a single convolutional neural network (CNN) generated global image feature appended to all mesh vertices equally to initialize the GCN stage, which transforms a template T-posed mesh into the target pose. In contrast, we propose for the first time the idea of using local image features per vertex. These features are sampled from the CNN image feature maps by utilizing pixel-to-mesh correspondences generated with DensePose [11]. Our quantitative and qualitative results on standard benchmarks show that using local features improves on global ones and leads to competitive performances with respect to the state-of-the-art.**

I. INTRODUCTION

Reconstructing human bodies e.g. [25], [22] and their parts (faces e.g. [7], [1], hands e.g. [4], [9]) from minimal, partial and noisy inputs is one of the most sought-after goals of human centered machine learning. In this regard, human shape and pose recovery from a single color image remains a popular problem in computer vision and graphics spurring a vast research literature, with applications in various areas such as action recognition, avatarization, human machine interaction, etc. While earlier approaches to 3D reconstruction relied on multi-view triangulation or depth information, the recent surge of deep learning has allowed the reduction of acquisition constraints to as little as a single input RGB image. The ill-posedness of this monocular setting is alleviated through learning strong statistical priors from large training datatsets with deep CNNs, which has shown to be successful especially for single shape class settings such as humans. Such approaches also benefit from transfer learning techniques by leveraging networks pre-trained on massive general datasets (e.g. ImageNet [6]). While several work [15], [18], [4] show that the learning can be further regularized by integrating differentiable parametric naked human body models (e.g. SMPL [24], GHUM [36], Frank [14]) within deep networks, other methods advocate predicting model-free 3D shape outputs (e.g. [25], [19]), with the prospect of more expressive results, whilst the parametric model is usually involved in generating training 3D pseudo-ground-truth in this case.

In this work, given a single color image, we tackle the problem of estimating human 3D shape and pose in the form of a fixed-topology triangle mesh, using a feed forward deep neural network. We specifically focus on improving on a class of model-free methods that propose to use GCNs for this task [19], [9], [20], [21], the graph's vertices and edges being defined as those of the 3D mesh representing the human shape. Traditionally, these methods extract a global latent feature vector from the image using a CNN, and this same vector is used as input feature to all the mesh vertices equally, as in [19]. The GCN then starts from these mesh-borne features and deforms a T-posed template mesh towards the target posed mesh. A noteworthy variant of this strategy consists in predicting a global feature and mapping it subsequently to initial low resolution mesh vertex features [9], [20], [21], and it was mostly explored for 3D hand prediction rather than full body partly due to the smaller mesh size.

In contrast to these methods, we propose here to use per vertex pixel-aligned local image features as initialization for the GCN stage, as illustrated in Fig.1. We use a method for predicting dense pixel-to-surface correspondences (DensePose [11]) of humans to map each visible vertex in the template geometry to a pixel in the input image. Bi-linear interpolation is then use to build a different feature vector per vertex, by sampling and stacking local image features at the vertex's corresponding 2D location in the image space, at different CNN feature depths. Given a T-posed initial mesh appended with vertex specific local image features, a GCN regresses the final mesh vertex positions. The network composed of the image CNN and mesh GCN is trained end-to-end using 3D supervision following the training scheme in [25].

We evaluate our method on standard benchmarks for human mesh prediction from images, namely 3DPW [35] and Human3.6M [13]. Our numerical and visual results demonstrate that using local features improves on using only global ones in GCN based human mesh recovery from single image. We also show that our method yields competitive results in comparison to the state-of-the-art methods.

II. RELATED WORK

There is a substantial body of work on the subject of human shape and pose estimation from a single image. We review in this section works that we deemed most relevant to the context of our contribution.

A. Optimization-based methods

Most current optimization-based methods rely on using 2D landmarks such as key-points and silhouettes [30], [3], [10] and optimize parametric models such as SCAPE [2] and

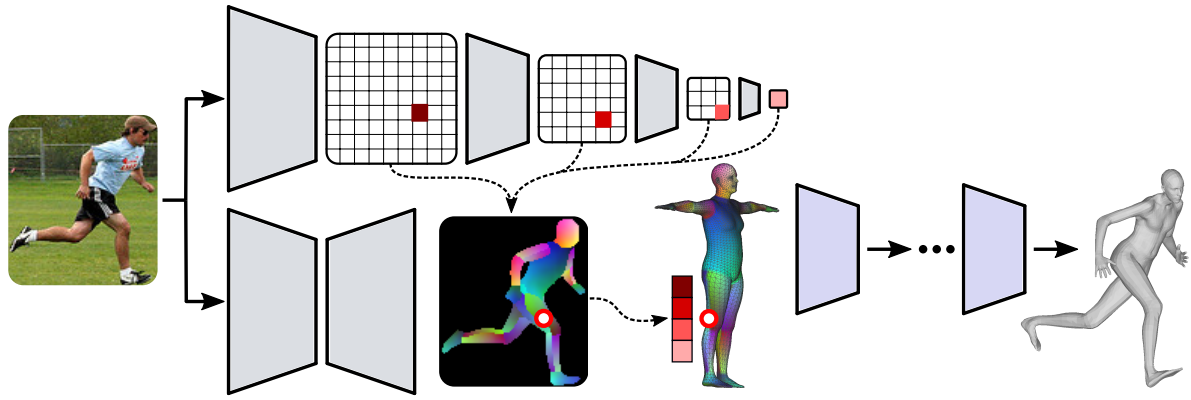

Fig. 1. Overview. Given an input color image, DensePose [11] *(Bottom left)* produces dense pixel-to-surface correspondences. Meanwhile, an image convolutional neural network (CNN) *(Top)* builds feature maps at multiple depths *(Shades of red)*. The correspondences are then used to sample *(Dashed lines)* local image features from the CNN feature maps for each template surface vertex at its corresponding image location *(Red Circle)*. Next, we use a graph convolutional network (GCN) *(Right)* to map the template surface with vertex specific local image features to the final posed surface.

SMPL [24] to fit them to these landmarks. The optimization objective consists of mainly two kinds of loss terms. The first is prior terms designed to penalize unnatural human poses and shapes. The second is data terms minimizing the difference between the inferred 2D landmarks obtained by projecting the predicted mesh onto the image and the ground-truth body landmarks. Additionally, there have been other recent methods [38] which incorporated additional terms such as part segmentation, scene and temporal constraints in the optimization objective. Although optimization-based methods provide generally reliable solutions, they are notoriously slow and prone to getting stuck in local minima especially with challenging initializations. This incentivizes learning-based solutions such as ours, which offer faster inference and do not require initialization.

B. Learning-based methods

1) Model-based: Model-based methods make use of a parametric 3D human body model to perform 3D human pose and shape estimation. The learning problem is thus reduced to learning the parameters associated with the body model from images and other types of input. A notable example is the work of [15] which directly regresses the SMPL parameters from a single RGB image, by performing a weak supervision comprising of a 2D key-point reprojection error and an adversarially learnt pose prior. SPIN [18] improves on this with a self-improving framework that incorporates 3D human model parameter optimization into the network training process. To deal with occlusions and noisy situations which make image-based methods more susceptible to failure, some approaches use alternative inputs such as 2D joint heatmaps [32], silhouettes [28], [33] and semantic segmentation maps [26]. In-spite of the aforementioned advantages behind model-based methods, such strategies can conversely also be somewhat restrictive. The tight relationship between the parameters and the model's output reduces the expressiveness of the generated meshes. Hence, as a model-free method, our framework focuses instead on directly regressing the 3D human body mesh vertices corresponding to an input image following seminal work.

2) Model-free: As the name implies, model-free methods do not rely on parametric models for 3D human body reconstruction. Instead, they directly regress an explicit body shape representation from the input images. Some of the earlier work uses a volumetric reconstruction approach with a voxel output [33], [42]. The main drawback of voxel-based methods is their inability to represent detailed surfaces due to memory limitations. Other methods use different representations such as depth maps, point clouds, etc. [8]. However, these suffer mostly from lacking surface continuity and neighborhood connectivity. To deal with these problems, recent methods propose to directly regress 3D SMPL meshes representing the output human body shape. To the best of our knowledge, this line of work started with CMR [19] which used a GCN to directly regress 3D coordinates of vertices from a global image feature. Pose2Mesh [5] also did the same but from 2D joints as input instead. The work of Lin et al. [22] yields arguably state-of-the-art performances through the use of Transformers [34] but requires considerable training time and data. Furthermore, there has been another class of methods focusing on learning dense correspondence between 2D images and 3D shapes. The seminal work of DensePose [11] which provides dense mapping from images to a human body model by regressing 2D correspondence maps has led arguably to the advent of much similar work focusing on dense correspondence. [37], [29] utilized DensePose correspondence maps for 3D human model recovery. However, they only leverage them as input images. Zhang et al. [40] predict and use local and global correspondence maps as input to further CNN stages for pose and shape prediction in a similar fashion. In contrast, we propose here to use correspondence maps to provide vertices with semantically meaningful image-aligned local features. In fact, our strategy is similar to DecoMR [39], where the authors establish dense correspondences between the surface and the input image, which is subsequently used to transfer image features to the UV-map domain and thereafter perform 3D coordinate regression with a 2D CNN. Differently, we

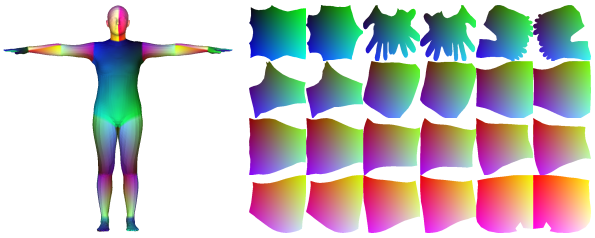

Fig. 2. Illustration of the template geometry IUV mapping using 24 body parts. The red channel is representative of the body part, while the green and blue ones span the UV coordinates within each individual part.

explore here the idea of performing vertex position regression from local image features using a convolutional network defined naturally on the same data representation as the output (i.e. GCN on the mesh as opposed to CNN on the UV-map).

III. METHOD

We describe in this section the various components of our method, which follows the illustration in Fig.1. Our input is a single RGB image and our output is a triangle human mesh with the SMPL [24] template topology. Given the input image, a convolutional neural network (CNN) is tasked with extracting 2D image features. These features are fed to a graph convolutional network (GCN) that transforms a template mesh to the final output mesh. The graph's topology is defined as per that of the mesh. Each vertex in the graph is initialized with a local feature extracted from the image encoder feature maps. This extraction is performed using a DensePose [11] correspondence map that maps pixel locations to the mesh's visible surface.

We use a ResNet50 [12] network that we denote by f to build convolutional image feature maps from an input image I, extracted in a coarse-to-fine fashion at 5 network stages:

$$\{\mathrm{F}_\mathrm{I}^l\}_{1\leq l\leq 5} = f(\mathrm{I}),$$

F_I^l being the feature map at stage l. The feature maps' spatial dimensions decrease gradually from the input image dimensions (H,W) downwards and their respective feature dimensions (spatial resolutions) are 64 (112×112), 256 (56×56), 512 (28×28), 1024 (14×14) and 2048 (1×1), summing up to $D=3904$. The final feature is a globally pooled one.

In parallel, the DensePose [11] network, denoted by h, predicts a dense mapping from the image pixels to the template mesh, in the form of a 3-channel image IUV:

$$\mathrm{IUV} = h(\mathrm{I}),$$

where $\mathrm{IUV}\in [\![0,24]\!]\times[0,1]\times[0,1]^{H\times W}$. The first channel indicates which one of 24 pre-defined body parts the pixel belongs to, 0 being the background label. The second and third channels indicate the UV-coordinate of the pixel in a pre-defined UV-map of that body part's template mesh (See Fig.2).

Given a vertex k in the template mesh, the corresponding pixel location $c_k\in [\![1,H]\!]\times[\![1,W]\!]$ in the input image is obtained using the following thresholded nearest-neighbor strategy:

$$\hat{c}_k = \underset{(i,j)}{\arg\min}\left(||\mathrm{IUV}(i,j)-\overline{\mathrm{IUV}}_k||_2\right),$$

$$c_k = \begin{cases}\hat{c}_k, & \text{if } ||\mathrm{IUV}(\hat{c}_k)-\overline{\mathrm{IUV}}_k||_2\leq\delta_k\\ \emptyset, & \text{otherwise}\end{cases}$$

where $\overline{\mathrm{IUV}}_k$ is the k^{th} vertex IUV coordinates in the pre-defined template geometry body partitioning and per part UV-map (Fig.2). Threshold δ_k is defined as the distance of the k^{th} vertex to its closest adjacent neighbor in the template UV-map space. This thresholding ensures no pixels are assigned to occluded surface regions.

We then construct the input local mesh-borne feature $\mathrm{F}_\mathrm{M}(k)$ for the k^{th} vertex as follows:

$$\mathrm{F}_\mathrm{M}(k) = \left[\mathrm{F}_\mathrm{I}^1(c_k),\ldots,\mathrm{F}_\mathrm{I}^4(c_k),\mathrm{F}_\mathrm{I}^5,c_k,\bar{x}_k,\bar{y}_k,\bar{z}_k\right]$$

where $\mathrm{F}_\mathrm{I}^l(\emptyset)=0$ for $1\leq l\leq 4$ so the non-visible vertices are assigned null local features. We note that the last feature F_I^5 is a global one and hence does not depend on the spatial 2D sampling. $(\bar{x}_k,\bar{y}_k,\bar{z}_k)$ are the k^{th} vertex coordinates in the T-pose initial mesh, hence $\mathrm{F}_\mathrm{M}(k)\in\mathbb{R}^{D+5}$.

Finally, a GCN g, whose topology is defined by the template mesh connectivity, takes as input the mesh local features $\mathrm{F}_\mathrm{M}\in\mathbb{R}^{N_v\times(D+5)}$ and predicts the final mesh vertex coordinates $\mathrm{M}\in\mathbb{R}^{N_v\times 3}$:

$$\mathrm{M} = g(\mathrm{F}_\mathrm{M}),$$

N_v being the total number of vertices ($N_v=6890$). g uses the formulation from [17] and follows the architecture described in [19], where regular graph convolutions are substituted by the semantic graph convolutions introduced in [41]. We note that body joints J can be obtained from meshes using a fixed linear regressor W: $\mathrm{J}=W\mathrm{M}$, where $\mathrm{J}\in\mathbb{R}^{N_j\times 3}$, N_j being the total number of joints ($N_j=29$).

We train the parameters of the convolutional networks f and g following the losses, training scheme, data augmentation and 3D supervision introduced in [25]. Our loss combines $L1$ reconstruction errors between the prediction and the ground-truth for mesh vertices, joints and edges as well as an additional surface normal based constraint:

$$L_\text{vertex} = \sum_{i\in\text{vertices}}||v_i-\tilde{v}_i||_1,$$

$$L_\text{joint} = \sum_{i\in\text{joints}}||j_i-\tilde{j}_i||_1,$$

$$L_\text{edge} = \sum_{(i,j)\in\text{edges}}\left|\,||v_i-v_j||_2-||\tilde{v}_i-\tilde{v}_j||_2\,\right|,$$

$$L_\text{normal} = \sum_{k\in\text{faces}}\sum_{(i,j)\in\text{k}}\left|\left\langle\frac{v_i-v_j}{||v_i-v_j||_2},\tilde{n}_k\right\rangle\right|,$$

where $\mathrm{J}=[j_1,\ldots,j_{N_j}]$ are the predicted 3D joints and $\tilde{\mathrm{J}}=[\tilde{j}_1,\ldots,\tilde{j}_{N_j}]$ the ground-truth ones. $\mathrm{M}=[v_1,\ldots,v_{N_v}]$ are the predicted mesh 3D vertices while $\tilde{\mathrm{M}}=[\tilde{v}_1,\ldots,\tilde{v}_{N_v}]$ are the

ground-truth ones. $\tilde{n}_k$ is the normal of the k^{th} face in the ground-truth mesh $\tilde{\text{M}}$. The normal and edge losses are used to ensure smoother and more visually pleasing results. The individual losses are combined with the following weighting scheme:

$$L = L_{\text{vertex}} + L_{\text{joint}} + 0.1L_{\text{normal}} + 0.1L_{\text{edge}}.$$

IV. RESULTS

We present in this section our experimental setup in addition to our results. We train our network on datasets Human3.6M [13] and MSCOCO [23] following the data augmentation and 3D supervision described in [25]. We use the Adam optimizer [16] in PyTorch on a Quadro RTX 5000 GPU for 12 epochs with a learning rate of 10^{-4}, followed by another 2 epochs using a learning rate of 10^{-5} and finally 1 extra epoch using a learning rate of 10^{-6}. The image feature extraction network f is initialized with the ImageNet [6] pre-trained weights.

A. Datasets and evaluation metrics

Human3.6M. Human3.6M [13] is a large scale indoor dataset with 3D joint coordinate annotations, and includes multiple subjects performing a variety of actions like walking, sitting and eating. Due to licensing issues, the corresponding groundtruth 3D meshes are not available. Hence, following [5], [25], we use the provided pseudo groundtruth 3D meshes obtained using SMPLify-X [27] for training. However, during inference, we use the groundtruth 3D joint coordinate annotations provided in Human3.6M [13] to keep evaluation fair. We follow the experiment setting of [5], [25] and train our models using subjects S1, S5, S6, S7 and S8. We test the models using subjects S9 and S11.

MSCOCO. MSCOCO [23] contains large-scale in-the-wild images with 2D bounding box and human joint coordinates annotations. Following [5], [25], we use the provided pseudo groundtruth 3D meshes obtained by fitting SMPLify-X [27] on the groundtruth 2D poses for training.

3DPW. 3DPW [35] is also an in-the-wild dataset consisting of 60 video sequences captured mostly in outdoor conditions. It contains 3D body pose and mesh annotations. We use this dataset only for evaluation purposes using its test set following [5], [25].

Concerning evaluation metrics, we report our performance for 3D pose estimation, in line with seminal work, using two metrics, namely mean per joint position error (MPJPE) and mean per joint position error after procrustes analysis (PA-MPJPE). MPJPE is the Euclidean distance (in mm) between the predicted and groundtruth 3D joints after root joint alignment. PA-MPJPE is the same after a further rigid alignment using Procrustes analysis.

B. Comparison with state-of-the-art methods

We evaluate our contribution numerically using the 3DPW [35] and Human3.6M [13] datasets. We report the MPJPE and PA-MPJPE metrics of our method in Table I when using only one global feature for all vertices (F_I^5) *Global (Ours)*, and also when using vertex specific local features (full F_M) *Local (Ours)*. For a fair comparison with the competition, we show other methods ([15], [19], [18], [5], [25]) trained on the same data as us and we relay their performances as they were reported in [25].

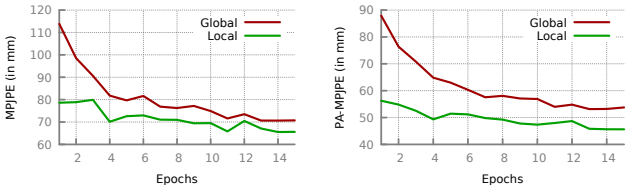

Fig. 3. MPJPE *(left)* and PA-MPJPE *(right)* testing losses on Human3.6M [13] for our baseline using only a global image feature (Global) and our proposed approach using per vertex pixel-aligned local features (Local).

Methods	Human3.6M [13]		3DPW [35]	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
HMR [15]	153.2	85.5	300.4	137.2
GraphCMR [19]	78.3	51.9	126.5	80.1
SPIN [18]	72.9	85.5	113.1	71.7
Pose2Mesh [5]	67.9	49.9	91.4	60.1
I2L-MeshNet [25]	55.7	41.7	95.4	60.8
Global (Ours)	70.74	53.76	112.51	67.98
Local (Ours)	65.61	45.62	110.31	66.52

TABLE I
COMPARISON OF MPJPE AND PA-MPJPE ON HUMAN3.6M [13] AND 3DPW [35]. ALL METHODS ARE TRAINED ON HUMAN3.6M [13] AND MSCOCO [23].

We firstly showcase the effect of using local features on the convergence of our model in Fig. 3. For both of the PA-MPJPE and MPJPE metrics, our pixel-aligned mesh features enable the model to reach significantly lower generalization errors on Human3.6M [13] at the same training epoch compared to our global feature baseline. Furthermore, we note that our *Global* baseline shares the same network design as GraphCMR [19]. However, by substituting regular convolutions with learnable adjacency matrix ones [41] and training with the same supervision and training scheme as in [25], we manage to improve its performance by roughly 8mm in MPJPE for Human3.6M [13], 14mm in MPJPE and 13mm in PA-MPJPE for 3DPW [35]. Our proposed *Local* method improves on our *Global* baseline substantially in almost all figures, by roughly 5mm in MPJPE and 8mm in PA-MPJPE for Human3.6M [13], and 2mm in MPJPE for 3DPW [35]. It is noteworthy that our *Local* version also achieves competitive results in comparison to the state-of-the-art, as it outperforms all methods presented in the table on Human3.6M [13] except for I2L-MeshNet [25], while ranking 3^{rd} on 3DPW [35] closely behind I2L-MeshNet [25] and Pose2Mesh [5], which uses 2D joints (from HRNet [31]) as input rather than a RGB image. While [5] is advantaged by the 2D pose input, we believe the performance of [25] is by virtue of their Lixel architecture which is not readily applicable to irregular graphs.

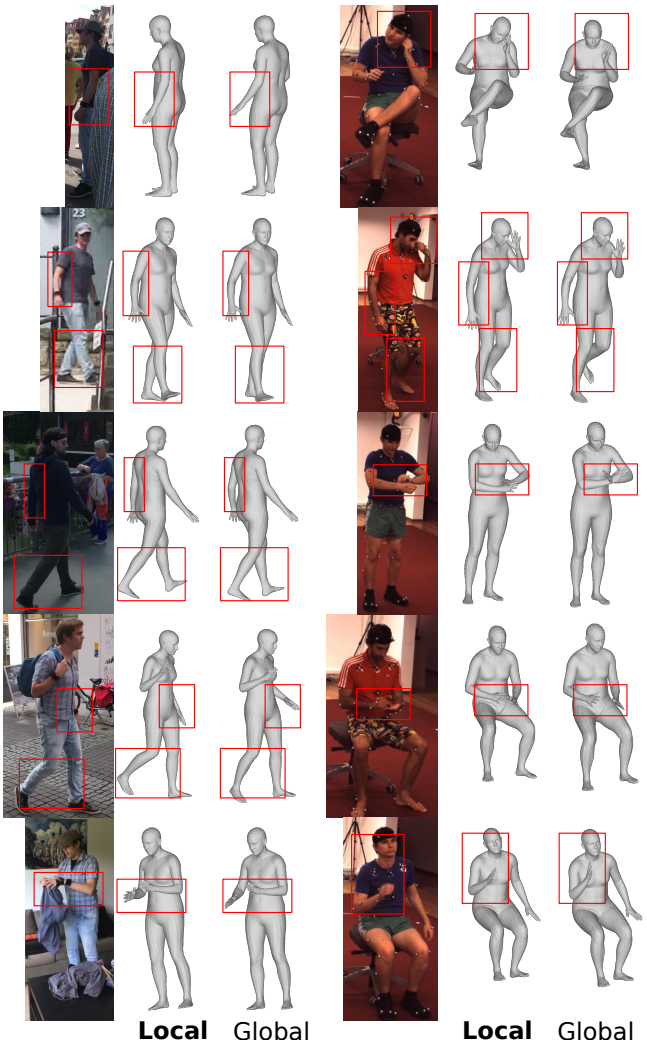

Fig. 4. Comparison of our baseline using only a global image feature (Global) and our proposed approach using per vertex pixel-aligned local features (Local) on the 3DPW [35] *(left)* and Human3.6M [13] *(right)* datasets.

The numerical superiority of our contribution compared to our baseline is also confirmed with qualitative results. As shown in Fig.4, we notice that using local image features in the GCN stage yields improved visual results, as witnessed by these examples from the 3DPW [35] and Human3.6M [13] datasets. The red boxes in the figure illustrate in particular the better positioning of the body limbs with our *Local* method compared to our *Global* baseline. We note that following [19], we show the results after an additional linear layer trained to predict SMPL instances from the previous output for smoother visual results.

V. CONCLUSION

We presented a method for 3D human shape and pose estimation from a single RGB image. The method is model-free and relies on a GCN that starts from a template T-posed mesh and regresses the final vertex coordinates. Contrarily to seminal work [19], we propose to initialize the graph convolutions with pixel-aligned vertex-specific features instead of only one global feature. These features are extracted at multiple feature map stages of an image CNN, and mapped subsequently to the graph vertices using a pixel-to-surface correspondence map [11]. Our results demonstrated the benefit of using local features in GCN based human 3D shape and pose estimation. Next, we will attempt to make the entire pipeline fully differentiable, by including the correspondence estimation network h training in the end-to-end learning framework alongside the image feature network f and graph network g.

REFERENCES

[1] V. F. Abrevaya, A. Boukhayma, P. H. Torr, and E. Boyer. Cross-modal deep face normals with deactivable skip connections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4979–4989, 2020.

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005.

[3] 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 *European conference on computer vision*, pages 561–578. Springer, 2016.

[4] A. Boukhayma, R. d. Bem, and P. H. Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[5] H. Choi, G. Moon, and K. M. Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pages 769–787. Springer, 2020.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018.

[8] V. Gabeur, J.-S. Franco, X. Martin, C. Schmid, and G. Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2232–2241, 2019.

[9] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

[10] P. Guan, A. Weiss, A. O. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1381–1388. IEEE, 2009.

[11] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] 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 transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[14] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8320–8329, 2018.

[15] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[18] 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

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2252–2261, 2019.

[19] N. Kolotouros, G. Pavlakos, and K. Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019.

[20] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4990–5000, 2020.

[21] D. Kulon, H. Wang, R. A. Güler, M. Bronstein, and S. Zafeiriou. Single image 3d hand reconstruction with mesh convolutions. *arXiv preprint arXiv:1905.01326*, 2019.

[22] K. Lin, L. Wang, and Z. Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[24] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[25] G. Moon and K. M. Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 752–768. Springer, 2020.

[26] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 international conference on 3D vision (3DV)*, pages 484–494. IEEE, 2018.

[27] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, 2019.

[28] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018.

[29] Y. Rong, Z. Liu, C. Li, K. Cao, and C. C. Loy. Delving deep into hybrid annotations for 3d human recovery in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5340–5348, 2019.

[30] L. Sigal, A. Balan, and M. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. *Advances in neural information processing systems*, 20:1337–1344, 2007.

[31] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[32] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki. Self-supervised learning of motion capture. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5236–5246. 2017.

[33] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36, 2018.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[35] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 601–617, 2018.

[36] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6184–6193, 2020.

[37] Y. Xu, S.-C. Zhu, and T. Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7760–7770, 2019.

[38] A. Zanfir, E. Marinoiu, and C. Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2148–2157, 2018.

[39] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang. 3d human mesh regression with dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7054–7063, 2020.

[40] H. Zhang, J. Cao, G. Lu, W. Ouyang, and Z. Sun. Learning 3d human shape and pose from dense body parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[41] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[42] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7739–7749, 2019.