# Convolutional Mesh Regression for Single-Image Human Shape Reconstruction

Nikos Kolotouros, Georgios Pavlakos, Kostas Daniilidis
University of Pennsylvania

## Abstract

*This paper addresses the problem of 3D human pose and shape estimation from a single image. Previous approaches consider a parametric model of the human body, SMPL, and attempt to regress the model parameters that give rise to a mesh consistent with image evidence. This parameter regression has been a very challenging task, with model-based approaches underperforming compared to nonparametric solutions in terms of pose estimation. In our work, we propose to relax this heavy reliance on the model's parameter space. We still retain the topology of the SMPL template mesh, but instead of predicting model parameters, we directly regress the 3D location of the mesh vertices. This is a heavy task for a typical network, but our key insight is that the regression becomes significantly easier using a Graph-CNN. This architecture allows us to explicitly encode the template mesh structure within the network and leverage the spatial locality the mesh has to offer. Image-based features are attached to the mesh vertices and the Graph-CNN is responsible to process them on the mesh structure, while the regression target for each vertex is its 3D location. Having recovered the complete 3D geometry of the mesh, if we still require a specific model parametrization, this can be reliably regressed from the vertices locations. We demonstrate the flexibility and the effectiveness of our proposed graph-based mesh regression by attaching different types of features on the mesh vertices. In all cases, we outperform the comparable baselines relying on model parameter regression, while we also achieve state-of-the-art results among model-based pose estimation approaches.* [1]

## 1. Introduction

Analyzing humans from images goes beyond estimating the 2D pose for one person [27, 47] or multiple people [2, 32], or even estimating a simplistic 3D skeleton [24, 25]. Our understanding relies heavily on being able to properly reconstruct the complete 3D pose and shape of people from monocular images. And while this problem is well addressed in settings with multiple cameras [8, 14], the excessive ambiguity, the limited training data, and the wide range of imaging conditions make this task particularly challenging in the monocular case.

Figure 1: Summary of our approach. Given an input image we directly regress a 3D shape with graph convolutions. Optionally, from the 3D shape output we can regress the parametric representation of a body model.

Traditionally, optimization-based approaches [1, 18, 49] have offered the most reliable solution for monocular pose and shape recovery. However, the slow running time, the reliance on a good initialization and the typical failures due to bad local minima have recently shifted the focus to learning-based approaches [15, 18, 28, 31, 39, 43], that regress pose and shape directly from images. The majority of these works investigate what is the most reliable modality to regress pose and shape from. Surface landmarks [18], pose keypoints and silhouettes [31], semantic part segmentation [28], or raw pixels [15] have all been considered as the network input. And while the input representation topic has received much debate, all the above approaches nicely conform to the SMPL model [21] and use its parametric representation as the regression target of choice. However, taking the decision to commit to a particular parametric space can be quite constraining itself. For example, SMPL is not modeling hand pose or facial expressions [14, 36]. What is even more alarming is that the model parameter space might not be appropriate as a regression target. In the case of SMPL, the pose space is expressed in the form of 3D

[1] Project Page: seas.upenn.edu/~nkolot/projects/cmr

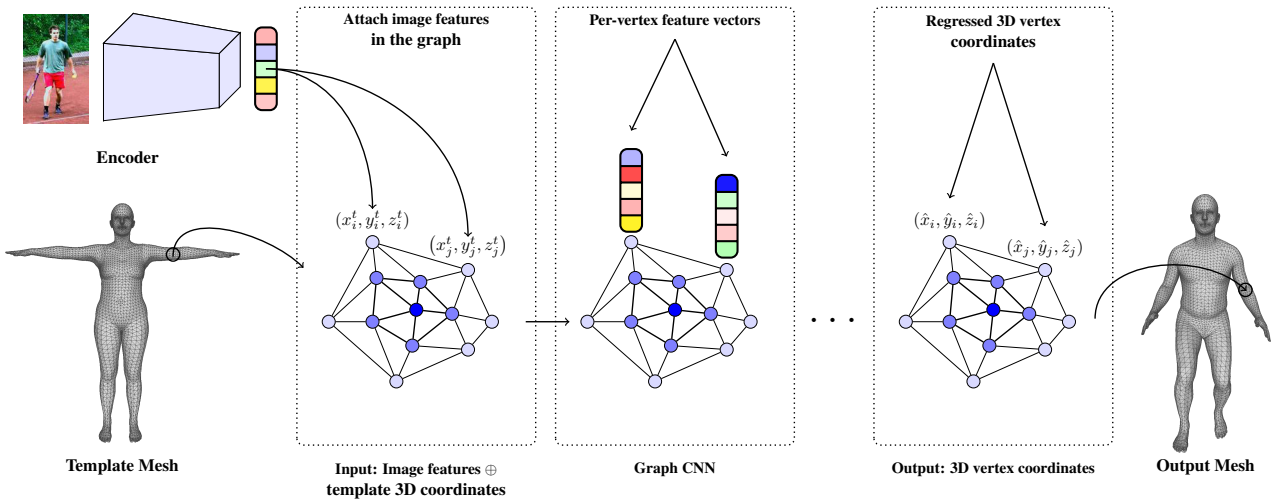


Figure 2: Overview of proposed framework. Given an input image, an image-based CNN encodes it in a low dimensional feature vector. This feature vector is embedded in the graph defined by the template human mesh by attaching it to the 3D coordinates $(x_i^t, y_i^t, z_i^t)$ of every vertex $i$. We then process it through a series of Graph Convolutional layers and regress the 3D vertex coordinates $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ of the deformed mesh.

rotations, a pretty challenging prediction target [23, 26]. Depending on the selected 3D rotation representation (e.g., axis angle, rotation matrices, quaternions), we might face problems of periodicity, non-minimal representation, or discontinuities, which complicate the prediction task. And in fact, all the above model-based approaches underperfom in pose estimation metrics compared to approaches regressing a less informative, yet more accurate, 3D skeleton through 3D joint regression [3, 24, 29, 38].

In this work, we propose to take a more hybrid route towards pose and shape regression. Even though we preserve the template mesh introduced by SMPL, we do not directly regress the SMPL model parameters. Instead, our regression target is the 3D mesh vertices. Considering the excessive number of vertices of the mesh, if addressed naively, this would be a particular heavy burden for the network. Our key insight though, is that this task can be effectively and efficiently addressed by the introduction of a Graph-CNN. This architecture enables the explicit encoding of the mesh structure in the network, and leverages the spatial locality of the graph. Given a single image (Figure 2), any typical CNN can be used for feature extraction. The extracted features are attached on the vertex coordinates of the template mesh, and the processing continues on the graph structure defined for the Graph-CNN. In the end, each vertex has as target its 3D location in the deformed mesh. This allows us to recover the complete 3D geometry of the human body without explicitly committing to a pre-specified parametric space, leaving the mesh topology as the only hand-designed choice. Conveniently, after estimating the 3D position for each vertex, if we need our prediction to conform to a specific model, we can regress its parameters quite reliably from the mesh geometry (Figure 1). This enables a more hybrid usage for our approach, making it directly comparable to model-based approaches. Furthermore, our graph-based processing is largely agnostic to the input type, allowing us to attach features extracted from RGB pixels [15], semantic part segmentation [28], or even from dense correspondences [6]. In all these cases we demonstrate that our approach outperforms the baselines that regress model parameters directly from the same type of features, while overall we achieve state-of-the-art pose estimation results among model-based baselines.

Our contributions can be summarized as follows:

- We reformulate the problem of human pose and shape estimation in the form of regressing the 3D locations of the mesh vertices, to avoid the difficulties of direct model parameter regression.
- We propose a Graph CNN for this task which encodes the mesh structure and enables the convolutional mesh regression of the 3D vertex locations.
- We demonstrate the flexibility of our framework by considering different input representations, always outperforming the baselines regressing the model parameters directly.
- We achieve state-of-the-art results among model-based pose estimation approaches.

## 2. Related work

There is rich recent literature on 3D pose estimation in the form of a simplistic body skeleton, e.g., [3, 19, 22, 24, 25, 29, 30, 34, 35, 38, 40, 41, 42, 50, 51]. However, in this Section, we focus on the more relevant works recovering the full shape and pose of the human body.

**Optimization-based shape recovery**: Going beyond a simplistic skeleton, and recovering the full pose and shape, initially, the most successful approaches followed optimization-based solutions. The work of Guan *et al*. [5] relied on annotated 2D landmarks and optimized for the parameters of the SCAPE parametric model that generated a mesh optimally matching this evidence. This procedure was made automatic with the SMPLify approach of Bogo *et al*. [1], where the 2D keypoints where localized through the help of a CNN [32]. Lassner *et al*. [18] included auxiliary landmarks on the surface of the human body, and additionally considered the estimated silhouette during the fitting process. Zanfir *et al*. [49] similarly optimized for consistency of the reprojected mesh with semantic parts of the human body, while extending the approach to work for multiple people as well. Despite the reliable results obtained, the main concern for approaches of this type is that they pose a complicated non-convex optimization problem. This means that the final solution is very sensitive to the initialization, the optimization can get stuck in local minima, and simultaneously the whole procedure can take several minutes to complete. These drawbacks have motivated the increased interest in learning-based approaches, like ours, where the pose and shape are regressed directly from images.

**Direct parametric regression**: When it comes to pose and shape regression, the vast majority of works adopt the SMPL parametric model and consider regression of pose and shape parameters. Lassner *et al*. [18] detect 91 landmarks on the body surface and use a random forest to regress the SMPL model parameters for pose and shape. Pavlakos *et al*. [31] rely on a smaller number of keypoints and body silhouettes to regress the SMPL parameters. Omran *et al*. [28] follow a similar strategy but use a part segmentation map as the intermediate representation. On the other hand, Kanazawa *et al*. [15] attempt to regress the SMPL parameters directly from images, using a weakly supervised approach relying on 2D keypoint reprojection and a pose prior learnt in an adversarial manner. Tung *et al*. [43] present a self-supervised approach for the same problem, while Tan *et al*. [39] rely on weaker supervision in the form of body silhouettes. The common theme of all these works is that they have focused on using the SMPL parameter space as a regression target. However, the 3D rotations involved as the pose parameters have created issues in the regression (e.g., discontinuities or periodicity) and typically underperform in terms of pose estimation compared to skeleton-only baselines. In this work, we propose to take an orthogonal approach to them, by regressing the 3D location of the mesh vertices by means of a Graph-CNN. Our approach is transparent to the type of the input representation we use, since the flexibility of the Graph network allows us to consider different types of input representations employed in prior work, like semantic part-based features [28], features extracted directly from raw pixels [15], or even dense correspondences [6].

**Nonparametric shape estimation**: Recently, nonparametric approaches have also been proposed for pose and shape estimation. Varol *et al*. [44] use a volumetric reconstruction approach with a voxel output. Different tasks are simultaneously considered for intermediate supervision. Jackson *et al*. [12] also propose a form of volumetric reconstruction by extending their recent face reconstruction network [11] to work for full body images. The main drawback of these approaches adopting a completely nonparametric route, is that even if they recover an accurate voxelized sculpture of the human body, there is none or very little semantic information captured. In fact, to recover the body pose, we need to explicitly perform an expensive body model fitting step using the recovered voxel map, as done in [44]. In contrast to them, we retain the SMPL mesh topology, which allows us to get dense semantic correspondences of our 3D prediction with the image, and in the end we can also easily regress the model's parameters given the vertices 3D location.

**Graph CNNs**: Wang *et al*. [46] use a Graph CNN to reconstruct meshes of objects from images by deforming an initial ellipsoid. However, mesh reconstruction of arbitrary objects is still an open problem, because shapes of objects even in the same class, e.g., chairs, do not have the same genus. Contrary to generic objects, arbitrary human shapes can be reconstructed as continuous deformations of a template model. In fact, recently there has been a lot of research in applying Graph Convolutions for human shape applications. Verma *et al*. [45] propose a new data-driven Graph Convolution operator with applications on shape analysis. Litany *et al*. [20] use a Graph VAE to learn a latent space of human shapes, that is useful for shape completion. Ranjan *et al*. [33] use a mesh autoencoder network to recover a latent representation of 3D human faces from a series of meshes. The main difference of our approach is that we do not aim to learn a generative shape model from 3D shapes, but instead perform single-image shape reconstruction; the input to our network is an image, not a 3D shape. The use of a Graph CNN alone is not new, but we consider as a contribution the insight that Graph CNNs provide a very natural structure to enable our hybrid approach. They assist us in avoiding the SMPL parameter space, which has been reported to have issues with regression [24, 31], while simultaneously allowing the explicit encoding of the graph structure in the network, so that we can leverage spatial locality and preserve the semantic correspondences.

## 3. Technical approach

In this Section we present our proposed approach for predicting 3D human shape from a single image. First, in Subsection 3.1 we briefly describe the image-based architecture that we use as a generic feature extractor. In Subsection 3.2 we focus on the core of our approach, the Graph CNN architecture that is responsible to regress the 3D vertex coordinates of the mesh that deforms to reconstruct the human body. Then, Subsection 3.3 describes a way to combine our non-parametric regression with the prediction of SMPL model parameters. Finally, Subsection 3.4 focuses on important implementation details.

### 3.1. Image-based CNN

The first part of our pipeline consists of a typical image-based CNN following the ResNet-50 architecture [7]. From the original design we ignore the final fully connected layer, keeping only the 2048-D feature vector after the average pooling layer. This CNN is used as a generic feature extractor from the input representation. To demonstrate the flexibility of our approach, we experiment with a variety of inputs, i.e., RGB images, part segmentation and DensePose input [6]. For RGB images we simply use raw pixels as input, while for the other representations, we assume that another network [6], provides us with the predicted part segmentation or DensePose. Although we present experiments with a variety of inputs, our goal is not to investigate the effect of the input representation, but rather we focus our attention on the graph-based processing that follows.

### 3.2. Graph CNN

At the heart of our approach, we propose to employ a Graph CNN to regress the 3D coordinates of the mesh vertices. For our network architecture we draw inspiration from the work of Litany *et al*. [20]. We start from a template human mesh with $N$ vertices as depicted in Figure 2. Given the 2048-D feature vector extracted by the generic image-based network, we attach these features to the 3D coordinates of each vertex in the template mesh. From a high-level perspective, the Graph CNN uses as input the 3D coordinates of each vertex along with the input features and has the goal of estimating the 3D coordinates for each vertex in the output, deformed mesh. This processing is performed by a series of Graph Convolution layers.

For the graph convolutions we use the formulation from Kipf *et al*. [17] which is defined as:

$$Y = \tilde{A}XW \tag{1}$$

where $X \in \mathbb{R}^{N\times k}$ is the input feature vector, $W \in \mathbb{R}^{k\times \ell}$ the weight matrix and and $\tilde{A} \in \mathbb{R}^{N\times N}$ is the row-normalized adjacency matrix of the graph. Essentially, this is equivalent to performing per-vertex fully connected operations followed by a neighborhood averaging operation. The neighborhood averaging is essential for producing a high quality shape because it enforces neighboring vertices to have similar features, and thus the output shape is smooth. With this design choice we observed that there is no need of a smoothness loss on the shape, as for example in [16]. We also experimented with the more powerful graph convolutions proposed in [45] but we did not observe quantitative improvement in the results, so we decided to keep our original and simpler design choice.

For the graph convolution layers, we make use of residual connections as they help in speeding up significantly the training and also lead in higher quality output shapes. Our basic building block is similar to the Bottleneck residual block [7] where $1 \times 1$ convolutions are replaced by per-vertex fully connected layers and Batch Normalization [9] is replaced by Group Normalization [48]. We noticed that Batch Normalization leads to unstable training and poor test performance, whereas with no normalization the training is very slow and the network can get stuck at local minima and collapse early during training.

Besides the 3D coordinates for each vertex, our Graph CNN also regresses the camera parameters for a weak-perspective camera model. Following Kanazawa *et al*. [15], we predict a scaling factor $s$ and a 2D translation vector $\mathbf{t}$. Since the prediction of the network is already on the camera frame, we do not need to regress an additional global camera rotation. The camera parameters are regressed from the graph embedding and not from the image features directly. This way we get a much more reliable estimate that is consistent with the output shape.

Regarding training, let $\hat{Y} \in \mathbb{R}^{N\times 3}$ be the predicted 3D shape, $Y$ the ground truth shape and $X$ the ground truth 2D keypoint locations of the joints. From our 3D shape we can also regress the location for the predicted 3D joints $\hat{J}_{3D}$ employing the same regressor that the SMPL model is using to recover joints from vertices. Given these 3D joints, we can simply project them on the image plane, $\hat{X} = s\mathbf{\Pi}(\hat{J}_{3D}) + \mathbf{t}$. Now, we train the network using two forms of supervision. First, we apply a per-vertex $L_1$ loss between the predicted and ground truth shape, i.e.,

$$\mathcal{L}_{shape} = \sum_{i=1}^{N} ||\hat{Y}_i - Y_i||_1. \tag{2}$$

Empirically we found that using $L_1$ loss leads to more stable training and better performance than $L_2$ loss. Additionally, to enforce image-model alignment, we also apply an $L_1$ loss between the projected joint locations and the ground truth keypoints, i.e.,

$$\mathcal{L}_J = \sum_{i=1}^{M} ||\hat{X}_i - X_i||_1. \tag{3}$$

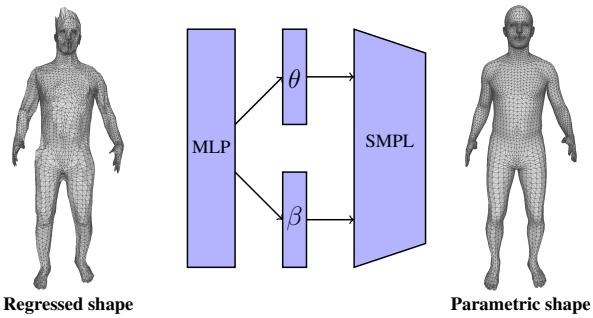


Figure 3: **Predicting SMPL parameters from regressed shape**. Given a regressed 3D shape from the network of Figure 2, we can use a Multi-Layer Perceptron (MLP) to regress the SMPL parameters and produce a shape that is consistent with the original non-parametric shape

Finally, our complete training objective is:

$$\mathcal{L} = \mathcal{L}_{shape} + \mathcal{L}_J. \tag{4}$$

This form of supervised training requires us to have access to images with full 3D ground truth shape. However, based on our empirical observation, it is not necessary for all the training examples to come with ground truth shape. In fact, following the observation of Omran *et al*. [28], we can leverage additional images that provide only 2D keypoint ground truth. In these cases, we simply ignore the first term of the previous equation and train only with the keypoint loss. We have included evaluation under this setting of weaker supervision in the Sup. Mat.

### 3.3. SMPL from regressed shape

Although we demonstrate that non-parametric regression is an easier task for the network, there are still many applications where a parametric representation of the human body can be very useful (e.g., motion prediction). In this Subsection, we present a straightforward way to combine our non-parametric prediction with a particular parametric model, i.e., SMPL. To achieve this goal, we train another network that regresses pose ($\theta$) and shape ($\beta$) parameters of the SMPL parametric model given the regressed 3D shape as input. The architecture of this network can be very simple, i.e., a Multi-Layer Perceptron (MLP) [37] for our implementation. This network is presented in Figure 3 and the loss function for training is:

$$\mathcal{L} = \mathcal{L}_{shape} + \mathcal{L}_J + \mathcal{L}_\theta + \lambda\mathcal{L}_\beta. \tag{5}$$

Here, $\mathcal{L}_{shape}$ and $\mathcal{L}_J$ are the losses on the 3D shape and 2D joint reprojection as before, while $\mathcal{L}_\theta$ and $\mathcal{L}_\beta$ are $L_2$ losses on the SMPL pose and shape parameters respectively.

As observed by previous works, e.g., [31, 24], it is challenging to regress the pose parameters $\theta$, which represent 3D rotations in the axis-angle representation. To avoid this, we followed the strategy employed by Omran *et al*. [28]. More specifically, we convert the parameters from axis-angle representation to a rotation matrix representation using the Rodrigues formula, and we set the output of our network to regress the elements of the rotation matrices. To ensure that the output is a valid rotation matrix we project it to the manifold of rotation matrices using the differentiable SVD operation. Although this representation does not explicitly improve our quantitative results, we observed faster convergence during training, so we selected it as a more practical option.

### 3.4. Implementation details

An important detail regarding our Graph CNN is that we do not operate directly on the original SMPL mesh, but we first subsample it by a factor of 4 and then upsample it again to the original scale using the technique described in [33]. This is essentially performed by precomputing downsampling and upsampling matrices $D$ and $U$ and left-multiply them with the graph every time we need to do resampling. This downsampling step helps to avoid the high redundancy in the original mesh due to the spatial locality of the vertices, and decrease memory requirements during training.

Regarding the training of the MLP, we employ a 2-step training procedure. First we train the network that regresses the non-parametric shape and then with this network fixed we train the MLP that predicts the SMPL parameters. We also experimented with training them end-to-end but we observed a decrease in the performance of the network for both the parametric and non-parametric shape.

## 4. Empirical evaluation

In this Section, we present the empirical evaluation of our approach. First, we discuss the datasets we use in our evaluation (Subsection 4.1), then we provide training details for our pipeline (Subsection 4.2), and finally, the quantitative and qualitative evaluation (Subsection 4.3) follows.

### 4.1. Datasets

We employ two datasets that provide 3D ground truth for training, Human3.6M [10] and UP-3D [18], while we evaluate our approach on Human3.6M and the LSP dataset [13].
**Human3.6M**: It is an indoor 3D pose dataset including subjects performing activities like Walking, Eating and Smoking. We use the subjects S1, S5, S6, S7 and S8 for training, and keep the subjects S9 and S11 for testing. We present results for two popular protocols (P1 and P2, as defined in [15]) and two error metrics (MPJPE and Reconstruction error, as defined in [51]).
**UP-3D**: It is a dataset created by applying SMPLify [1] on natural images of humans and selecting the successful fits. We use the training set of this dataset for training.

| Method | MPJPE | Reconst. Error |
|---|---|---|
| SMPL Parameter Regression [15] | - | 77.6 |
| Mesh Regression (FC) | 200.8 | 105.8 |
| Mesh Regression (Graph) | **102.1** | 69.0 |
| Mesh Regression (Graph + SMPL) | 113.2 | **61.3** |

Table 1: Evaluation of 3D pose estimation in Human3.6M (Protocol 2). The numbers are MPJPE and Reconstruction errors in mm. Our graph-based mesh regression (with or without SMPL parameter regression) is compared with a method that regresses SMPL parameters directly, as well as with a naive mesh regression using fully connected (FC) layers instead of a Graph-CNN.

**LSP**: It is a 2D pose dataset, including also segmentation annotations provided by Lassner *et al*. [18]. We use the test set of this dataset for evaluation.

### 4.2. Training details

For the image-based encoder, we use a ResNet50 model [7] pretrained on ImageNet [4]. All other network components (Graph CNN and MLP for SMPL parameters) are trained from scratch. For our training, we use the Adam optimizer, and a batch size of 16, with the learning rate set to $3e-4$. We did not use learning rate decay. Training with data only from Human3.6M lasts for 10 epochs, while mixed training with data from Human3.6M and UP-3D requires training for 25 epochs, because of the greater image diversity. To train the MLP that regresses SMPL parameters from our predicted shape, we use 3D shapes from Human3.6M and UP-3D. Finally, for the models using Part Segmentation or DensePose [6] predictions as input, we use the pretrained network of [6] to provide the corresponding predictions.

### 4.3. Experimental analysis

**Regression target**: For the initial ablative study, we aim to investigate the importance of our mesh regression for 3D human shape estimation. To this end, we focus on the Human3.6M dataset and we evaluate the regressed shape through 3D pose accuracy. First, we evaluate the direct regression of the 3D vertex coordinates, in comparison to generating the 3D shape implicitly through regression of the SMPL model parameters directly from images. The most relevant baseline in this category is the HMR method of [15]. In Table 1, we present the comparison of this approach (*SMPL parameter regression*) with our non-parametric shape regression (*Mesh Regression - (Graph)*). For a more fair comparison, we also include our results for the MLP that regresses SMPL parameters using our non-parametric mesh as input (*Mesh Regression - (Graph + SMPL)*). In both cases, we outperform the strong baseline of [15], which demonstrates the benefit of estimating a more flexible non-parametric regression target, instead of regressing the model parameters in one shot.

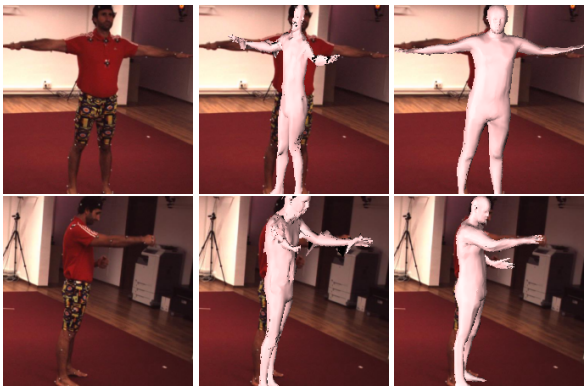

Image FC Graph CNN

Figure 4: Using a series of fully connected (FC) layers to regress the vertex 3D coordinates severely complicates the regression task and gives non-smooth meshes, since the network cannot leverage directly the topology of the graph.

| Input | Regression Type | MPJPE P1 | MPJPE P2 | Reconst. Error P1 | Reconst. Error P2 |
|---|---|---|---|---|---|
| RGB | Parameter [15] | 88.0 | - | 58.1 | 56.8 |
| | Mesh (Graph + SMPL) | **74.7** | 71.9 | **51.9** | **50.1** |
| Parts | Parameter [28] | - | - | - | 59.9 |
| | Mesh (Graph + SMPL) | 80.4 | 77.4 | 56.1 | **53.3** |
| DP[6] | Parameter [15] | 82.7 | 79.5 | 57.8 | 54.9 |
| | Mesh (Graph + SMPL) | **78.9** | **74.2** | **55.3** | **51.0** |

Table 2: Comparison of direct SMPL parameter regression versus our proposed mesh regression on Human3.6M (Protocol 1 and 2) for different input representations. The numbers are mean 3D joint errors in mm, with and without Procrustes alignment (Rec. Error and MPJPE respectively). Our results are computed after regressing SMPL parameters from our non-parametric shape. Number are taken from the respective works, except for the baseline of [15] on DensePose images, which is evaluated by us.

Beyond the regression target, one of our contributions is also the insight that the task of regressing 3D vertex coordinates can be greatly simplified when a Graph CNN is used for the prediction. To investigate this design choice, we compare it with a naive alternative that regresses vertex coordinates with a series of fully connected layers on top of our image-based encoder (*Mesh Regression - (FC)*). This design clearly underperforms compared to our Graph-based architecture, demonstrating the importance of leveraging the mesh structure through the Graph CNN during the regression. The benefit of graph-based processing is demonstrated also qualitatively in Figure 4.

**Input representation**: For the next ablative, we demonstrate the effectiveness of our mesh regression for different

| Input | Output shape | MPJPE P1 | MPJPE P2 | Reconst. Error P1 | Reconst. Error P2 |
|---|---|---|---|---|---|
| RGB | Non parametric | 75.0 | 72.7 | 51.2 | 49.3 |
| | Parametric | 74.7 | 71.9 | 51.9 | 50.1 |
| Parts | Non parametric | 78.0 | 73.4 | 54.6 | 50.6 |
| | Parametric | 80.4 | 77.4 | 56.1 | 53.3 |
| DP[6] | Non parametric | 78.0 | 72.3 | 55.3 | 50.3 |
| | Parametric | 78.9 | 74.2 | 55.3 | 51.0 |

Table 3: Comparison on Human3.6M (Protocol 1 and 2) of our non-parametric mesh with the SMPL parametric mesh regressed from our shape. Numbers are 3D joint errors in mm. The performance of the two baselines is similar.

| Method | Reconst. Error |
|---|---|
| Lassner *et al.* [18] | 93.9 |
| SMPLify [1] | 82.3 |
| Pavlakos *et al.* [31] | 75.9 |
| NBF [28] | 59.9 |
| HMR [15] | 56.8 |
| Ours | **50.1** |

Table 4: Comparison with the state-of-the-art on Human3.6M (Protocol 2). Numbers are Reconstruction errors in mm. Our approach outperforms the previous baselines.

| | FB Seg. | | Part Seg. | |
|---|---|---|---|---|
| | acc. | f1 | acc. | f1 |
| SMPLify *oracle* [1] | 92.17 | 0.88 | 88.82 | 0.67 |
| SMPLify [1] | 91.89 | 0.88 | 87.71 | 0.64 |
| SMPLify on [31] | 92.17 | 0.88 | 88.24 | 0.64 |
| Bodynet [44] | 92.75 | 0.84 | - | - |
| HMR [15] | 91.67 | 0.87 | 87.12 | 0.60 |
| Ours | 91.46 | 0.87 | 88.69 | 0.66 |

Table 5: Segmentation evaluation on the LSP test set. The numbers are accuracies and f1 scores. We include approaches that are purely regression-based (bottom) and approaches that perform some optimization (post)-processing (top). Our approach is competitive with the state-of-the-art.

types of input representations, i.e., RGB images, Part Segmentation as well as DensePose images [6]. The complete results are presented in Table 2. The RGB model is trained on Human3.6M + UP-3D whereas the two other models only on Human3.6M. For every input type, we compare with state-of-the-art methods [15, 28] and show that our method outperforms them in all setting and metrics. Interestingly, when training only with Human3.6M data, RGB input performs worse than the other representations (Table 1), because of over-fitting. However, we observed that RGB features capture richer information for in-the-wild images, thus we select it for the majority of our experiments.

Image Non-parametric Parametric

Figure 5: Examples of erroneous reconstructions. Typical failures can be attributed to challenging poses, severe self-occlusions, or interactions among multiple people.

**SMPL from regressed shape**: Additionally we examine the effect of estimating the SMPL model parameters from our predicted 3D shape. As it can be seen in Table 3, adding the SMPL prediction, using a simple MLP on top of our non-parametric shape estimate, only has a small effect in the performance (positive in some cases, negative in others). This means that our regressed 3D shape encapsulates all the important information needed for the model reconstruction, making it very simple to recover a parametric representation (if needed), from our non-parametric shape prediction.

**Comparison with the state-of-the-art**: Next, we present comparison of our approach with other state-of-the-art methods for 3D human pose and shape estimation. For Human3.6M, detailed results are presented in Table 4, where we outperform the other baselines. We clarify here that different methods use different training data (e.g., Pavlakos *et al.* [31] do not use any Human3.6M data for training, NBF *et al.* [28] uses only data from Human3.6M, while HMR [15] makes use of additional images with 2D ground truth only). However, here we collected the best results reported by each approach on this dataset.

Besides 3D pose, we also evaluate 3D shape through silhouette reprojection on the LSP test set. Our approach outperforms the regression-based approach of Kanazawa *et al.* [15], and is competitive to optimization-based baselines, e.g., [1], which tend to perform better than regression approaches (like ours) in this task, because they explicitly optimize for the image-model alignment.

**Qualitative evaluation**: Figures 5 and 6 present qual-

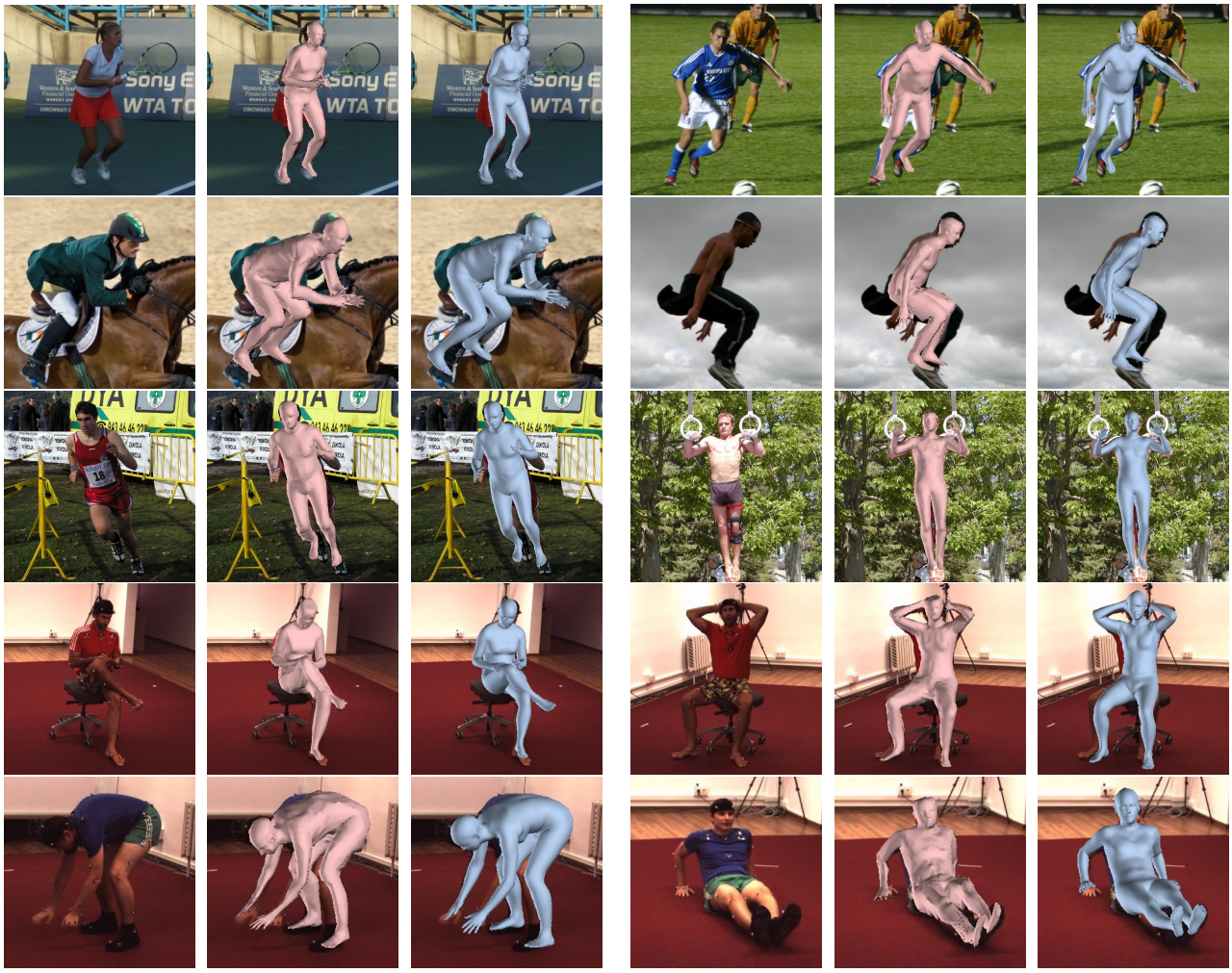


Figure 6: Successful reconstructions of our approach. Rows 1-3: LSP [13]. Rows 4-5: Human3.6M [10]. With light pink color we indicate the regressed non parametric shape and with light blue the SMPL model regressed from the previous shape.

itative examples of our approach, including both the non-parametric mesh and the corresponding SMPL mesh regressed using our shape as input. Typical failures can be attributed to challenging poses, severe self-occlusions, as well as interactions among multiple people.

**Runtime**: On a 2080 Ti GPU, network inference for a single image lasts 33ms, which is effectively real-time.

## 5. Summary

The goal of this paper was to address the problem of pose and shape estimation by attempting to relax the heavy reliance of previous works on a parametric model, typically SMPL [21]. While we retain the SMPL mesh topology, instead of directly predicting the model parameters for a given image, our target is to first estimate the locations of the 3D mesh vertices. For this to be achieved effectively, we propose a Graph-CNN architecture, which explicitly encodes the mesh structure and processes image features attached to its vertices. Our convolutional mesh regression outperforms the relevant baselines that regress model parameters directly for a variety of input representations, while ultimately, it achieves state-of-the-art results among model-based pose estimation approaches. Future work can focus on current limitations (e.g., low resolution of output mesh, missing details in the recovered shape), as well as opportunities that this non-parametric representation provides (e.g., capture aspects missing in many human body models, like hand articulation, facial expressions, clothing and hair).

**Acknowledgements:** We gratefully appreciate support through the following grants: NSF-IIP-1439681 (I/UCRC), NSF-IIS-1703319, NSF MRI 1626008, ARL RCTA W911NF-10-2-0016, ONR N00014-17-1-2093, ARL DCIST CRA W911NF-17-2-0181, the DARPA-SRC C-BRIC, and by Honda Research Institute.

## References


[1] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016. 1, 3, 5, 7

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017. 1

[3] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 668–683, 2018. 2, 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[5] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *ICCV*, 2009. 3

[6] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 2, 3, 4, 6, 7

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 6

[8] Yinghao Huang, Federica Bogo, Christoph Classner, Angjoo Kanazawa, Peter V Gehler, Ijaz Akhter, and Michael J Black. Towards accurate marker-less human shape and pose estimation over time. In *3DV*, 2017. 1

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[10] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014. 5, 8

[11] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In *ICCV*, 2017. 3

[12] Aaron S Jackson, Chris Manafas, and Georgios Tzimiropoulos. 3D human body reconstruction from a single image via volumetric regression. In *ECCVW*, 2018. 3

[13] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 5, 8

[14] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018. 1

[15] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1, 2, 3, 4, 5, 6, 7

[16] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *CVPR*, 2018. 4

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 4

[18] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *CVPR*, 2017. 1, 3, 5, 6, 7

[19] Sijin Li and Antoni B Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014. 3

[20] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *CVPR*, 2018. 3, 4

[21] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015. 1, 8

[22] Diogo C Luvizon, David Picard, and Hedi Tabia. 2D/3D pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018. 3

[23] Siddharth Mahendran, Haider Ali, and Rene Vidal. A mixed classification-regression framework for 3D pose estimation from 2D images. In *BMVC*, 2018. 2

[24] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3D human pose estimation. In *ICCV*, 2017. 1, 2, 3, 5

[25] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3D human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017. 1, 3

[26] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Košecká. 3D bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 2

[27] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 1

[28] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, 2018. 1, 2, 3, 5, 6, 7

[29] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *CVPR*, 2018. 2, 3

[30] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR*, 2017. 3

[31] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *CVPR*, 2018. 1, 3, 5, 7

[32] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016. 1, 3

[33] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3D faces using convolutional mesh autoencoders. In *ECCV*, 2018. 3, 5

[34] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3D pose estimation in the wild. In *NIPS*, 2016. 3

[35] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-classification-regression for human pose. In *CVPR*, 2017. 3

[36] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017. 1

[37] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 5

[38] Xiao Sun, Bin Xiao, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 2, 3

[39] J Tan, Ignas Budvytis, and Roberto Cipolla. Indirect deep structured learning for 3D human body shape and pose prediction. In *BMVC*, 2017. 1, 3

[40] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3D human pose with deep neural networks. In *BMVC*, 2016. 3

[41] Bugra Tekin, Pablo Marquez Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2D and 3D image cues for monocular body pose estimation. In *ICCV*, 2017. 3

[42] Denis Tome, Christopher Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. In *CVPR*, 2017. 3

[43] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NIPS*, 2017. 1, 3

[44] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3D human body shapes. In *ECCV*, 2018. 3, 7

[45] Nitika Verma, Edmond Boyer, and Jakob Verbeek. FeaStNet: Feature-steered graph convolutions for 3D shape analysis. In *CVPR*, 2018. 3, 4

[46] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single rgb images. In *ECCV*, 2018. 3

[47] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 1

[48] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 4

[49] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3D pose and shape estimation of multiple people in natural scenes–the importance of multiple scene constraints. In *CVPR*, 2018. 1, 3

[50] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017. 3

[51] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a CNN coupled with a geometric prior. *PAMI*, 41(4):901–914, 2019. 3, 5