SMPLR: Deep SMPL reverse for 3D human pose and shape recovery

Meysam Madadi¹, Hugo Bertiche² and Sergio Escalera¹,²
¹ Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra (Barcelona), Catalonia Spain
² Dept. Mathematics and Informatics, Universitat de Barcelona, Catalonia, Spain

Abstract

A recent trend in 3D human pose and shape estimation is to use deep learning and statistical morphable body models, such as the parametric Skinned Multi-Person Linear Model (SMPL). However, regardless of the advances in having both body pose and shape, SMPL-based solutions have shown difficulties on achieving accurate predictions. This is due to the unconstrained nature of SMPL, which allows unrealistic poses and shapes, hindering its direct regression or application on the training of deep models. In this paper we propose to embed SMPL within a deep model to efficiently estimate 3D pose and shape from a still RGB image. We use 3D joints as an intermediate representation to regress SMPL parameters which are again recovered as SMPL output. This module can be seen as an autoencoder where encoder is modeled by deep neural networks and decoder is modeled by SMPL. We refer to this procedure as SMPL reverse (SMPLR). Then, input 3D joints can be estimated by any convolutional neural network (CNN). To handle input noise to SMPLR, we propose a denoising autoencoder between CNN and SMPLR which is able to recover structured error. We evaluate our method on SURREAL and Human3.6M datasets showing significant improvement over SMPL-based state-of-the-art alternatives.

1. Introduction

3D human pose estimation from still RGB images remains an open challenge. 3D human pose estimation needs to deal with changes in lighting conditions, cluttered background, occlusions, inter and intra subject pose variability, as well as ill-posed depth ambiguity. Due to its nature, accurate annotation of captured data is not a trivial task and most available datasets are captured under controlled environments [5, 20, 12].

Estimation of 3D joints locations has a number of applications in computer vision, including human-robot interaction, gaming, or sports performance analysis, just to mention a few. However, 3D joints do not implicitly show the anatomy of the body shape. Being able to estimate body shape or mesh along with joints allows us to increase method applicability, such as movie editing, body soft-biometrics measurements or cloth retexturing, among others. Recently, researchers started to build deep models able to estimate body mesh [18, 6, 16] by the design of realistic morphable models like Skinned Multi-Person Linear Model (SMPL) [8]. While early SMPL-based solutions [1, 7] are proposed to use regularization to constraint gradients, recent works apply adversarial training [6], or directly estimate body volume in multi-task CNNs [26].

In this paper, we propose a deep neural network inspired in SMPL model to estimate 3D human pose and mesh given a still RGB image. SMPL is a parametric model that renders a realistic mesh based on PCA shape components along with relative angle-axis rotations of joints computed with respect to their father’s node in a defined kinematic tree. Instead of directly regressing SMPL pose and shape parameters from input image, which may generate undesirable artifacts in unseen data [6], we propose to regress these parameters from an intermediate set of representations. Unlike previous works [18, 16] which use 2D joints or part segmentation as intermediate representations, we use 3D joints and demonstrate that previous representations are suboptimal for pose and shape regression tasks. We claim such a regression does not implicitly handle input noise, causing
an offset error embedded in the model. Therefore, we also design a denoising autoencoder network [28] as an extra module able to handle structured error.

In more detail, given a CNN, we estimate a set of 3D joints and landmarks which are fed into a denoising autoencoder (DAE) network with skip connections. DAE plays as a bridge between CNN and SMPL module. In the final module, SMPL parameters are regressed. We call this procedure SMPL reverse (SMPLR). One can imagine it as an encoder-decoder module where intermediate embeddings are pose and shape components. We define encoder as a number of Multi-layer Perceptron (MLP) Networks while decoder is a bridge between CNN and SMPL module. In the final module, SMPL parameters are regressed. We call this procedure SMPL reverse (SMPLR). One can imagine it as an encoder-decoder module where intermediate embeddings are pose and shape components. We define encoder as a number of Multi-layer Perceptron (MLP) Networks while decoder is a bridge between CNN and SMPL module. In the final module, SMPL parameters are regressed. We call this procedure SMPL reverse (SMPLR).

We show the proposed model output in Fig. 1. In this section we review state-of-the-art papers for 3D human pose estimation from single RGB images.

Lifting 2D to 3D. Depth regression given 2D joints has been an active research topic in 3D human pose recovery. Nevertheless, this is an ill-posed problem where several 3D poses can be projected to the same 2D joints. Chen and Ramanan [3] show that copying depth from 3D mocap data can provide a fair estimation when a nearest 2D matching is given. However, Moreno [13] shows that distance of random pairs of poses has more ambiguity in Cartesian space than Euclidean distance matrix. Advances in recent works show that directly using simple [11] or cascade [4] MLP networks can be more accurate. Additionally, 2D joints can be noisy with wrong estimations, making previous solutions suboptimal. In this regard, Yang et al. [29] use adversarial training and benefit from available 3D data along with 2D data to infer depth information. In our case, the proposed denoising autoencoder is used to lift 2D pose to 3D in the lack of 3D ground truth data, providing accurate input to SMPLR.

Direct regression. It refers to regressing 3D pose directly from RGB image. Due to the nonlinear nature of the human pose, 3D pose regression without modeling correlation of joints is not a trivial task. Brau and Jiang [2] estimate 3D joints and camera parameters without direct supervision on them. Instead, they use several loss functions for projected 2D joints, bones size and independent Fisher priors. Sun et al. [23] propose a compositional loss function based on relative joints with respect to a defined kinematic tree. They separate 2D joints and depth estimation in the loss.

Probability maps are probability vectors assigned to each pixel/volume such that for each joint a likelihood is computed. In this way, 2D heatmaps have visual evidence in the image and different solutions are applied to infer third dimension. For instance, Tome et al. [25] combine 2D belief maps with 3D mocap data in an iterative way. They train a probabilistic PCA to learn distribution of 3D mocap data. Then at each iteration they minimize a function based on 2D belief map and learned 3D model to find the most probable 3D pose. Since probability maps are dense predictions, fully convolutional networks are usually applied by different approaches. Luo et al. [9] extend SHN to estimate 2D joint heatmaps along with limb orientation map in each stack. Pavlakos et al. [17] extend SHN to output 3D volumetric data by a coarse-to-fine architecture where at each stack the third dimension is linearly increased with 2D heatmaps. Following this trend, we extend SHN to volumetric heatmaps with a fixed depth resolution 16 and show it is working better than coarse-to-fine approaches. Recently, Nibali et al. [15] propose marginal heatmaps from different axis viewpoints. They compute 3D joints based on soft-argmax and apply multiple loss functions on the heatmaps and 3D joints.

Pose and shape estimation. Given a detailed body model, it is common to estimate body volume or mesh along with 3D pose. Early models were mainly based on the combination of simple geometric objects [22, 21], while recent models are PCA-based parametric models like SMPL [8]. Bogo et al. [1] were the first to apply SMPL for body pose and shape recovery. Their method was based on regular optimization procedures given an objective function minimizing projected joints and pre-estimated 2D joints. They applied several constraints including pose angles and shape priors. Lassner et al. [7] extended previous work by including a bi-directional distance between projected mesh and body silhouette. Later, researchers started to embed SMPL
3. Methodology

We estimate 3D human joints \( \mathbf{J} = \{j\}_1^K \) and shape \( \mathbf{T} = \{t\}_1^C \) from a single RGB image \( I \), where \( j, t \in \mathbb{R}^3 \), and \( K \) and \( C \) are the number of body joints and mesh points, respectively. In order to compute a detailed mesh, we use SMPL model [8] following recent trends in human pose and shape recovery. The goal is to estimate SMPL parameters from image \( I \) using deep learning without directly regressing them. This way, we avoid possible artifacts while keeping the architecture flexible in the lack/presence of ground truth data. Our network contains three main modules shown in Fig. 2. Next, we explain each module details.

3.1. SMPL review

SMPL is a statistical parametric function \( M(\beta, \theta; \Phi) \) which maps shape parameters \( \beta \) and axis-angle pose parameters \( \theta \) into vertices \( \mathbf{T} \), given learned model parameters \( \Phi \). Given a template average body shape \( \mathbf{T}^* \) and a dataset of scanned bodies, two sets of PCA coefficients \( \mathbf{S} = [\mathbf{S}_1, ..., \mathbf{S}_9] \in \mathbb{R}^{3C \times |\beta|} \) and \( \mathbf{P} = [\mathbf{P}_1, ..., \mathbf{P}_K] \in \mathbb{R}^{3C \times 9K} \) are learned to form model parameters \( \Phi(\beta, \theta) = 10, C = 6890 \) and \( K = 24 \) including pelvis as root). Then, Template shape \( \mathbf{T}^* \) can be updated to \( \mathbf{T}^* \) by summing to the displacements offsets computed based on parameters \( \beta \) and bases \( \mathbf{S}_i \). Bases \( \mathbf{P}_i \) are responsible for small pose-based displacements due to body soft-tissue behavior and have small contribution in the shape deformation. Given joints kinematic tree, a set of relative rotation matrices \( \mathbf{R} = [\mathbf{R}_1, ..., \mathbf{R}_K] \in \mathbb{R}^{3 \times 3} \) are computed for each joint with respect to their parents in the kinematic tree. Each \( \mathbf{R}_i \) is a function of \( \theta_i \in \mathbb{R}^3 \) and is computed based on Rodriguez formulation. These rotation matrices are used for two reasons: i) to update template shape in rest pose \( \theta^* \) by basis \( \mathbf{P}_i \), and ii) to form body pose based on standard rendering techniques in which body parts are rotated relatively in the kinematic tree. The rendering must be applied based on updated joints. Therefore, joints in template mode are updated beforehand by a regressor function \( \mathcal{J} \) (as part of parameters \( \Phi \)) and updated vertices \( \mathbf{T}^* \). SMPL model has several characteristics. First of all it is a differentiable function which makes it possible to be used along with deep networks. However, it does not implicitly constraints invalid pose and shape values and, thus, it is a many-to-one function. This means, given a RGB image, end-to-end training of SMPL model from scratch on top of a CNN may converge to a non-optimal solution. One of the reasons is the usage of Rodriguez formulation and axis-angle pose parameters, as it is known not to be unique. A possible solution is to directly use rotation matrices as proposed in [7, 16]. Secondly, SMPL is a generative model which allows us to generate millions of mocap data for free. Even it can be used to generate synthetic realistic images [27].

3.2. SMPL reverse

A natural way of embedding SMPL in a deep network is to estimate \( \beta \) and \( \theta \) given image \( I \) and feed them to SMPL. However, due to previous limitations, this is a challenging
task. Besides, direct regression of SMPL parameters may generate artifacts [6, 24]. Instead, researchers use intermediate representations like 2D joints/body silhouette [18] or body segments [16] to regress SMPL parameters. Although such representations are easier to be annotated from RGB images than ground truth SMPL data, they provide sub-optimal mapping to SMPL parameters. In this paper, we instead propose an autoencoder-like scheme, i.e. the input and output of the system are identical, while SMPL parameters are obtained in the encoder and SMPL is taken as decoder. We refer to this model as SMPL reverse (SMPLR).

We model SMPLR encoder with deep MLP networks. We design two independent networks \( R = \Omega (N; \phi_\Psi) \) and \( \beta = \Psi(B; \phi_\beta) \) with the same architecture for pose and shape estimation, respectively, where \( \phi_\) correspond to network parameters. Since we define SMPLR as an autoencoder, naturally the input to the system must be \( J \) and \( T \). However, \( T \) is a high dimensional vector and all points do not necessarily contribute equally in the computation of pose and shape parameters, wasting network capacity if all of them are considered. To cope with this issue, we select a subset of points as landmarks \( L \subset T \), which mostly represent body shape and complement \( J \). We show the 18 selected landmarks and their assigned kinematic tree in Fig. 3. Next, we explain networks details.

Given \( J \) and \( L \) concatenated as \( JL \) and kinematic tree \( \kappa \in \mathbb{R}^{2 \times 42} \), we define \( N \) to be the vector of normalized relative joints as:

\[
N_i = \frac{JL_{\kappa(i,1)} - JL_{\kappa(2,i)}}{\|JL_{\kappa(1,i)} - JL_{\kappa(2,i)}\|_2}, \text{ for } i \in [2..42],
\]

where \( \kappa(\cdot, i) \) defines fatherhood indices. The reason for this normalization is that to compute relative joint rotation \( \mathbf{R} \), we do not need to know relative distances. Besides, this frees network capacity from unnecessary data variability. Such relative distances can be embedded in the computation of shape parameters. Therefore, given relative distances \( \mathbf{B}^* \) computed from template mesh \( T^* \), we define normalized relative distances \( \mathbf{B} \) as:

\[
\mathbf{B}_i = \|JL_{\kappa(1,i)} - JL_{\kappa(2,i)}\|_2 - B^*_i, \text{ for } i \in [2..42].
\]

We also include gender as an extra term to the shape parameters \( \beta \) since SMPL originally provides two different models, male and female. By doing so we can estimate gender to select proper SMPL model and differentiate SMPL based on this parameter.

**SMPLR architecture.** We design the network as a deep MLP network with 9 sequential fully connected layers with hidden neurons 1024, 1024, 512, 512, 512, 256, 256, 256, 11, 11. We apply a residual block with 2 fully connected layers with hidden neurons 256 and 11 branching from fifth layer and concatenated to the second last layer. This residual block helps to have a more stable training and accurate result. All layers except last are followed by batch normalization and ReLU activation. We assign tanh activation to the last layer of \( \Omega \), while \( \Psi \) has no activation in the last layer. An advantage of SMPLR is that it can be trained by the aim of millions of generated mocap data. Also \( JL \), as intermediate image representation, has visual evidence from the image and can be estimated by any relevant CNN directly from the image while the whole model can be trained end-to-end. In section 3.3 we explain applied CNN for this task.

Independent training of \( \Omega \) and \( \Psi \) is critical before combination to SMPL due to aforementioned many-to-one behavior of SMPL. End-to-end training of SMPLR provides an improvement over pose and shape parameters and consequently over SMPL output. The reason is that SMPL provides an explicit correlation of pose and shape components with respect to \( J \) and \( T \). To supervise shape and pose regression networks, a typical \( \mathcal{L}_2 \) loss has been applied in recent works [24, 18]. However, we found \( \mathcal{L}_2 \) loss has problems in convergence and generalization in case of noisy inputs. Instead, we used \( \mathcal{L}_1 \) loss for both \( \Omega \) and \( \Psi \) networks and it worked well in practice. For supervision of end-to-end training, one can apply a loss on both \( J \) and \( T \) as well with respect to their ground truth values. However, we found applying a loss on the whole set of points \( T \) to be unnecessary since the points are not distributed equally on the body surface and many of them have low contribution in the computation of shape. Therefore we replace \( T \) with its predefined set \( L \) and apply an \( \mathcal{L}_1 \) loss on \( J \) and \( L \) while keeping intermediate supervision on \( \Omega \) and \( \Psi \).

**3.3. CNN backbone**

To apply SMPLR in practice we need to initialize \( JL \) to feed it to the system. \( JL \) contains visual information cues from the image, and state-of-the-art CNN are able to predict it accurately. For this task, we select SHN [14] as a standard network for human pose estimation. SHN is a stack of hourglass modules which are encoder-decoder networks with residual connections between encoder-decoder
layers. The outputs of each stack are i) a fusion of feature layers fed into the next stack and ii) a heatmap for each joint which has a supervision on it. We extend SHN to volumetric heatmaps, as a trend in recent works [26, 19], using 5 stacks. A volumetric heatmap can be defined as an extension of 2D heatmap to third dimension. First, we define a fixed size cube with 2.5 meters for each axis and transform the joints within this cube such that the root joint is located at the center of cube. Then each voxel is defined as a grid box within this reference cube and take a value from the Gaussian distribution with respect to 3D Euclidean distance to ground truth voxel and variance 1. Thus, volumetric heatmap is a tensor with shape $W \times H \times D$ with $W$, $H$ and $D$ as width, height and depth length, respectively. We define tensor shape as $64 \times 64 \times 16$. Although, this may introduce an offset error within depth, we found increasing $D$ does not show a significant improvement while increases computational complexity exponentially. Volumetric heatmaps are defined and computed for all joints and stacked in the 4th dimension ($64 \times 64 \times 16 \times 41$).

To train the network, we found cross-entropy with sigmoid activation works better than $L_2$ loss. Although the network is training as expected, the loss function does not compute any correlation among joints and network features do not learn such correlations explicitly. A solution is to use multi-task learning from different domains like body semantical segments or silhouette or 2D/3D joints [26]. However, multi-task learning needs a careful design of flow information and loss balancing among tasks. In this paper, we propose to include volumetric limb heatmaps along with joint heatmaps. By doing so we can enforce the network to learn part correlations implicitly. We compute limb heatmaps for the main 8 bones of the body: lower arm, upper arm, lower leg and upper leg for both left and right parts, along with hands, feet and shoulders. Then we combine them into 4 major parts to avoid extra computation: left arm, right arm, left leg and right leg. To compute limb heatmaps we extend joint Gaussian across bone between two joints using the orthogonal distance of each voxel to the connecting limb segment. Thus, we extend volumetric heatmaps to a tensor with shape $64 \times 64 \times 16 \times 45$. We show a sample volumetric heatmap in Fig. 4. Finally we apply soft-argmax [30, 10] to compute joint locations from heatmaps.

3.4. Denoising autoencoder

Estimated joints by SHN have structured noise. For instance, in the case of occluded joints the error is higher due to their ambiguity and the lack of visual evidence. Even the error of visible joints can be modeled by Gaussian. Such structured or Gaussian noise can be detected and learned explicitly helping to further improve initial estimation fed into SMPLR module. Denoising autoencoder (DAE) networks [28] are useful tools for such a scenario and are shown to be able to learn structured patterns of the input better than ordinal autoencoders.

In this paper, we propose a DAE network as a bridge between SHN and SMPLR. With the proposed DAE we are able to denoise 3D inputs or lift 2D estimations to 3D. This procedure can be critical for error prune CNNs, for instance fast shallow networks. Moreover, it can be detached from CNN and trained independently given millions of mocap data with adversarial noise generation mimicking structured or Gaussian noise. The architecture of DAE is shown in Fig. 2. We apply two dropouts, one right after the input and the other after last encoder layer. By doing so we mimic a two-stack DAE [28] without loss of generalization. By applying skip connections between encoder and decoder layers we force the network to learn noise structure faster and more stable way. The input to DAE is initial JL and the output is denoised JL. We apply $L_1$ loss to train the network. To force network to be more aware of adjacent joints correlations, we apply an $L_1$ loss on B (Eq. 2) as well.

4. Experiments

4.1. Training details

All models and experiments had been implemented on TensorFlow and performed on a GTX 1080 Ti. We used Adam optimizer in all trainings with a learning rate of 0.01 for SHN and 0.001 for DAE, pose and shape estimator networks. SHN converged in 150-250 epochs with batch size 6-10 samples. The rest of networks in the ablation analysis were trained with batch size 256. We have used a keeping probability 0.8 for drop-out layer in DAE.

4.2. Datasets

UP-3D [7]. This dataset was designed by fitting a gender neutral SMPL model into images from LSP, LSP-extended and MPII-HumanPose datasets, keeping samples with better estimations. This yields a total of 8515 labeled images in the wild, splitted into 5703 for training, 1423 for validation and 1389 for test. Every sample is provided with 2D annotations and SMPL parameters.

SURREAL [27]. Synthetic dataset of humans generated with SMPL model, thus with exact annotations. This is the
| Model        | Hd | Ts | Sr | Ew | Wt | Hp | Kn | Ft | Avg. Jt | Avg. Lm | Avg. Bn |
|--------------|----|----|----|----|----|----|----|----|---------|---------|---------|
| SHN          | 47.5 | 23.0 | 44.2 | 77.2 | 112.0 | 16.3 | 61.9 | 102.7 | 62.8 | - | 10.4 |
| SHN_nL       | 46.2 | 23.0 | 43.0 | 75.5 | 110.2 | 15.3 | 61.2 | 102.7 | 61.9 | 61.5 | 9.3 |
| SHN_aug      | 45.0 | 22.3 | 41.1 | 72.7 | 105.7 | 14.4 | 59.6 | 99.8 | 57.5 | 59.3 | 9.0 |
| SHN_{aug}^L | 40.8 | 20.9 | 38.0 | 66.8 | 93.4 | 14.3 | 55.7 | 92.9 | 53.0 | 54.3 | 9.7 |
| DAE          | 45.8 | 22.2 | 42.2 | 75.4 | 108.5 | 14.4 | 61.8 | 103.2 | 59.2 | 61.1 | 9.5 |
| DAE_{aug}    | 51.4 | 23.1 | 46.2 | 83.7 | 121.7 | 15.1 | 66.6 | 115.4 | 65.2 | 66.1 | 11.8 |
| SMPLR        | 16.3 | 8.9 | 13.6 | 17.7 | 19.7 | 5.5 | 11.6 | 21.3 | 14.3 | 14.3 | 6.6 |
| Ψ_{aug}      | 16.5 | 9.1 | 13.8 | 17.3 | 19.9 | 5.8 | 11.7 | 21.3 | 14.4 | 11.5 | 6.6 |
| Ω             | 55.1 | 22.3 | 48.8 | 85.8 | 127.1 | 13.4 | 68.6 | 122.3 | 67.8 | - | - |
| Ω_{aug}      | 50.1 | 20.1 | 44.9 | 83.6 | 123.6 | 12.4 | 63.9 | 111.9 | 63.8 | - | - |
| E2E          | 53.6 | 22.7 | 47.5 | 85.1 | 125.6 | 14.1 | 65.6 | 113.7 | 66.0 | 68.5 | 8.9 |
| DAE + Ψ + Ω  | 40.5 | 22.4 | 38.2 | 67.1 | 93.7 | 15.6 | 52.1 | 84.7 | 51.8 | 52.9 | 9.5 |

Table 1: Ablation study of model components. Error in mm. Hd: Head, Ts: Torso, Sr: Shoulder, Ew: Elbow, Wt: Wrist, Hp: Hip, Kn: Knee, Ft: Foot, Jt: Joints, Lm: Landmarks and Bn: Bone length, \{ \}_{nL}: input without landmarks, \{ \}_aug: training with augmentation, \{ \}_L: including limbs heatmaps in the output, \{ \}_smpl: training along with SMPL loss.

dataset used for most of our analyses. It is composed of 68036 videos containing SMPL generated body moving on top of random backgrounds, split into 55001 for training, 507 for validation and 12528 for test samples.

Human3.6M [5]. Human3.6M is the largest dataset offering high precision 3D data thanks to MoCap sensors and calibrated cameras. It is composed of RGB videos of 11 different subjects performing 15 different actions twice while recorded from 4 different viewpoints. This corresponds to around 3.6 million frames. Standard evaluation protocol is training on subjects S1, S5, S6, S7 and S8, leaving S9 and S11 for validation and finally S2, S3, S4, S10 for testing.

4.3. Evaluation protocol

We evaluate the models by mean per joint position error (MPIPE) in millimeter (mm). The same metric is extended to surface points to report error on generated body mesh. Following related works we apply two protocols: Protocol 1 where all joints/points are subtracted from the root joint and, and Protocol 2 where estimated joints are aligned with ground truth through Procrustes analysis.

4.4. Ablation study

In this section we study different components of the proposal on SURREAL validation set. For this task we sample the training dataset into around 89K frames such that every pair of samples has a minimum per joint distance of 100mm. Therefore, subsampling has a uniform distribution over the whole dataset. We use the same setup as Sec. 4.1 to train each component. We show the results and each component description in Tab. 1.

4.4.1 Stacked Hourglass

Here we analyze the different techniques implemented to enhance the performance of Stacked Hourglass. The results are shown in Tab. 1 under SHN row.

Default Volumetric SHN is a simple extension of SHN to volumetric heatmaps. We use 16 depth bins in all models. We define a few baseline networks to address experiments in the next components. SHN and SHN_{nL} are the models trained based on joints with and without landmarks (totally 18). Comparing them shows 1mm improvement in average by including landmarks. We also trained SHN with 32 depth bins. However we found no gain in the results while increasing complexity exponentially. Then we train SHN including data augmentation (SHN_{aug}). Besides regular image data augmentation like random color noise, flipping and rotation, two extra methods are applied. First, random background. By using binary masks for subjects provided at each frame, we remove the default background of the samples and replace it with a random image from VOC Pascal dataset. Similarly, we place random objects on random positions of the image to artificially create occlusions [19]. Again, the data from which objects come from is VOC Pascal dataset and, in both cases, we do not use images containing humans. Table 1 shows this augmentation is specially useful for joint prediction, while landmarks, although show improvement, do not benefit that much.

Limb Volumetric Heatmaps. We include 4 additional volumetric heatmaps in the outputs of SHN. These heatmaps correspond to limb representations, composed by creating segments from joint to joint. By fitting these heatmaps we expect to enforce the model to learn spatial relationships among joints to improve generalization. The
4.4.2 Denoising autoencoder

In this section we analyze DAE on top of baseline model $SHN$. The results are shown in Tab. 1 under DAE row. We train DAE independent to $SHN$ with three different inputs: i) uniform noise with constraints for each joint, ii) $SHN$ output (model $DAE$) and iii) $SHN$ output with depth set to 0 (model $DAE_{2d}$). As a result, uniform noise has an average error 61.7mm (not shown in the table) similar to $DAE$ (61.9mm). This is while $DAE$ improves baseline by 3mm when the input has structured error coming from $SHN$. In this way wrist takes the most benefit. For the third experiment, we want to test how DAE performs in the lack of image-level 3D ground truth data. As a result, the average error is slightly more than 65mm ($DAE_{2d}$). Although, the average error is 3mm higher than the baseline, it shows DAE can lift 2D pose to 3D with successful results given it is an ill-posed problem. We note that training $DAE_{2d}$ comparing to $DAE$ converges way slower.

4.4.3 SMPLR

In this section we analyze SMPLR on top of baseline model $SHN$. The results are shown in Tab. 1 under SMPLR row. Similar to DAE, we train shape and pose networks $\Psi$ and $\Omega$ independent to $SHN$. We report the error based on SMPL results. To analyze the effect of each network on the final results, we train $\Psi$ and $\Omega$ independently. We then feed ground-truth shape to $\Omega$ and ground-truth pose to $\Psi$.

Model $\Psi_{nL}$ is trained based on $SHN_{nL}$, i.e. the input has not landmarks. While the average joints error (14mm) is identical among $\Psi_{nL}$ and $\Psi$, the error on the landmarks (11.5 vs 14.3) shows benefit in estimating shape. We then compare pose parameters estimation. Model $\Omega_{\text{smpl}}$ is trained with $L_1$ loss on pose parameters along with SMPL loss. Comparing $\Omega$ and $\Omega_{\text{smpl}}$ shows 4mm gain by end-to-end training with SMPL. We found training $\Psi$ including SMPL loss does not change its results. In general, the higher source of errors in SMPLR is in pose parameters rather than shape. As an extra experiment, we evaluated the accuracy of gender estimation in $\Psi$ and we achieved 89.5% accuracy. This level of accuracy is critical for SMPL rendering since it has two different models for each gender.
which can have a direct impact on the mesh error.

4.5. End-to-end training

Here, we describe how the end-to-end training was performed. Thanks to joint location estimation methodology, the model is differentiable and trainable end-to-end. SMPLR model appears to be very sensitive to noise, which makes whole network training hard. For that reason, we explore the possibility of using an already trained SMPLR (model $DAE + \Psi + \Omega$) on top of $SHN_{aug}^L$ to improve the learning of the latter by back-propagating errors through SMPLR. We first show the results of SMPL output by stitching previously trained models into $DAE + \Psi + \Omega$ model. We found end-to-end training of this model on top of $SHN_{aug}^L$ does not show much improvement. Therefore, we propose the next procedure for end-to-end training. We first train $DAE$ and SMPLR on ground truth. Once trained, it is frozen and appended to the end of $SHN_{aug}^L$.

The order of magnitude of SHN loss is several times lower than SMPLR losses, therefore, without a proper balancing, the weights of the $SHN_{aug}^L$ vanish after few training steps. We empirically set this balance to be around $1e-5$ and $1e-6$. Finetuning is performed with a low learning rate to ensure the learning stability, empirically set to $1e-4$. The network shows improvement after the first training epoch, and after an additional epoch it fully converges. To ensure that this improvement is the result of appending $DAE$ and SMPLR, we trained $SHN_{aug}^L$ alone for more than 10 epochs without showing any improvement before end-to-end training. Looking at table 1 under E2E row (model $ALL$), one can see this model slightly improves performance, but on a higher degree on the legs. Note the reported error is based on $SHN_{aug}^L$ after end-to-end training.

4.6. State-of-the-art comparison

**Human3.6M.** We compare our results to state-of-the-art on Human3.6M in Tab. 2, split in two sets: SMPL-based solutions vs. only 3D pose recovery. In the former, we outperform state-of-the-art for both evaluation protocols, specially in protocol 1 improving [6] over 16mm. We note that we use $\Psi$ network trained on SURREAL dataset to estimate shape parameters, since Human3.6M contains just 5 subjects in the training set. The results in the second set shows our simple modifications to SHN achieves competitive near state-of-the-art results. Comparing to [17], a fixed small depth resolution of 16 bins in the volumetric heatmap works better than a coarse-to-fine setup. We show some qualitative results in Fig. 5.

**SURREAL.** The only competitor on this dataset is [26]. We show similar performance to this work for SMPL surface and 3D joints error using protocol 1. While SMPL output has 10mm error worse than SHN due to the offset error embedded in the regressor, our SHN-final slightly improves [26] results for 3D joint estimation. Given the ability to generate images in this dataset based on mocap data, it is likely to train SHN on this dataset to learn structured error in DAE for in-the-wild scenarios. While this is applicable in our method, [26] needs all ground truth data to train the network. We show some qualitative images in Fig. 5.

**UP-3D.** We use this dataset to show some qualitative results for in-the-wild scenario. In one setup we train all networks with UP-3D ground truth data plus a subset of SURREAL of 18K images. We show some samples from the test set in Fig. 6. The SMPL error in test set is around 90mm which is below 100.5mm error reported in [18]. In the second setup we just use 2D data to train SHN. Then, we use previously trained $DAE$ and SMPLR networks to lift 2D joints to 3D and render body mesh. The results shown in Fig. 6 shows accurate estimates in this in-the-wild scenario.

5. Conclusions

We proposed a deep based framework to recover 3D pose and shape from a still RGB image. Our model is composed of SHN backbone followed with a $DAE$ and a network capable of reversing SMPL from sparse data. Such model, capable of end-to-end training, is able to accurately reconstruct body mesh. We experimentally found that processing SHN output joints with $DAE$ is able to remove structured error. We have shown that SMPL model can be reversed and used to recover 3D pose and shape. Furthermore, the combination of both $DAE$ and SMPLR has proven to ease the learning process. Finally, we exploit SMPLR capabilities in the training of deep learning networks by backpropagating SMPL related errors through the SHN. We evaluated our proposal on SURREAL and Human3.6M datasets, outperforming SMPL-based state-of-the-art alternatives.
