Appendix: Learning Body Shape and Pose from Dense Correspondences

1 Deform-and-learn iterative training strategy

The overview of our deform-and-learn strategy is depicted in Fig. 1. It alternates deformable surface registration that fits a 3D model to 2D images and training of deep neural network that predicts 3D body shape and pose from a single image. As the first step of an iteration, we train a conditional generative adversarial networks (cGANs) that predicts 3D joint positions from 2D joint positions (Section 2), which will guide the registration process. Given image-surface dense correspondences, the registration step fits a template model to images (Section 3). After registration, we obtain a collection of body parameters θ_{fit} which is then used as supervisional signals θ_{anno} in order to train deep ConvNets that predicts body parameters θ_{conv} (Section 4). Also, the joint positions obtained using registration are used for supervising cGANs. The body parameter estimation results are used as initial solutions of surface registration in the next round. This training process is iterated for several times to get better results. Note that in the very beginning the initial pose of registration is in the T-pose, θ_0.

2 cGANs with geometric constraints for 3D human pose

We propose to use cGANs to predict depths of joints from 2D keypoints in an unsupervised manner. The results of the generator is used as soft constraints to guide image-surface registration in the next section.

We take a similar approach as Kudo et al. [6] and Chen et al. [3] where the 3D joint positions produced by a generator network (G) is projected to the image plane to obtain 2D joint positions and a discriminator (D) judges real or fake in 2D image space. The key difference of our model from previous approaches is that it incorporates joint position supervisions produced by registration to gradually improve its performance. It also incorporates geometric constraints, such as bone symmetry constraints, to further constrain the space of solution. The network architecture is depicted in Fig. 2. The input to the generator is the 2D key points of N joints and the output is depths of those joints. The predicted depths values z_i are then concatenated with x_i and y_i coordinates, rotated around the vertical axis and projected to the image space. The discriminator inputs the projected joint positions as $fake$ and the 2D keypoint data as $real$. For both networks, we use multi-layer perceptron (MLP) with eight linear layers to map 2D coordinates to depths and binary class.

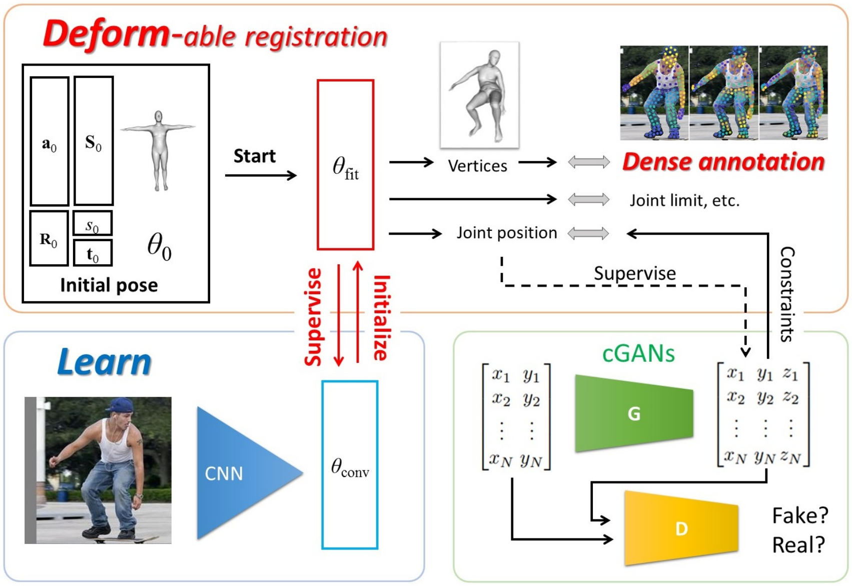

Figure 1: Overview of our deform-and-learn training strategy that iteratively performs deformable registration and deep learning. Let θ be the parameters of the body model, such as body shape and pose. In the very beginning, the initial pose of registration is in the T-pose, θ_0. Given dense image-to-surface correspondences, the first registration step fits a template model to images. After registration, we obtain a collection of θ_{fit} which is then used as supervisional signals θ_{anno} to train deep ConvNets that predicts body parameters θ_{conv}. The results of ConvNets are used as initial poses of deformable registration in the next round. These two steps are iterated to get better results.

Let $\mathbf{u}$ be the 2D joint positions of a skeleton. Also let us denote an angle around the vertical axis as ϕ. Our 3D human pose cGANs uses the following standard adversarial loss functions for G and D:

$$\mathcal{L}^{G}_{\text{adv}} = E_{\mathbf{u},\phi}[\log(1 - D(f(\mathbf{u}, G(\mathbf{u}); \phi))) \tag{1}$$

$$\mathcal{L}^{D}_{\text{adv}} = E[\log D(\mathbf{u})] \tag{2}$$

where f denotes the rotation and the projection function. Note that we validate the pose from multiple views, where we empirically set angles [deg] as $\phi = \{45, 60, 90, 135, 180, 235, 270\}$ to validate each predicted pose. We could use more viewing angles but we found this sufficient.

In addition to the adversarial loss, the geometric loss is also applied. Specifically, we use the bone symmetry loss $\mathcal{L}_{\text{sym}}$ that constrain the left and right limb be similar and the bone ratio loss $\mathcal{L}_{\text{ratio}}$ that minimizes the difference between the normalized

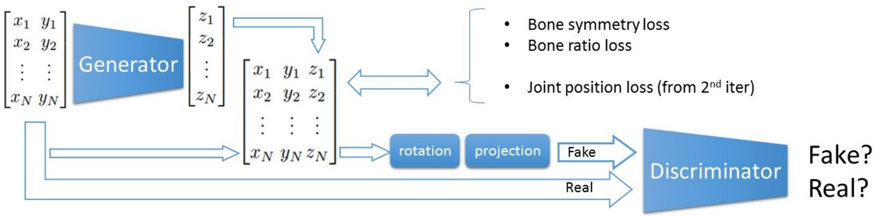

Figure 2: cGANs with geometric constraints for 3D human pose estimation. The input to the generator is the 2D key points of N joints and the output is depths of those joints. Once the generator outputs the depths values z_i, they are concatenated with x_i and y_i coordinates. This 3D joint position is rotated about the vertical axis and projected to the image space. The discriminator inputs the projected joint positions as $fake$ and the 2D keypoint data as $real$. In addition to the adversarial loss, we incorporate geometric constraints to further constrain the space of solution. Furthermore, from the 2nd iteration, we incorporate a joint position loss to improve prediction accuracy.

bone length prediction and that of dataset. The bone ratio loss $\mathcal{L}_{\text{ratio}}$ is defined as:

$$\mathcal{L}_{\text{ratio}} = \sum_{e \in \mathcal{B}} \| \frac{l_e}{l_{\text{trunk}}} - \frac{\bar{l}_e}{\bar{l}_{\text{trunk}}} \|^2 \tag{3}$$

where $\frac{l_e}{l_{\text{trunk}}}$ is the ratio of the bone length for bone e in a set of bones $\mathcal{B}$ in a skeleton with respect to the trunk length and $\frac{\bar{l}_e}{\bar{l}_{\text{trunk}}}$ is that of the average skeleton. Let $\mathcal{B}_s$ be the set of symmetry pairs of bone segments which contains indices of bones e.g., the left and right forearm. Then the bone symmetry loss $\mathcal{L}_{\text{sym}}$ is defined as:

$$\mathcal{L}_{\text{sym}} = \sum_{i,j \in \mathcal{B}_s} \|l_i - l_j\|^2 \tag{4}$$

where l_i and l_j is the lengths of the bone for symmetry bone pairs. After the first iteration, to improve estimation, we add a joint loss that penalizes the deviation of joint position predictions from that of registration results. This is enforced as the MSE loss:

$$\mathcal{L}_{\text{joint}} = \sum_{i \in \mathcal{J}} \|\mathbf{p}_i - \bar{\mathbf{p}}_i\|^2 \tag{5}$$

where $\mathbf{p}_i$ is the joint position at joint i. We mix the above losses to train the generator such that the loss is:

$$\mathcal{L}^G = \epsilon \mathcal{L}^G_{\text{adv}} + \mathcal{L}_{\text{ratio}} + \mathcal{L}_{\text{sym}} + \mu \mathcal{L}_{\text{joint}} \tag{6}$$

where ϵ is the weight for controlling the strength of the adversarial term, which we set to 0.1 in this paper. mu is the weight for $\mathcal{L}_{\text{joint}}$ which is decreased from 100 to 1.

3 Image-surface deformable registration

We propose a deformable surface registration technique to fit a template mesh model to images to obtain 3D body shape and pose annotations for training deep ConvNets. Here deformable registration is formulated as a gradient-based method based on back propagations, which can be implemented with a deep learning framework and parallelized with GPU. With the automatic differentiation mechanisms provided with a deep learning framework, adding and minimizing various kinds of losses have made easy and straightforward. As a result, the proposed deformable registration technique thus incorporates kinematic, geometric and correspondence losses.

Given image-surface dense correspondences annotated on images, the template mesh is fitted to images by optimizing body parameters $\theta = [\mathbf{a}, \mathbf{S}, \mathbf{R}, s, \mathbf{t}]$ subject to kinematic and geometric constraints. In total, the overall loss function for our registration is of the form:

$$\begin{aligned} \mathcal{L}_{\text{regist}} = {} & \omega_{\text{dense}}\mathcal{L}_{\text{dense}} + \omega_{\text{KP}}\mathcal{L}_{\text{KP}} \\ & + \omega_{\text{scale}}\mathcal{L}_{\text{scale}} + \omega_{\text{joint}}\mathcal{L}_{\text{joint}} + \omega_{\text{det}}\mathcal{L}_{\text{det}} \end{aligned} \tag{7}$$

where $\mathcal{L}_{\text{dense}}$ and $\mathcal{L}_{\text{KP}}$ are the dense correspondence and key point losses that penalize the alignment inconsistency of the body model and images defined in terms of dense correspondences and key points. The losses $\mathcal{L}_{\text{scale}}$ and $\mathcal{L}_{\text{joint}}$ is the segment scaling smoothness and kinematic loss for regularization. The transformation determinant loss $\mathcal{L}_{\text{det}}$ makes the determinant of the global transformation positive. In addition, ω_{dense}, ω_{KP}, ω_{scale}, ω_{joint} and ω_{det} are the respective weights for the above defined losses. The initialization of body parameters is provided from the predictions of deep ConvNets. For the very first iteration where the Convnet predictions are not available, segment scale $\mathbf{S}$ is set 1 for all segments and pose $\mathbf{a}$ is set to 0 for all joints, which means that registration is started from the T pose.

3.1 Correspondence fit loss

The correspondence loss comprises two losses: the dense correspondence loss $\mathcal{L}_{\text{Dense}}$ and keypoint loss $\mathcal{L}_{\text{KP}}$.

Dense correspondence loss Let us define a set of image-surface correspondences $\mathcal{C} = \{(\mathbf{p}_1, \mathbf{v}_{\text{idx}(1)}) \dots (\mathbf{p}_N, \mathbf{v}_{\text{idx}(N)})\}$, where $\mathbf{p}$ is the image points. In addition $\text{idx}(i)$ is the index of the mesh vertices that is matched with image point i. Now we can define the dense correspondence loss as:

$$\mathcal{L}_{\text{dense}} = \sum_{i \in \mathcal{C}} \|\mathbf{p}_i - \mathbf{x}_{\text{idx}(i)}\|^2 \tag{8}$$

Here the mean squared error (MSE) between image point annotations $\mathbf{p}_i$ and the corresponding points on a surface projected to the 2D image $\mathbf{x}_{\text{idx}(i)}$ are calculated.

Key point loss To produce 3D poses with statistically valid depths, the results of cGAN is used to guide deformable registration. Instead of attaching a discriminator to the registration framework, the depth values from cGAN and the ground truth 2D joint coordinates are provided as a soft constraint to constrain the position of the 3D joints based on the MSE loss:

$$\mathcal{L}_{\mathrm{KP}} = \sum_{i \in \mathcal{J}} \|x_i - \bar{x}_i\|^2 + \sum_{i \in \mathcal{J}} \|y_i - \bar{y}_i\|^2 + \sum_{i \in \mathcal{J}} \|z_i - z_i^{\mathrm{GAN}}\|^2 \tag{9}$$

where $\bar{x}_i$ and $\bar{y}_i$ are the ground truth of 2D key points. Also z_i^{GAN} is the depth at joint i predicted by cGANs.

3.2 Geometric and kinematic loss

Since we attract the template mesh to 2D image coordinates, the problem is ill-posed and deformations are not constrained. Thus we introduce the regularization terms that avoids extreme deformations.

Segment scaling smoothness To avoid extreme segment scalings, we introduce the scaling smoothness loss, which minimizes difference between scalings of adjacent segments:

$$\mathcal{L}_{\mathrm{scale}} = \sum_{e \in \mathcal{B}} \|\mathbf{S}_e - \mathbf{S}_{\mathrm{adj}(e)}\|^2 \tag{10}$$

Joint angle smoothness and limit loss To prevent extreme poses, we introduce joint angle smoothness loss and joint limit loss. The joint smoothness loss is enforced at every joint in a skeleton, $\mathcal{J}$, and will contribute to avoid extreme bending. To avoid hyper-extensions which will bend certain joints like the elbows and knees (where we represent as $\mathcal{J}'$) in the negative direction, we introduce the joint limit loss. The regularizations that act on joints are thus represented as:

$$\mathcal{L}_{\mathrm{angle}} = \sum_{i \in \mathcal{J}} \|\mathbf{a}_i\|^2 + \sum_{i \in \mathcal{J}'} \|\exp(\mathbf{a}_i)\|^2 \tag{11}$$

where the first term minimizes joint angles whereas the latter term penalizes rotations violating natural constraints by taking exponential and minimizing it.

Transformation determinant loss Since we use a rotation matrix for representing the global rotation at the root, it is necessary to apply a constraint on a matrix to keep its determinant to positive. Thus, we define the transformation determinant loss as:

$$\mathcal{L}_{\mathrm{det}} = \exp(-\det(\mathbf{R})) \tag{12}$$

4 Estimating 3D body shape and pose from a single image

4.1 Deep ConvNets for body shape and pose regression

Using the results obtained by deformable registration as annotations for training deep ConvNets, we regress body shape and pose parameters with an image. We also add the dense correspondence and keypoint losses as in Section 3.1 for additional supervisions. In total, we minimize the loss function of the form:

$$\mathcal{L}_{\text{conv}} = \alpha \mathcal{L}_{\text{regress}} + \beta \mathcal{L}_{\text{dense}} + \gamma \mathcal{L}_{\text{KP}} \tag{13}$$

where $\mathcal{L}_{\text{regress}}$ is the regression loss for body parameters. α, β and γ are the respective weights. Let θ_i be the parameters for i-th sample, the regression loss is defined as:

$$\mathcal{L}_{\text{regress}} = \sum_i \text{smooth}_{L1}(\theta_i - \bar{\theta}_i) \tag{14}$$

where $\bar{\theta}$ is the annotation provided from the registration step. Here we use the smooth L1 loss because of its robustness to outliers. This choice was more effective than the L2 loss in contributing to decreasing the error during the iterative training strategy in the presence of potential outliers and noisy annotations.

The body model is similar to the one we used for registration, except for the pose representation, where we found that the use of quaternions improved stability and convergence of training than axis angle, which is probably due to the fact that the values of quaternions are in between -1 and 1 and is easier for ConvNets to learn with than axis angles. Other parameters are same as the ones used in Section 3, which results in 132 parameters in total. Note that the global rotation is regressed using 9 parameters and the Gram Schmidt orthogonalization is used to make a transformation into a rotation. We use ResNet50 [4] pretrained on the ImageNet dataset as the base network.

5 Experimental results

5.1 Implementation and training detail

Our method is implemented using Pytorch. We use the Adam optimizer for all the steps in our approach. Training takes 2-3 days using a NVIDIA Quadro P6000 graphics card with 24 GB memory. At each iteration, the multi-view cGANs is trained for 50 epochs with the batch size of 1024 and the learning rate of 0.0002. The body regressor is trained for 30 epochs with the batch size of 30 and the learning late of 0.0001. We set the parameters in the loss function to $\alpha = \gamma = 1$ and $\beta = 10$. For deformable surface registration, we use the learning rate of 0.1 and batch size of 10. We empirically set the parameters to $\omega_{\text{dense}} = 1000$, $\omega_{\text{KP}} = 1$, $\omega_{\text{scale}} = 10$, $\omega_{\text{joint}} = 0.001$ and $\omega_{\text{det}} = 1$. For the first training iteration, in order to recover a global rotation, we set $\omega_{\text{scale}} = 100$ and $\omega_{\text{joint}} = 1$ to make the body model stiff, which is a common strategy in deformable registration [1]. We perform 300 forward-backward passes during the registration step at the 1st iteration. From the second iteration, 100 forward-backward passes were sufficient, since we start from the ConvNet prediction.

5.2 Dataset

DensePose DensePose dataset [10] contains images with dense annotations of part-specific UV coordinates (Fig. 3), which are provided on the MS COCO images. To

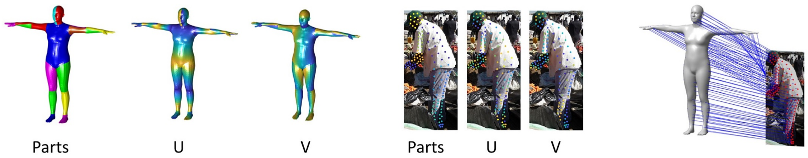

Figure 3: Dense image-surface correspondences between the template body surface and image points are found from DensePose annotations [10] by searching nearest points in the UV space of each body part.

obtain part-specific UV coordinates, body surfaces of a SMPL human body model are partitioned into 24 regions and each of them are unwrapped so that vertices have UV coordinates. Thus, every vertex on the model have unique parameterizations. Images are manually annotated by human annotators with part indices and UV coordinates to establish dense image-to surface correspondences.

To use this dense correspondences in 3D model fitting, we find the closest points from image pixels to surface vertices in UV coordinates of every part. The nearest neighbor search is done in this direction because image pixels are usually coarser than surface vertices. We were able to obtain approximately $40k$ annotated training images.

DensePoseTrack We also use 7k images from the DensePoseTrack dataset [8], where labeling is done by a semi automated annotation method of dense correspondences using motion cues to propagate annotations through time.

Human3.6M Human 3.6M dataset is a large scale dataset [5] for 3D human pose detection. This dataset contains 3.6 million images of 15 everyday activities, such as walking, sitting and making a phone call, which is performed by 7 professional actors and is taken from four different views. 3D positions of joint locations captured by MoCap systems are also available in the dataset. In addition, 2D projections of those 3D joint locations into images are available. To obtain dense annotations for this dataset, we use Mosh [7] to obtain SMPL body and pose parameters from the raw 3D Mocap markers and then projected mesh vertices onto images to get dense correspondences between images and a template mesh. We collected $65k$ images with dense correspondence annotations.

MPII 2D human pose 2D keypoint labels in this dataset were used to train the cGANs. The images from MPII 2D human pose dataset [2] is used for testing and was not used in training.

5.3 Protocol and metric

We followed the same evaluation protocol used in previous approaches [9, 11] for evaluation on Human3.6M dataset, where we use 5 subjects (S1, S5, S6, S7, S8) for training and the rest 2 subjects (S9, S11) for testing. The error metric for evaluating 3D joint positions is called mean per joint position error (MPJPE) in mm. Following [11] the output joint positions from ConvNets is scaled so that the sum of all 3D bone lengths is equal to that of a canonical average skeleton.

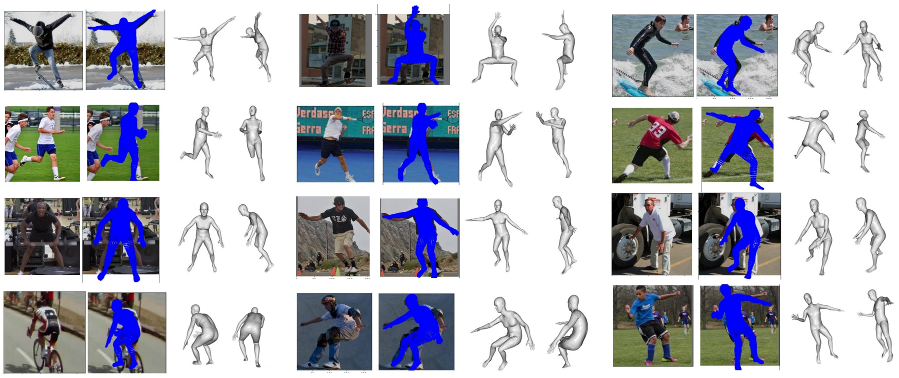

Figure 4: Qualitative result. From left to right: original image, overlay, 3D reconstruction results viewing from the front and side. Our technique is able to recover body shape and pose from in-the wild images. Note that the viewing distance of the 3D reconstruction does not exactly match with that of an input image.

We also evaluate the fit of the body model to images based on the mean per pixel error and mean per vertex error which measures distances from the ground truth to the predicted vertices in 2D image space and 3D space. Prior to calculating the per-vertex error, we obtain a similarity transformation by Procrustes analysis and align the predicted vertices to the ground truth.

5.4 Qualitative results

In Figs. 4 we show our results on body shape and pose estimation.

References

[1] B. Amberg, S. Romdhani, and T. Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *CVPR*, 2007.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and S. Bernt. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, June 2014.

[3] C. Chen, A. Tyagi, A. Agrawal, D. Drover, M. V. Rohith, S. Stojanov, and J. M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. *CoRR*, abs/1904.04812, 2019.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] 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, jul 2014.

[6] Y. Kudo, K. Ogaki, Y. Matsui, and Y. Odagiri. Unsupervised adversarial learning of 3d human pose from 2d joint locations, 2018.

[7] M. M. Loper, N. Mahmood, and M. J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 33(6):220:1–220:13, Nov. 2014.

[8] N. Neverova, J. Thewlis, R. A. Güler, I. Kokkinos, and A. Vedaldi. Slim densepose: Thrifty learning from sparse annotations and motion cues. abs/1906.05706, 2019.

[9] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. *CoRR*, abs/1611.07828, 2016.

[10] G. Riza, N. Natalia, and K. Iasonas. Densepose: Dense human pose estimation in the wild. *arXiv*, 2018.

[11] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Weakly-supervised transfer for 3d human pose estimation in the wild. *arXiv preprint arXiv:1704.02447*, 2017.