Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation

Mohamed Omran[1]   Christoph Lassner[2*]   Gerard Pons-Moll[1]   Peter V. Gehler[2*]

Bernt Schiele[1]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

[2]Amazon, Tübingen, Germany

{mohomran, gpons, schiele}@mpi-inf.mpg.de, {classner, pgehler}@amazon.com[*]

Abstract

Direct prediction of 3D body pose and shape remains a challenge even for highly parameterized deep learning models. Mapping from the 2D image space to the prediction space is difficult: perspective ambiguities make the loss function noisy and training data is scarce. In this paper, we propose a novel approach (Neural Body Fitting (NBF)). It integrates a statistical body model within a CNN, leveraging reliable bottom-up semantic body part segmentation and robust top-down body model constraints. NBF is fully differentiable and can be trained using 2D and 3D annotations. In detailed experiments, we analyze how the components of our model affect performance, especially the use of part segmentations as an explicit intermediate representation, and present a robust, efficiently trainable framework for 3D human pose estimation from 2D images with competitive results on standard benchmarks. Code will be made available at http://github.com/mohomran/neural_body_fitting

1. Introduction

Much research effort has been successfully directed towards predicting 3D keypoints and stick-figure representations from images of people. Here, we consider the more challenging problem of estimating the parameters of a detailed statistical human body model from a single image.

We tackle this problem by incorporating a model of the human body into a deep learning architecture, which has several advantages. First, the model incorporates limb orientations and shape, which are required for many applications such as character animation, biomechanics and virtual reality. Second, anthropomorphic constraints are automatically satisfied – for example limb proportions and symmetry. Third, the 3D model output is one step closer to a faithful 3D reconstruction of people in images.

Traditional *model-based* approaches typically optimize an objective function that measures how well the model fits the image observations – for example, 2D keypoints [6, 24]. These methods do not require paired 3D training data (images with 3D pose), but only work well when initialized close to the solution. By contrast, initialization is not required in forward prediction models, such as CNNs that directly predict 3D keypoints. However many images with 3D pose annotations are required, which are difficult to obtain, unlike images with 2D pose annotations.

Therefore, like us, a few recent works have proposed hybrid CNN architectures that are trained using model-based loss functions [56, 62, 22, 38]. Specifically, from an image, a CNN predicts the parameters of the SMPL 3D body model [28], and the model is re-projected onto the image to evaluate the loss function in 2D space. Consequently, 2D pose annotations can be used to train such architectures. While these hybrid approaches share similarities, they all differ in essential design choices, such as the amount of 3D vs 2D annotations for supervision and the input representation used to lift to 3D.

To analyze the importance of such components, we introduce Neural Body Fitting (NBF), a framework designed to provide fine-grained control over all parts of the body fitting process. NBF is a hybrid architecture that integrates a statistical body model within a CNN. From an RGB image or a semantic segmentation of the image, NBF directly predicts the parameters of the model; those parameters are passed to SMPL to produce a 3D mesh; the joints of the 3D mesh are then projected to the image closing the loop. Hence, NBF admits both full 3D supervision (in the model or 3D Euclidean space) and weak 2D supervision (if images with only 2D annotations are available). NBF combines the advantages of direct bottom-up methods and top-down methods. It requires neither initialization nor large amounts of
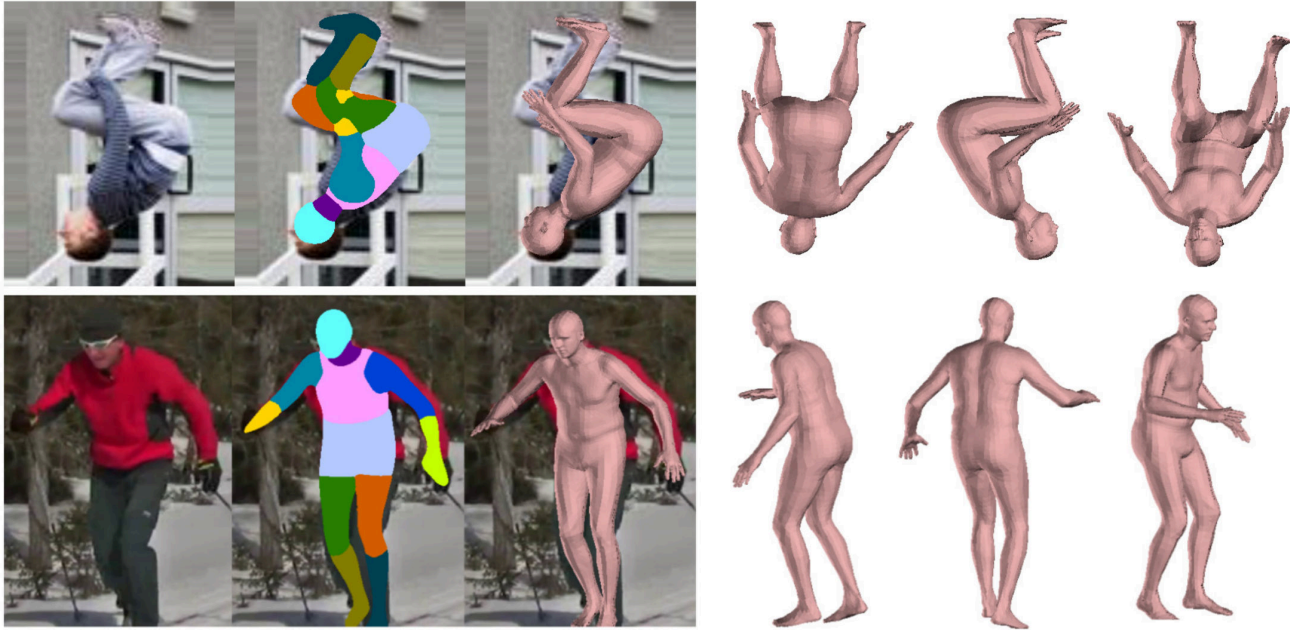

Figure 1: Given a single 2D image of a person, we predict a semantic body part segmentation. This part segmentation is represented as a color-coded map and used to predict the parameters of a 3D body model.

3D training data.

One key question we address with our study is whether to use an intermediate representation rather than directly lifting to 3D from the raw RGB image. Images of humans can vary due to factors such as illumination, clothing, and background clutter. Those effects do not necessarily correlate with pose and shape, thus we investigate whether a simplification of the RGB image into a semantic segmentation of body parts improves 3D inference. We also consider the granularity of the body part segmentation as well as segmentation quality, and find that: (i) a color-coded 12-body-part segmentation contains sufficient information for predicting shape and pose, (ii) the use of such an intermediate representation results in competitive performance and easier, more data-efficient training compared to similar methods that predict pose and shape parameters from raw RGB images, (iii) segmentation quality is a strong predictor of fit quality.

We also demonstrate that only a small fraction of the training data needs to be paired with 3D annotations. We make use of the recent UP-3D dataset [24] that contains 8000 training images in the wild along with 3D pose annotations. Larger 2D datasets exist, but UP-3D allows us to perform a controlled study.

In summary, our contribution is twofold: first we introduce NBF, which unites deep learning-based with traditional model-based methods taking advantage of both. Second, we provide an in-depth analysis of the necessary components to achieve good performance in hybrid architectures and provide insights for its real-world applicability that we believe may hold for many related methods as well.

2. Related Work

Human pose estimation is a well-researched field and we focus on 3D methods; for a recent extensive survey on the field we refer to [48].

Model-based Methods. To estimate human pose and shape from images, model-based [42] works use a parametric body model or template. Early models were based on geometric primitives [34, 11, 39, 50, 54], while more recent ones are estimated from 1000s of scans of real people, and are typically parameterized by separate body pose and shape components [5, 13, 28, 71, 41], with few exceptions, *e.g.*, [21]. Most model-based approaches fit a model to image evidence through complex non-linear optimization, requiring careful initialization to avoid poor local minima.

To reduce the complexity of the fitting procedure, the output of 2D keypoint detection has been used as additional guidance. The progress in 2D pose estimation using CNNs [65, 7, 15] has contributed significantly to the task of 3D pose estimation even in challenging in-the-wild scenarios. For example, the 3D parameters of SMPL [28] can be obtained with reasonable accuracy by fitting it to 2D keypoints [6, 24]. However, lifting to 3D from 2D information alone is an ambiguous problem. Adversarial learning can potentially identify plausible poses [18, 22]. By contrast,

the seminal works of [57, 53] address lifting by reasoning about kinematic depth ambiguities. Recently, using monocular video and geometric reasoning, accurate and detailed 3D shape, including clothing is obtained [3, 2].

Learning-Based Models. Recent methods in this category typically predict 3D keypoints or stick figures from a single image using a CNN. High capacity models are trained on standard 3D datasets [17, 49], which are limited in terms of appearance variation, pose, backgrounds and occlusions. Consequently, it is not clear – despite excellent performance on standard benchmarks – how methods [59, 25, 37, 26] generalize to in-the-wild images. To add variation, some methods resort to generating synthetic images [46, 64, 23] but it is complex to approximate fully realistic images with sufficient variance. Similar to model-based methods, learning approaches have benefited from the advent of robust 2D pose methods – by matching 2D detections to a 3D pose database [8, 66], by regressing pose from 2D joint distance matrices [35], by exploiting pose and geometric priors for lifting [69, 1, 51, 19, 32, 70, 47]; or simply by training a feed forward network to directly predict 3D pose from 2D joints [30]. Another way to exploit images with only 2D data is by re-using the first layers of a 2D pose CNN for the task of 3D pose estimation [60, 33, 31]. Pavlakos et al. [36] take another approach by relying on weak 3D supervision in form of a relative 3D ordering of joints, similar to the previously proposed PoseBits [40].

Closer to ours are approaches that train using separate 2D and 3D losses [44, 67, 55]. However, since they do not integrate a statistical body model in the network, limbs and body proportions might be unnatural. Most importantly, they only predict 3D stick figures as opposed to a full mesh.

Some works regress correspondences to a body model which are then used to fit the model to depth data [43, 58]. Recently, correspondences to the SMPL body surface are regressed from images directly [12] by leveraging dense keypoint annotations; however, the approach can not recover 3D human pose and shape. [63] fits SMPL to CNN volumetric outputs as a post-process step. 3D model fitting within a CNN have been proposed for faces [61]; faces however, are not articulated like bodies which simplifies the regression problem.

A few recent works (concurrent to ours) integrate the SMPL [28] model within a network [62, 22, 38]. The approaches differ primarily in the proxy representation used to lift to 3D: RGB images [22], images and 2D keypoints [62] and 2D keypoints and silhouettes [38], and the kind of supervision (3D vs 2D) used for training. In contrast to previous work, we analyze the importance of such components for good performance. Kanazawa et al. [22] also integrate a learned prior on the space of poses. We draw inspiration from model-based and learning approaches in several aspects of our model design. Firstly, NBF addresses an impor-

tant limitation: it does not require an initialization for optimization because it incorporates a CNN based bottom-up component. Furthermore, at test time, NBF predictions are fast and do not require optimization. By integrating SMPL directly into the CNN, we do not require multiple network heads to backpropagate 2D and 3D losses. Lastly, we use a semantic segmentation as proxy representation, which (1) abstracts away irrelevant image information for 3D pose, (2) is a richer semantic representation than keypoints or silhouettes, and (3) allows us to analyze the importance of part granularity and placement for 3D prediction.

3. Method

Our goal is to fit a 3D mesh of a human to a single static image (see Figure 2). This task involves multiple steps and we want to apply 3D but also 2D losses due to the strongly varying difficulty to obtain ground truth data. Nevertheless, we aim to build a simple processing pipeline with parts that can be optimized in isolation and avoiding multiple network heads. This reduces the number of hyperparameters and interactions, *e.g.*, loss weights, and allows us to consecutively train the model parts. There are two main stages in the proposed architecture: in a first stage, a body part segmentation is predicted from the RGB image. The second stage takes this segmentation to predict a low-dimensional parameterization of a body mesh.

3.1. Body Model

For our experiments we use the SMPL body model due to its good trade-off between high anatomic flexibility and realism. SMPL parameterizes a triangulated mesh with $N = 6890$ vertices with pose parameters $\theta \in \mathbb{R}^{72}$ and shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ – optionally the translation parameters $\gamma \in \mathbb{R}^3$ can be taken into account as well. Shape $B_s(\boldsymbol{\beta})$ and pose dependent deformations $B_p(\boldsymbol{\theta})$ are first applied to a base template $\mathbf{T}_\mu$; then the mesh is posed by rotating each body part around skeleton joints $J(\boldsymbol{\beta})$ using a skinning function W:

$$M(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(T(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}), \qquad (1)$$

$$T(\boldsymbol{\beta}, \boldsymbol{\theta}) = \mathbf{T}_\mu + B_s(\boldsymbol{\beta}) + B_p(\boldsymbol{\theta}), \qquad (2)$$

where $M(\boldsymbol{\beta}, \boldsymbol{\theta})$ is the SMPL function, and $T(\boldsymbol{\beta}, \boldsymbol{\theta})$ outputs an intermediate mesh in a T-pose after pose and shape deformations are applied. SMPL produces realistic results using relatively simple mathematical operations – most importantly for us SMPL is fully differentiable with respect to pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$. All these operations, including the ones to determine projected points of a posed and parameterized 3D body, are available in Tensorflow. We use them to make the 3D body a part of our deep learning model.
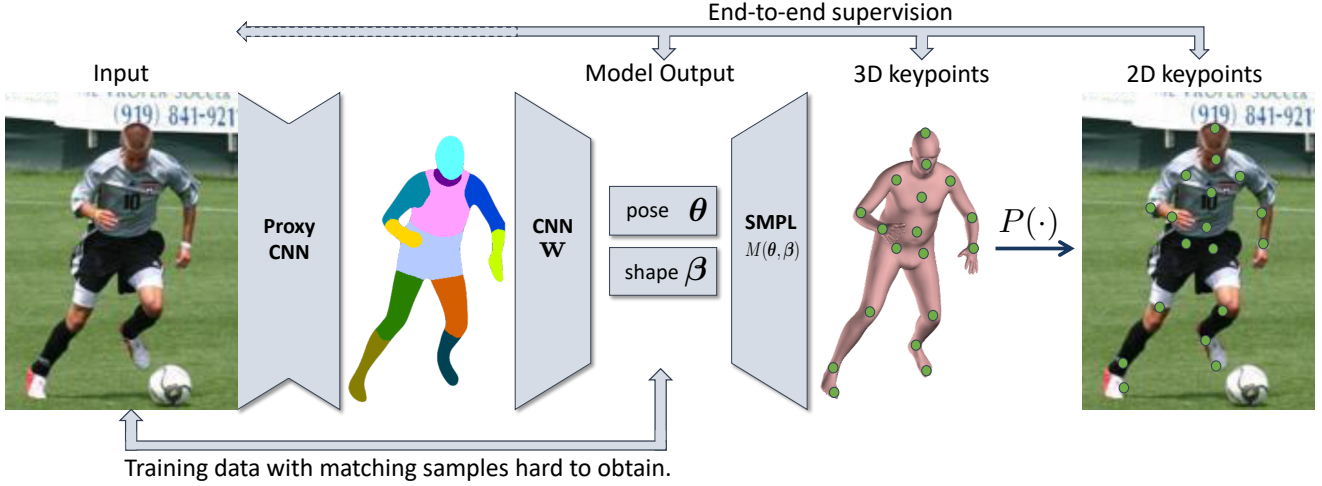
Figure 2: *Summary of our proposed pipeline.* We process the image with a standard semantic segmentation CNN into 12 semantic parts (see Sec. 4.2). An encoding CNN processes the semantic part probability maps to predict SMPL body model parameters (see Sec. 3.2). We then use our SMPL implementation in Tensorflow to obtain a projection of the pose-defining points to 2D. With these points, a loss on 2D vertex positions can be back propagated through the entire model (see Sec. 3.3).

3.2. Neural Body Fitting Parameterization

NBF predicts the parameters of the body model from a colour-coded part segmentation map $\mathbf{I} \in \mathbb{R}^{224 \times 224 \times 3}$ using a CNN-based predictor parameterized by weights w. The estimators for pose and shape are thus given by $\boldsymbol{\theta}(w, \mathbf{I})$ and $\boldsymbol{\beta}(w, \mathbf{I})$ respectively.

We integrate the SMPL model and a simple 2D projection layer into our CNN estimator, as described in Sec. 3.1. This allows us to output a 3D mesh, 3D skeleton joint locations or 2D joints, depending on the kind of supervision we want to apply for training while keeping the CNN monolithic.

Mathematically, the function $N_{3D}(w, \mathbf{I})$ that maps from semantic images to meshes is given by

$$
\begin{aligned}
N_{3D}(w, \mathbf{I}) &= M(\boldsymbol{\theta}(w, \mathbf{I}), \boldsymbol{\beta}(w, \mathbf{I})) \qquad (3) \\
&= W(T(\boldsymbol{\beta}(w, \mathbf{I}), \boldsymbol{\theta}(w, \mathbf{I}), \\
&\qquad J(\boldsymbol{\beta}(w, \mathbf{I})), \boldsymbol{\theta}(w, \mathbf{I}), \mathbf{W})), \quad (4)
\end{aligned}
$$

which is the SMPL Equation (1) parameterized by network parameters w. NBF can also predict the 3D joints $N_J(w, \mathbf{I}) = J(\boldsymbol{\beta}(w, \mathbf{I}))$, because they are a function of the model parameters. Furthermore, using a projection operation $\pi(\cdot)$ we can project the 3D joints onto the image plane

$$
N_{2D}(w, \mathbf{I}) = \pi(J(w, \mathbf{I})), \qquad (5)
$$

where $N_{2D}(w, \mathbf{I})$ is the NBF function that outputs 2D joint locations. All of these operations are differentiable and allow us to use gradient-based optimization to update model parameters with a suitable loss function.

3.3. Loss Functions

We experiment with the following loss functions:
3D latent parameter loss: This is an L1 loss on the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. Given a paired dataset $\{\mathbf{I}_i, \boldsymbol{\theta}_i, \boldsymbol{\beta}_i\}_i^N$, the loss is given by:

$$
\mathcal{L}_{lat}(w) = \sum_i^N |\mathbf{r}(\boldsymbol{\theta}(w, \mathbf{I}_i)) - \mathbf{r}(\boldsymbol{\theta}_i)| + |\boldsymbol{\beta}(w, \mathbf{I}_i) - \boldsymbol{\beta}_i|, \quad (6)
$$

where $\mathbf{r}$ are the vectorized rotation matrices of the 24 parts of the body. Similar to [24, 38], we observed better performance by imposing the loss on the rotation matrix representation of $\boldsymbol{\theta}$ rather than on its 'native' axis angle encoding as defined in SMPL. This requires us to project the predicted matrices to the manifold of rotation matrices. We perform this step using SVD to maintain differentiability.
3D joint loss: Given a paired dataset with skeleton annotations $\{\mathbf{I}_i, \boldsymbol{\theta}_i, \mathbf{J}\}_i^N$ we compute the loss in terms of 3D joint position differences as:

$$
\mathcal{L}_{3D}(w) = \sum_i^N \|N_J(w, \mathbf{I}_i) - \mathbf{J}_i\|^2 \qquad (7)
$$

2D joint loss: If the dataset $\{\mathbf{I}_i, \mathbf{J}_{2D}\}_i^N$ provides solely 2D joint position ground truth, we define a similar loss in terms of 2D distance and rely on error backpropagation through the projection:

$$
\mathcal{L}_{2D}(w) = \sum_i^N \|N_{2D}(w, \mathbf{I}_i) - \mathbf{J}_{2D,i}\|^2 \qquad (8)
$$

Joint 2D and 3D loss: To maximize the amounts of usable training data, ideally multiple data sources can be combined with a subset of the data $\mathcal{D}_{3D}$ providing 3D annotations and another subset $\mathcal{D}_{3D}$ providing 2D annotations. We can trivially integrate all the data with different kinds of supervision by falling back to the relevant losses and setting them to zero if not applicable.

$$\mathcal{L}_{2D+3D}(w, \mathcal{D}) = \mathcal{L}_{2D}(w, \mathcal{D}_{2D}) + \mathcal{L}_{3D}(w, \mathcal{D}_{3D}) \quad (9)$$

In our experiments, we analyze the performance of each loss and their combinations. In particular, we evaluate how much gain in 3D estimation accuracy can be obtained from weak 2D annotations which are much cheaper to obtain than accurate 3D annotations.

4. Results

4.1. Evaluation Settings

We used the following three datasets for evaluation: UP-3D, [24], HumanEva-I [49] and Human3.6M[17]. We perform a detailed analysis of our approach on UP-3D and Human3.6M, and compare against state-of-the-art methods on HumanEVA-I and Human3.6M.

UP-3D, is a challenging, in-the-wild dataset that draws from existing pose datasets: LSP [20], LSP-extended [20], MPII HumanPose [4], and FashionPose [10]. It augments images from these datasets with rich 3D annotations in the form of SMPL model parameters that fully capture shape and pose, allowing us to derive 2D and 3D joint as well as fine-grained segmentation annotations. The dataset consists of training (5703 images), validation (1423 images) and test (1389 images) sets. For our analysis, we use the training set and provide results on the validation set.

The HumanEVA-I dataset is recorded in a controlled environment with marker-based ground truth synchronized with video. The dataset includes 3 subjects and 2 motion sequences per subject. Human3.6M also consists of similarly recorded data but covers more subjects, with 15 action sequences per subject repeated over two trials. For our analysis on this dataset, we reserve subjects S1, S5, S6 and S7 for training, holding out subject S8 for validation. We compare to the state of art on the test sequences S9 and S11.

4.2. Implementation Details

Data preparation: To train our model, we require images paired with 3D body model fits (i.e. SMPL parameters) as well as pixelwise part labels. The UP-3D dataset provides such annotations, while Human3.6M does not. However, by applying MoSH [29] to the 3D MoCap marker data provided by the latter we obtain the corresponding SMPL parameters, which in turn allows us to generate part labels by rendering an appropriately annotated SMPL mesh [24].

Scale ambiguity: The SMPL shape parameters encode among other factors a person's size. Additionally, both distance to the camera and focal length determine how large a person appears in an image. To eliminate this ambiguity during training, we constrain scale information to the shape parameters by making the following assumptions: The camera is always at the SMPL coordinate origin, the optical axis always points in the same direction, and a person is always at a fixed distance from the camera. We render the ground truth SMPL fits and scale the training images to fit the renderings (using the corresponding 2D joints). This guarantees that the the only factor affecting person size in the image are the SMPL shape parameters. At test-time, we estimate person height and location in the image using 2D DeeperCut keypoints [16], and center the person within a 512x512 crop such that they have a height of 440px, which roughly corresponds to the setting seen during training.

Architecture: We use a two-stage approach: The first stage receives the 512x512 input crop and produces a part segmentation. We use a RefineNet [27] model (based on ResNet-101 [14]). This part segmentation is color-coded, resized to 224x224 and fed as an RGB image to the second stage, itself composed of two parts: a regression network (ResNet-50) that outputs the 226 SMPL parameters (shape and pose), and a non-trainable set of layers that implement the SMPL model and an image projection. Such layers can produce a 3D mesh, 3D joints or 2D joints given the predicted pose and shape. Training both stages requires a total of 18 (12+6) hours on a single Volta V100 GPU. More details are provided in the supplementary material as well as in the code (to be released).

4.3. Analysis

Which Input Encoding? We investigate here what input representation is effective for pose and shape prediction. Full RGB images certainly contain more information than for example silhouettes, part segmentations or 2D joints. However, some information may not be relevant for 3D inference, such as appearance, illumination or clothing, which might make the network overfit to nuisances factors

To this end, we train a network on different image representations and compare their performance on the UP-3D and Human3.6M validation sets. We compare RGB images, color-coded part segmentations of varying granularities, and color-coded joint heatmaps (see supplementary material for examples). We generate both using the ground truth SMPL annotations to establish an upper bound on performance, and later consider the case where we do not have access to such information at test time.

The results are reported in Table 1. We observe that explicit part representations (part segmentations or joint heatmaps) are more useful for 3D shape/pose estimation compared to RGB images and plain silhouettes. The dif-

type of input	UP	H36M
RGB	98.5	48.9
Segmentation (1 part)	95.5	43.0
Segmentation (3 parts)	36.5	37.5
Segmentation (6 parts)	29.4	36.2
Segmentation (12 parts)	27.8	33.5
Segmentation (24 parts)	28.8	31.8
Joints (14)	28.8	33.4
Joints (24)	27.7	33.4

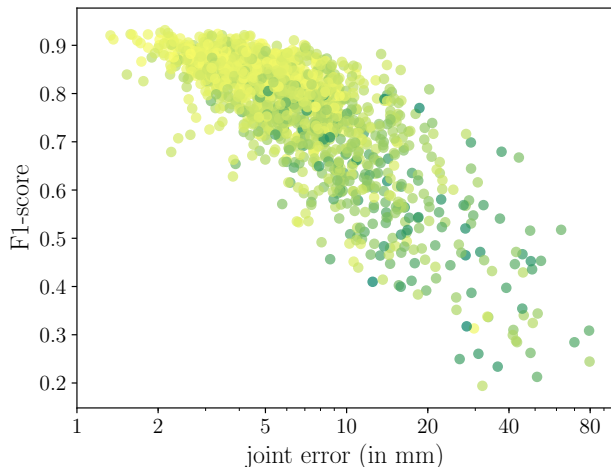Table 1: *Input Type vs. 3D error in millimeters*

Figure 3: *Segmentation quality (F1-score) vs. fit quality (3D joint error).* The darkness indicates the difficulty of the pose, i.e. the distance from the upright pose with arms by the sides.

Val \ Train	VGG	ResNet	RefineNet	GT
VGG	107.2	119.9	135.5	140.7
ResNet	97.1	96.3	112.2	115.6
RefineNet	89.6	89.9	82.0	83.3
GT	62.3	60.5	35.7	27.8

Table 2: *Effect of segmentation quality on the quality of the 3D fit prediction modules ($err_{joints3D}$)*

ference is especially pronounced on the UP-3D dataset, which contains more visual variety than the images of Human3.6M, with an error drop from 98.5 mm to 27.8 mm when using a 12 part segmentation. This demonstrates that a 2D segmentation of the person into sufficient parts carries a lot of information about 3D pose/shape, while also providing full spatial coverage of the person (compared to joint heatmaps). Is it then worth learning separate mappings first from image to part segmentation, and then from part segmentation to 3D shape/pose? To answer this question we first need to examine how 3D accuracy is affected by the quality of real predicted part segmentations.

Which Input Quality? To determine the effect of segmentation quality on the results, we train three different *part segmentation networks*. Besides RefineNet, we also train two variants of DeepLab [9], based on VGG-16 [52] and ResNet-101 [14]. These networks result in IoU scores of 67.1, 57.0, and 53.2 respectively on the UP validation set. Given these results, we then train four *3D prediction networks* - one for each of the part segmentation networks, and an additional one using the ground truth segmentations. We report 3D accuracy on the validation set of UP3D for each of the four 3D networks, diagonal numbers of Table 2. As one would expect, the better the segmentation, the better the 3D prediction accuracy. As can also be seen in Table 2, better segmenters at test time always lead to improved 3D accuracy, even when the 3D prediction networks are trained with poorer segmenters. This is perhaps surprising, and it indicates that mimicking the statistics of a particular segmentation method at training time plays only a minor role (for example a network trained with GT segmentations and tested using RefineNet segmentations performs comparably to a network that is trained using RefineNet segmentations (83.3mm vs 82mm)). To further analyze the correlation between segmentation quality and 3D accuracy, in Figure 3 we plot the relationship between F-1 score and 3D reconstruction error. Each dot represents one image, and the color its respective difficulty – we use the distance to mean pose as a proxy measure for difficulty. The plot clearly shows that

the higher the F-1 score, the lower the 3D joint error.

Which Types of Supervision? We now examine different combinations of loss terms. The losses we consider are L_{lat} (on the latent parameters), L_{3D} (on 3D joint/vertex locations), L_{2D} (on the projected joint/vertex locations). We compare performance using three different error measures: (i) $err_{joints3D}$, the Euclidean distance between ground truth and predicted SMPL joints (in mm). (ii) $PCKh$ [4], the percentage of correct keypoints with the error threshold being 50% of head size, which we measure on a per-example basis. (iii) err_{quat}, quaternion distance error of the predicted joint rotations (in radians).

Given sufficient data - the full 3D-annotated UTP training set with mirrored examples (11406) - only applying a loss on the model parameters yields reasonable results, and in this setting, additional loss terms don't provide benefits. When only training with L_{3D}, we obtain similar results in terms of $err_{joints3D}$, however, interestingly err_{quat} is significantly higher. This indicates that predictions produce accurate 3D joints positions in space, but the limb orientations are incorrect. This further demonstrates that methods trained to produce only 3D keypoints do not capture orien-

Loss		$\text{err}_{\text{joints3D}}$	PCKh	err_{quat}
L_{lat}		83.7	93.1	0.278
$L_{lat} + L_{3D}$		82.3	93.4	0.280
$L_{lat} + L_{2D}$		83.1	93.5	0.278
$L_{lat} + L_{3D} + L_{2D}$		82.0	93.5	0.279
L_{3D}		83.7	93.5	1.962
L_{2D}		198.0	94.0	1.971

Table 3: *Loss ablation study.* Results in 2D and 3D error metrics (*joints3D*: Euclidean 3D distance, *mesh*: average vertex to vertex distance, *quat*: average body part rotation error in radians).

Ann.perc. / Error	100	50	20	10	5	2	1	0
$\text{err}_{\text{joints3D}}$	83.1	82.8	82.8	83.6	84.5	88.1	93.9	198
err_{quat}	0.28	0.28	0.27	0.28	0.29	0.30	0.33	1.97

Table 4: *Effect of 3D labeled data.* We show the 3D as well as the estimated body part rotation error for varying ratios of data with 3D labels. For all of the data, we assume that 2D pose labels are available. Both errors saturate at 20% of 3D labeled training examples.

Method	Mean	Median
Ramakrishna et al. [45]	168.4	145.9
Zhou et al. [68]	110.0	98.9
SMPLify [6]	79.9	61.9
Random Forests [24]	93.5	77.6
SMPLify (Dense) [24]	74.5	59.6
Ours	64.0	49.4

Table 5: **HumanEva-I results.** 3D joint errors in mm.

Method	Mean	Median
Akhter & Black [1]	181.1	158.1
Ramakrishna et al. [45]	157.3	136.8
Zhou et al. [68]	106.7	90.0
SMPLify [6]	82.3	69.3
SMPLify (dense) [24]	80.7	70.0
SelfSup [62]	98.4	-
Pavlakos et al. [38]	75.9	-
HMR (H36M-trained)[22]	77.6	72.1
HMR [22]	**56.8**	-
Ours	59.9	52.3

Table 6: **Human 3.6M.** 3D joint errors in mm.

tation, which is needed for many applications.

We also observe that only training with the 2D reprojection loss (perhaps unsurprisingly) results in poor performance in terms of 3D error, showing that some amount of 3D annotations are necessary to overcome the ambiguity inherent to 2D keypoints as a source of supervision for 3D.

Due to the SMPL layers, we can supervise learning with any number of joints/mesh vertices. We thus experimented with the 91 landmarks used by [24] for their fitting method but find that the 24 SMPL joints are sufficient in this setting.

How Much 3D Supervision Do We Need? The use of these additional loss terms also allows us to leverage data for which no 3D annotations are available. With the following set of experiments, we attempt to answer two questions: (i) Given a small amount of 3D-annotated data, does extra 2D-annotated data help?, (ii) What amount of 3D data is necessary? To this end we train multiple networks, each time progressively disabling the 3D latent loss and replacing it with the 2D loss for more training examples. The results are depicted in Table 4. We find that performance barely degrades as long as we have a small amount of 3D annotations. In contrast, using small amounts of 3D data and no extra data with 2D annotations yields poor performance. This is an important finding since obtaining 3D annotations is difficult compared to simple 2D keypoint annotations.

Qualitative Results A selection of qualitative results from the UP-3D dataset can be found in Figure 4. We show examples from the four different error quartiles. Fits from the first three quartiles still reproduce the body pose somewhat faithfully, and only in the last row and percentile, problems become clearly visible. We show more failure modes in the supplementary material.

4.4. Comparison to State-of-the-Art

Here we compare to the state of the art on HumanEva-I (Table 5) and Human3.6M (Table 6). We perform a per-frame rigid alignment of the 3D estimates to the ground truth using Procrustes Analysis and report results in terms of reconstruction error, i.e. the mean per joint position error after alignment (given in mm). The model we use here is trained on Human3.6M data.

We compare favourably to similar methods, but these are not strictly comparable since they train on different datasets. Pavlakos et al. [38] do not use any data from Human3.6M, whereas HMR [22] does, along with several other datasets. We retrained the latter with the original code only using Human3.6M data for a more direct comparison to ours (HMR (H36M-trained) in Table 6). Given Table 1, we hypothesize that their approach requires more training data for good performance because it uses RGB images as input.

5. Conclusion

In this paper, we make several principled steps towards a full integration of parametric 3D human pose models into
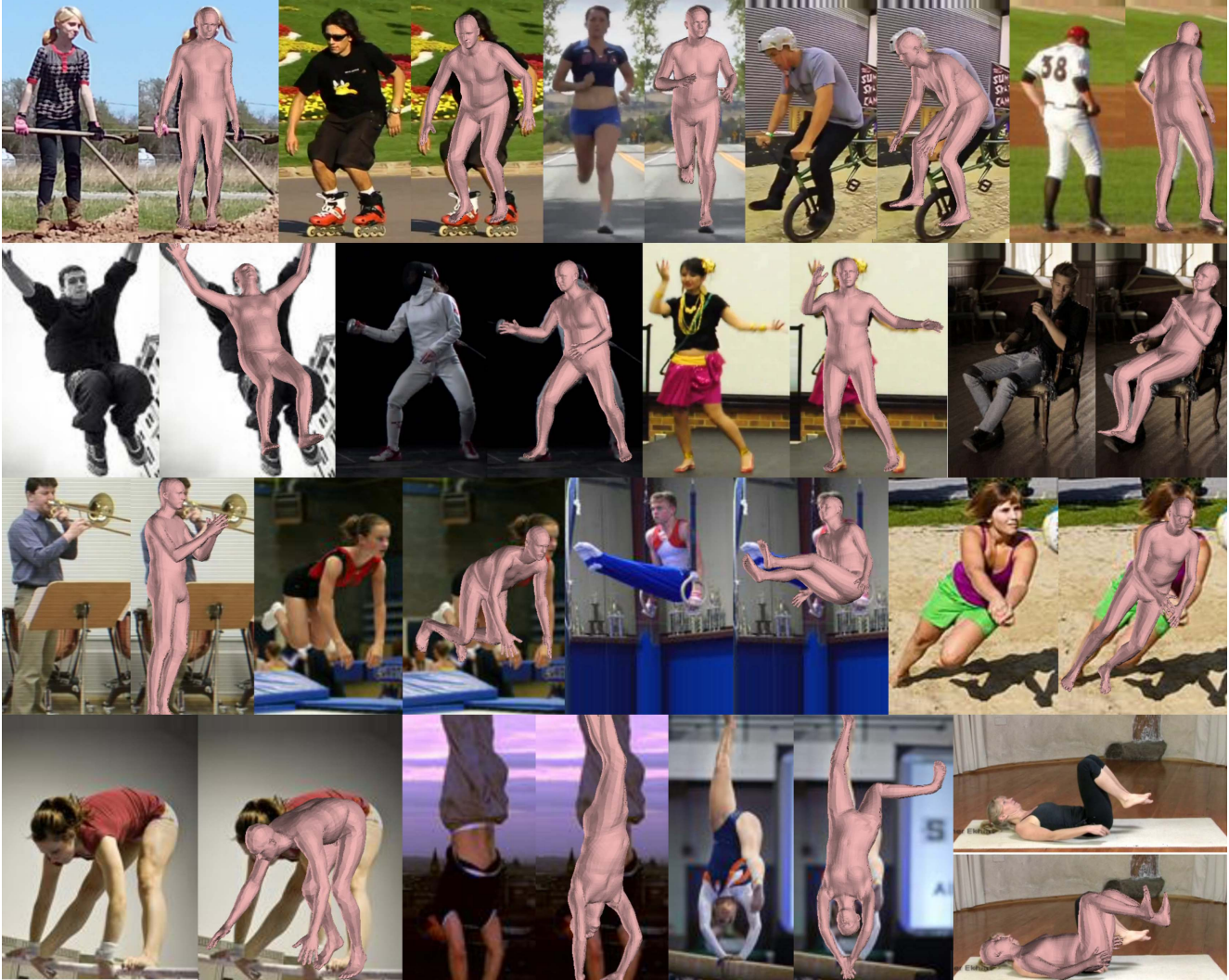
Figure 4: *Qualitative results by error quartile in terms of $err_{joints3D}$*. The rows show representative examples from different error quartiles, top to bottom: 0-25%, 25-50%, 50-75%, 75-100%

deep CNN architectures. We analyze (1) how the 3D model can be integrated into a deep neural network, (2) how loss functions can be combined and (3) how a training can be set up that works efficiently with scarce 3D data.

In contrast to existing methods we use a region-based 2D representation, namely a 12-body-part segmentation, as an intermediate step prior to the mapping to 3D shape and pose. This segmentation provides full spatial coverage of a person as opposed to the commonly used sparse set of keypoints, while also retaining enough information about the arrangement of parts to allow for effective lifting to 3D.

We used a stack of CNN layers on top of a segmentation model to predict an encoding in the space of 3D model parameters, followed by a Tensorflow implementation of the 3D model and a projection to the image plane. This full integration allows us to finely tune the loss functions and

enables end-to-end training. We found a loss that combines 2D as well as 3D information to work best. The flexible implementation allowed us to experiment with the 3D losses only for parts of the data, moving towards a weakly supervised training scenario that avoids expensive 3D labeled data. With 3D information for only 20% of our training data, we could reach similar performance as with full 3D annotations.

We believe that this encouraging result is an important finding for the design of future datasets and the development of 3D prediction methods that do not require expensive 3D annotations for training. Future work will involve extending this to more challenging settings involving multiple, possibly occluded, people.

References

[1] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1455, 2015. 3, 7

[2] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Detailed human avatars from monocular video. In *3D Vision (3DV), 2018 Sixth International Conference on*, 2018. 3

[3] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Video based reconstruction of 3d people models. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. 3

[4] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 5, 6

[5] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. 2

[6] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 7

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Real-time multi-person 2d pose estimation using part affinity fields. In *Proc. of CVPR*, 2016. 2

[8] C.-H. Chen and D. Ramanan. 3d human pose estimation = 2d pose estimation + matching. In *CVPR 2017-IEEE Conference on Computer Vision & Pattern Recognition*, 2017. 3

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 6

[10] M. Dantone, J. Gall, C. Leistner, and L. V. Gool. Body parts dependent joint regressors for human pose estimation in still images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2131–2143, Nov 2014. 5

[11] D. M. Gavrila and L. S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Proc. CVPR*, pages 73–80. IEEE, 1996. 2

[12] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. *arXiv preprint arXiv:1802.00434*, 2018. 3

[13] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. In *Computer Graphics Forum*, volume 28, pages 337–346, 2009. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 6

[15] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. Arttrack: Articulated multi-person tracking in the wild. In *30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, Honolulu, HI, USA, 2017. IEEE. 2

[16] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision (ECCV)*, 2016. 5

[17] 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 (PAMI)*, 36(7):1325–1339, 2014. 3, 5

[18] D. Jack, F. Maire, A. Eriksson, and S. Shirazi. Adversarially parameterized optimization for 3d human pose estimation. 2017. 2

[19] E. Jahangiri and A. L. Yuille. Generating multiple diverse hypotheses for human 3d pose consistent with 2d joint detections. In *IEEE International Conference on Computer Vision (ICCV) Workshops (PeopleCap)*, 2017. 3

[20] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2010. doi:10.5244/C.24.12. 5

[21] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. *arXiv preprint arXiv:1801.01615*, 2018. 2

[22] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 7, 8

[23] C. Lassner, G. Pons-Moll, and P. V. Gehler. A generative model of people in clothing. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[24] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 4, 5, 7

[25] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision (ACCV)*, pages 332–347, 2014. 3

[26] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2848–2856, 2015. 3

[27] G. Lin and I. R. Anton Milan, Chunhua Shen. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[28] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 1, 2, 3

[29] 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. 5

[30] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[31] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 3

[32] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *3D Vision (3DV), 2018 Sixth International Conference on*, 2018. 3

[33] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics*, 36(4), July 2017. 3

[34] D. Metaxas and D. Terzopoulos. Shape and non-rigid motion estimation through physics-based synthesis. *IEEE Trans. PAMI*, 15(6):580–591, 1993. 2

[35] F. Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *CVPR 2017-IEEE Conference on Computer Vision & Pattern Recognition*, 2017. 3

[36] G. Pavlakos, X. Zhou, and K. Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[37] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR 2017-IEEE Conference on Computer Vision & Pattern Recognition*, 2017. 3

[38] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3, 4, 7

[39] R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *International Conference on Computer Vision, Vancouver, Canada*, number CVLAB-CONF-2001-005, pages 394–401, 2001. 2

[40] G. Pons-Moll, D. J. Fleet, and B. Rosenhahn. Posebits for monocular human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2344, 2014. 3

[41] G. Pons-Moll, J. Romero, N. Mahmood, and M. J. Black. Dyna: a model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)*, 34:120, 2015. 2

[42] G. Pons-Moll and B. Rosenhahn. *Model-Based Pose Estimation*, chapter 9, pages 139–170. Springer, 2011. 2

[43] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric regression forests for correspondence estimation. *International Journal of Computer Vision*, 113(3):163–175, 2015. 3

[44] A.-I. Popa, M. Zanfir, and C. Sminchisescu. Deep multitask architecture for integrated 2d and 3d human sensing. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[45] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *European Conference on Computer Vision*, pages 573–586. Springer, 2012. 7

[46] G. Rogez and C. Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Advances in Neural Information Processing Systems*, pages 3108–3116, 2016. 3

[47] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *arXiv preprint arXiv:1803.00455*, 2018. 3

[48] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016. 2

[49] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and

baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision (IJCV)*, 87(1-2):4–27, 2010. 3, 5

[50] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–421. IEEE, 2004. 2

[51] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A joint model for 2d and 3d pose estimation from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3634–3641, 2013. 3

[52] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[53] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003. 3

[54] C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *Proc. ICCV*, pages 951–958. IEEE, 2011. 2

[55] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. *arXiv preprint arXiv:1704.00159*, 2017. 3

[56] J. K. V. Tan, I. Budvytis, and R. Cipolla. Indirect deep structured learning for 3d human body shape and pose prediction. In *BMVC*, volume 3, page 6, 2017. 1

[57] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 677–684, 2000. 3

[58] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 103–110. IEEE, 2012. 3

[59] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured Prediction of 3D Human Pose with Deep Neural Networks. In *British Machine Vision Conference (BMVC)*, 2016. 3

[60] B. Tekin, P. Márquez-Neila, M. Salzmann, and P. Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[61] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2017. 3

[62] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems*, pages 5242–5252, 2017. 1, 3, 7

[63] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. Bodynet: Volumetric inference of 3d human body shapes. *arXiv preprint arXiv:1804.04875*, 2018. 3

[64] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from Synthetic Humans. In *CVPR*, 2017. 3

[65] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[66] H. Yasin, U. Iqbal, B. Krüger, A. Weber, and J. Gall. A Dual-Source Approach for 3D Pose Estimation from a Single Image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[67] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 398–407, 2017. 3

[68] X. Zhou, S. Leonardos, X. Hu, and K. Daniilidis. 3D shape estimation from 2D landmarks: A convex relaxation approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4447–4455, 2015. 7

[69] X. Zhou, M. Zhu, S. Leonardos, K. Derpanis, and K. Daniilidis. Sparseness Meets Deepness: 3D Human Pose Estimation from Monocular Video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[70] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 3

[71] S. Zuffi and M. J. Black. The stitched puppet: A graphical model of 3d human shape and pose. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3537–3546. IEEE, 2015. 2

A. Further Qualitative Results

One of our findings is the high correlation between input segmentation quality and output fit quality. We provide some additional qualitative examples that illustrate this correlation. In Fig. 5, we present the four worst examples from the validation set in terms of 3D joint reconstruction error when we use our trained part segmentation network; in Fig. 6, we present the worst examples when the network is trained to predict body model parameters given the ground truth segmentations. This does not correct all estimated 3D bodies, but the remaining errors are noticeably less severe.

B. Training Details

We present examples of paired training examples and ground truth in Fig 7.

Segmentation Network We train our own TensorFlow implementation of a RefineNet [4] network (based on ResNet-101) to predict the part segmentations. The images are cropped to 512x512 pixels, and we train for 20 epochs with a batch size of 5 using the Adam [3] optimizer. Learning rate and weight decay are set to 0.00002 and 0.0001 respectively, with a polynomial learning rate decay. Data augmentation improved performance a lot, in particular horizontal reflection (which requires re-mapping the labels for left and right limbs), scale augmentation (0.9 - 1.1 of the original size) as well as rotations (up to 45 degrees). For training the segmentation network on UP-3D we used the 5703 training images. For Human3.6M we subsampled the videos, only using every 10th frame from each video, which results in about 32000 frames. Depending on the amount of data, training the segmentation networks takes about 6-12 hours on a Volta V100 machine.

Fitting Network For the fitting network we repurpose a ResNet-50 network pretrained on ImageNet to regress the SMPL model parameters. We replace the final pooling layer with a single fully-connected layer that outputs the 10 shape and 216 pose parameters. We train this network for 75 epochs with a batch size of 5 using the Adam optimizer. The learning rate is set to 0.00004 with polynomial decay and we use a weight decay setting of 0.0001. We found that an L1 loss on the SMPL parameters was a little better than an L2 loss. We also experimented with robust losses (e.g. Geman-McClure [2] and Tukey's biweight loss [1]) but did not observe benefits. Training this network takes about 1.5 hours for the UP-3D dataset and six hours for Human3.6M.

Data Augmentation At test-time we cannot guarantee that the person will be perfectly centered in the input crop, which can lead to degraded performance. We found it thus critical to train both the segmentation network and the fitting network with strong data augmentation, especially by introducing random jitter and scaling. For the fitting network, such augmentation has to take place prior to training since it affects the SMPL parameters. We also mirror the data, but this requires careful mirroring of both the part labels as well as the SMPL parameters. This involves remapping the parts, as well as inverting the part rotations.

References

[1] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab. Robust optimization for deep regression. 2015.

[2] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21, 1987.

[3] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2014.

[4] G. Lin and I. R. Anton Milan, Chunhua Shen. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
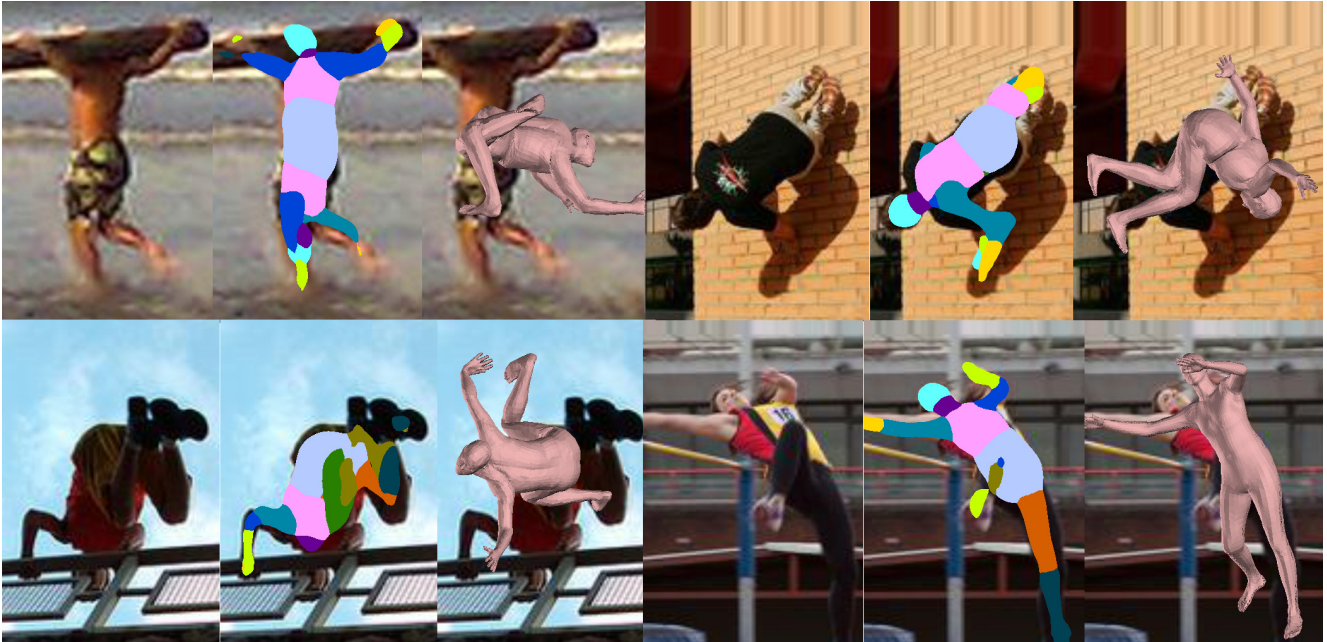
Figure 5: Worst examples from the validation set in terms of 3D error given imperfect segmentations.

Figure 6: Worst examples from the validation set in terms of 3D error given perfect segmentations.
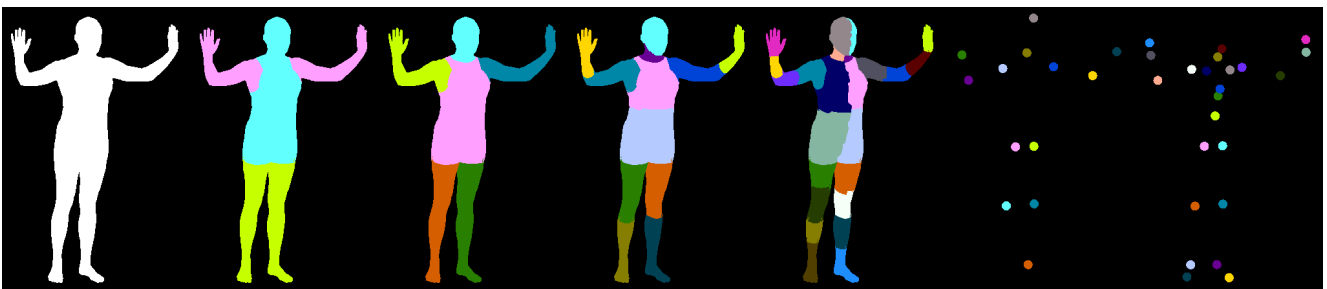

Figure 7: Example training images annotations illustrating different types and granularities.