3D Human Pose Estimation via Explicit Compositional Depth Maps

Haiping Wu,[1]* **Bin Xiao**[2]
[1]McGill University, [2]BydeDance AI Lab
haiping.wu2@mail.mcgill.ca, xiaobin.ailab@bytedance.com

Abstract

In this work, we tackle the problem of estimating 3D human pose in camera space from a monocular image. First, we propose to use densely-generated limb depth maps to ease the learning of body joints depth, which are well aligned with image cues. Then, we design a lifting module from 2D pixel coordinates to 3D camera coordinates which explicitly takes the depth values as inputs, and is aligned with camera perspective projection model. We show our method achieves superior performance on large-scale 3D pose datasets Human3.6M and MPI-INF-3DHP, and sets the new state-of-the-art.

Introduction

In this work, we aim to tackle the problem of estimating 3D human pose from a monocular RGB image. The problem is inherently ambiguous, since there could be as many 3D human poses that have the same projected 2D pose. While being hard, this task serves as foundation of many applications, such as surveillance, human action/activity recognition, human computer/robot interaction, etc.

Existing methods typically fall into two categories. The first one (Pavlakos et al. 2017) directly estimates 3D coordinates (X, Y, Z) from images, without any intermediate 2D pose information supervision. The second one estimates 2D pixel coordinates (x, y) in the images first, then lifts to 3D pose using either learned transformation (Martinez et al. 2017), ground truth camera parameters (Sun et al. 2018) or re-scaling (Zhou et al. 2017) by weak perspective projection assumption. Decoupling the learning process of 3D pose coordinates (X, Y, Z) into learning the pixel coordinates (x, y) first, then lifting to 3D pose shows better generalization power. This is due to the relatively maturity of 2D human pose estimation methods and easily annotated large-scale 2D human pose datasets (e.g. COCO, MPII). In contrast, 3D human pose datasets are hard to capture, often limited in studio environment (Ionescu et al. 2014).

3D coordinates of (X, Y, Z) are not equally observed in the sense that monocular images only have the projected (X, Y) captured. For the depth representation of joints, a single scalar value is the common learning target among existing methods (Martinez et al. 2017). However, this representation does not explicitly link the depth values to the image locations, which indicates the learning process needs localizing the joints and estimating the depth values of them concurrently. Nevertheless, state-of-the-art 2D human pose estimation methods typically obtain 2D joints locations via estimating generated heatmaps, which are well aligned with images. Inspired by this, we choose to predict depth values in the form of depth value maps which are aligned with images.

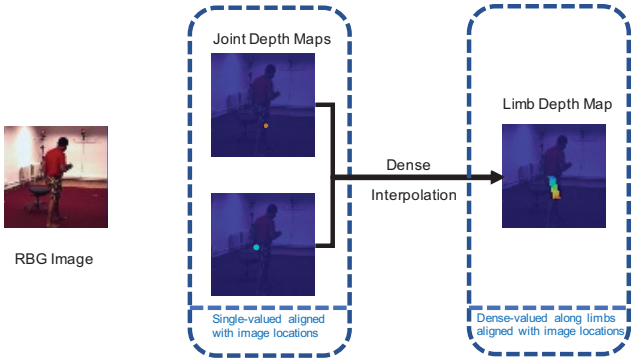

Figure 1: Joint Depth Maps and Limb Depth Maps representation for depth values. Both aligned with image locations. Depth values of points on limbs are interpolated from joints of the limbs by assuming limbs are rigid parts. For simplicity, only Joint Depth Maps of left knee and left hip are shown, and the Limb Depth Map of the limb connecting left knee and left hip are shown.

Naively, depth maps could be generated by assigning the depth values of joints to the corresponding pixel locations, leaving the value of other non-joint locations zero. We refer this method as Joint Depth Maps. Joint Depth Maps representation manages to associate depth values with image cues, however it gives too sparse supervision signal for learning. Further, we propose to composite dense-valued depth maps by assuming the limbs are rigid and thus inter-

*The work was done when Haiping Wu was an intern at ByteDance AI Lab.

polating dense values along the limbs. Figure 1 shows an example of generated Joint Depth Maps and Limb Depth Maps.

Many methods first obtain the 2D pixel coordinates of human joints first, then lift to 3D camera coordinates. For the lifting process, some methods (Sun et al. 2018) utilize ground truth camera parameters, which are not generally available in the real world applications. Some methods (Zhou et al. 2017) re-scale the human skeleton using average skeleton scale by assuming a weak perspective projection model. However, this assumption does not generally hold true and would introduce errors which cannot be corrected since they consider the lifting process as a post-processing stage and is not modeled in an end-to-end fashion. Others (Martinez et al. 2017) (Pavllo et al. 2019) utilize fully-connected networks or convolutional neural networks to modeling the lifting process, often taking the vectorized 2D coordinates or 2D heatmaps as inputs. However, by looking into the perspective projection model, (X, Y) locations in camera coordinates converted from (x, y) in pixel coordinates would need camera parameters as well as depth values of each joint. Existing methods predict (X, Y, Z) values simultaneously, which is a nested process. Therefore, we propose to decouple the process by first explicitly estimating depth values from images, then predicting (X, Y) given the extra depth values information.

In this work, we propose a novel framework for 3D human pose estimation. The pipeline first estimates 2D pixel coordinates (x, y) using existing 2D human pose estimation modules, as well as depth values represented as densely-generated Limb Depth Maps are predicted. Then a lifting module is designed for converting pixel coordinates to the target camera coordinates explicitly given the learned depth values. Figure 2 shows the pipeline of our method.

We show that our method achieves the new state-of-the-arts performance on the public available large-scale 3D human pose dataset Human3.6M (Ionescu et al. 2014) and in-the-wild MPI-INF-3DHP benchmark (Mehta et al. 2017a) with a simple pipeline. Our contributions could be summarized as follows.

- We propose a new representation for depth values of human joints, Limb Depth Maps, to ease the learning. The maps are generated by densely interpolating depth values along limbs under the assumption that limbs are rigid.
- We design a lifting module from 2D to 3D pose, which explicitly takes the depth values as inputs, aligned with perspective projection model, and show empirically the advantage of it for the learning process.
- We show that our method significantly outperforms previous methods on Human3.6M and MPI-INF-3DHP datasets and show the generalization ability of our method to in-the-wild images.

Related Works

Human pose estimation has been actively studied in the computer vision community. We review some recent and relevant works on 2D and 3D pose estimation.

2D pose estimation Given the easily annotated 2D human pose datasets, 2D human pose estimation has been greatly improved recently. CPM (Wei et al. 2016) proposes to use multi-stage convolution networks for refining predictions. Hourglass (Newell, Yang, and Deng 2016) uses repeated bottom-up, top-down architectures with intermediate supervision for better localizing joints. CPN (Chen et al. 2018) adapts a cascade pyramid networks to relieve the learning of 'hard' keypoints. SimpleBaselines (Xiao, Wu, and Wei 2018) proposes to add a few transposed convolution layers upon the output of ResNet, achieves state-of-the-art performance while keeping a relatively simple pipeline. HRNet (Sun et al. 2019) designs a network which could maintain high-resolution representation, leading to superior performance on 2D human pose estimation. We utilize SimpleBaselines (Xiao, Wu, and Wei 2018) and HRNet (Sun et al. 2019) as the feature extraction networks in this work for the simplicity and state-of-the-art performance.

3D pose estimation from 2D keypoints A line of works try to estimate 3D human pose directly from 2D joints coordinates, which implicitly model the human structures. (Martinez et al. 2017) achieves reasonable performance by directly predicting 3D coordinates from vectorized 2D coordinates with several fully-connected layers and residual connections. (Chen and Ramanan 2017) builds a library of 3D poses and find the most similar one that matches the detected 2D coordinates. (Zhao et al. 2019) proposes to utilize Graph Convolutional Networks with semantics to better modeling the process of lifting from 2D coordinates to 3D pose. (Pavllo et al. 2019) adds temporal convolution networks to incorporate multi frames' information to obtain temporal-consistent 3D human pose. However, these methods do not exploit the full image features for the depth estimation, where many rich image cues could be used to alleviate the ambiguous nature of 3D human pose estimation.

3D pose estimation from images Many approaches utilize the advance of deep learning methods, such as Convolutional Neural Networks (CNN), to address the 3D pose estimation problem. (Li and Chan 2014) models the problem as direct coordinates regression from images. (Tekin et al. 2016) uses a pose autoencoder to capture human body structure. (Pavlakos et al. 2017) represents 3D coordinates as voxels in 3D grids, and predicts a 3D Guassian-like volumetric representation in a coarse-to-fine manner. (Zhou et al. 2017) additional predicts 2D heatmaps, while adding 2D pose dataset such as MPII to learning process for generalization. (Sun et al. 2017) imposes constrains by additional predicts joint-to-joint (limb) relationships. (Sun et al. 2018) uses a soft argmax operation to obtain the 2D locations and the depth values. In this work, we also choose to utilize image cues into the learning process.

(Luo, Chu, and Yuille 2018) generates limb orientation maps, an extension of 2D Part Affinity Fields (Cao et al. 2017), to obtain the depth value of joints. All points on the same limb would have the same unit orientation vector indicating the direction of the limb in 3D camera space, and would need limb length information to recover the absolute depth of joints. Different from (Luo, Chu, and Yuille 2018),

we directly learn depth values of points on the limbs, and during inference, no extra limb scale information is needed.

(Habibie et al. 2019) proposes to embed 3D pose cues in latent space of the learning process, however there is no depth supervision to the hidden feature maps. Different from (Habibie et al. 2019), we propose to explicit give supervisions of depth to make it more meaningful, and show that it is important to do so for the lifting process to 3D pose.

3D pose estimation with constrains 3D human pose estimation from monocular images is an inherently ambiguous problem. Many works try to regularize the learning process with various constrains. (Zhou et al. 2017) imposes geometric constrain on bone length on 2D labeled only data. (Habibie et al. 2019) adds bone length loss to the predicted 3D pose, and utilizes a 3D to 2D reconstruction module to regularize the learned 3D pose. (Fang et al. 2018) defines pose grammar which allows incorporating human body configuration (i.e., kinematics, symmetry, motor coordination). (Pavlakos, Zhou, and Daniilidis 2018) uses ordinal depth annotation of 2D datasets, which is easier to get than 3D pose annotation, to alleviate the ambiguity of depth. These constraints are imposed on either 2D pixel coordinates outputs or 3D camera coordinates outputs. Our method could benefit in both situation since we have both the 2D pixel and 3D camera coordinates as outputs.

Method

In this section, first we describe the overall pipeline of our method. Then we elaborate two major components of the pipeline. First, we show how our method deals with the depth representation. Second, we show that explicitly providing depth values is necessary when converting from 2D pixel coordinates to 3D camera coordinates.

Overall Framework

The overall pipeline of estimating 3D human joints in camera coordinates from a monocular image is shown in Figure 2. First, a feature extractor takes in the image and outputs a set of feature maps of three kinds, namely the 2D heatmaps, depth maps and hidden feature maps. For the 2D heatmaps and depth maps, we impose supervision using ground truths. The hidden feature maps are used to capture extra information besides 2D locations and depth values of body joints, such as image features. Then a 3D lifting module takes in all the feature maps and outputs the 3D pose (X, Y, Z) in camera coordinates. The major difference of our lifting module to previous ones is that ours explicitly takes the depth values of joints as inputs for outputting X, Y in camera coordinates.

The framework could have the advantage of introducing large-scale 2D annotated-only data, which shows to have generalization ability for in-the-wild images. It also explicitly give intermediate supervision to depth values, which will be shown important for the lifting process in Eq. 6. The whole framework is trained end-to-end to output the camera coordinates of human joints. When inference, no ground truth camera parameters or limb/skeleton scale information is needed as in previous methods (Sun et al. 2018) (Zhou et al. 2017).

Many additional constrains are adapted in previous works, such as bone length (Zhou et al. 2017), symmetry or 3D to 2D reconstruction consistency (Habibie et al. 2019), temporal consistency (Pavllo et al. 2019). We do not add them in this work to better analyze the major components of our method and keep it a simple pipeline. Nevertheless, these techniques could be easily added upon our pipeline, which should further improve the performance.

Depth Representation

We discuss three possible representation for depth values. Given the 3D human joints locations $\mathcal{J} = \{\mathbf{J}_i\}_{i=1}^K$ in camera coordinates, where K is the number of joints, and the corresponding 2D locations $\mathcal{S} = \{\mathbf{S}_i\}_{i=1}^K$ in pixel coordinates.

Direct Scalar One naive way of getting depth values is through direct regression. Many previous works (Martinez et al. 2017) (Pavllo et al. 2019) use fully-connected layers or convolutional layers to obtain a vectorized representation for each joint, and a single scalar value for the depth dimension. It is used in methods in the category of lifting from images and lifting from 2D pixel coordinates. However, direct regression does not align depth values with images cues, which makes it of less generalization power, e.g. not scale invariant.

Joint Depth Maps Depth values for each joint could be linked with pixel coordinates by assigning the depth values of joints to corresponding pixel locations, resulting maps of depth values. Specifically, joint depth maps $\mathcal{D}^{Joint} = \{\mathbf{D}_k^{Joint}\}_{k=1}^K$, at an image point $\mathbf{p} = (x, y)$, are defined as the following formula:

$$\mathbf{D}_k^{Joint}(\mathbf{p}) = \begin{cases} Z_k & \text{if } (x_k, y_k) = (x, y) \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where (x_k, y_k) is the corresponding pixel location of joint k, Z_k is the depth value for joint k. In total, K joint depth maps are generated, one for each joint. Figure 1 shows an example of Joint Depth Maps.

Limb Depth Maps Joint depth maps associate depth values with image locations. However, it is too sparse as supervision signal, since for each joint depth map, there is at most one pixel location that has non-zero value. Take a step further, we propose to generate depth maps in a dense manner.

Typically, the depth values for two points in pixel coordinates are irrelevant with respect to the relationship of their locations. By assuming human body limbs as rigid parts, we could obtain coarse depth values of the points along limbs by interpolation. More specifically, given the 3D locations $\mathbf{J}_{k_1} = (X_{k_1}, Y_{k_1}, Z_{k_1})$, $\mathbf{J}_{k_2} = (X_{k_2}, Y_{k_2}, Z_{k_2})$ of joints k_1, k_2 belong to a limb L_{k_1,k_2}, and the corresponding 2D pixel coordinates $\mathbf{S}_{k_1} = (x_{k_1}, y_{k_1})$, $\mathbf{S}_{k_2} = (x_{k_2}, y_{k_2})$, for an image point $\mathbf{p}$, the depth value of it is defined as

$$\mathbf{D}_l^{Limb}(\mathbf{p}) = \begin{cases} \frac{\mathbf{u}\cdot\hat{\mathbf{v}}}{\mathbf{v}}\Delta Z_{k_1,k_2} + Z_{k_1} & \text{if } \mathbf{p} \text{ on limb } L_{k_1,k_2} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

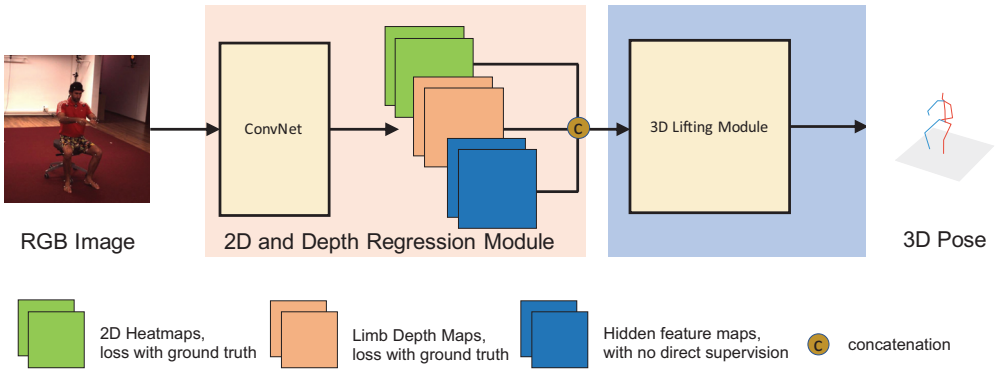

Figure 2: Overall pipeline of the proposed method. 2D heatmaps, Limb Depth Maps are first generated and supervised, along with hidden features. Then the 3D Lifting module takes 2D heatmaps, Limb depth maps, as well as hidden feature maps as inputs, and outputs the predicted 3D pose. State-of-the-art 2D pose estimation networks are adapted as 2D and Depth Regression Module. 3D Lifting Module consists of a sequence of convolutional layers, followed by an average pooling layer and the final linear layer for outputs.

where $\mathbf{u} = \mathbf{p} - \mathbf{p}_{k_1}$ is vector from $\mathbf{p}_{k_1}$ to point $\mathbf{p}$, $\mathbf{v} = \mathbf{p}_{k_2} - \mathbf{p}_{k_1}$ is the vector from point $\mathbf{p}_{k_1}$ to $\mathbf{p}_{k_2}$, and $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|_2$ is the unit vector, $\Delta Z_{k_1,k_2} = Z_{k_2} - Z_{k_1}$ is depth value difference of joints k_1, k_2. A point is defined as on the limb when its distance to the limb is within a certain threshold. $\mathcal{D}^{Limb} = \{\mathbf{D}_l^{Limb}\}_{l=1}^{L}$ is the Limb Depth Maps for in total L limbs. Figure 1 shows an example of Limb Depth Map. We generate depth maps for each limb, which could avoid conflicts when joints or limbs are overlapping to each other. Densely-generated depth maps provide dense supervisions which are aligned with image cues. In some bad poses (e.g. occluded), the depth value of joints might not be easily inferred, while other points on the limb might be in good shape to infer with, and then propagate through the limb. Note that for the generated depth maps, the depth values of areas that do not belong to any limbs are actually unknown. Here we define their values as zero, and we ignore losses in these areas during training.

Explicit Depth Values for 3D Pose Estimation

In this section, we describe why and how depth values are explicitly used in the lifting process from 2D pixel coordinates to 3D camera coordinates.

Camera projection model First we review how points in camera coordinates are projected to pixel coordinates. For any point (X, Y, Z) in camera coordinates, the corresponding pixel coordinates (x, y) could be obtained using camera parameters by assuming a perspective projection model:

$$[x, y, 1]^\intercal = \mathbf{K}[X, Y, Z, 1]^\intercal / Z_c, \tag{3}$$

$$\mathbf{K} = \begin{bmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where $\mathbf{K}$ is the camera calibration matrix, α_x, α_y, p_x, p_y are camera parameters.

Inversely, the camera coordinates (X, Y) could be computed as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} (x - p_x)Z/\alpha_x \\ (y - p_y)Z/\alpha_y \end{bmatrix}. \tag{5}$$

Explicit depth value for 2D to 3D conversion We consider the lifting process of converting 2D joints from pixel coordinates to 3D coordinates in camera space. As we could see from Eq. 5, converting pixel coordinates to camera coordinates for X and Y axes would need the value of Z axis (depth) first. Previous methods typically estimate (X, Y, Z) simultaneously, which makes the learning process nested. Thus, we propose to estimate depth values for each joint first, as well as the 2D pixel coordinates. Then lifting to 3D camera coordinates via learned transformation taking into account both the 2D pixel coordinates and depth values. The transformation process could be modeled as

$$\mathcal{J} = f(g(\mathcal{S}), h(\mathbf{Z}), N(\mathbf{I})), \tag{6}$$

where $\mathcal{J} = \{\mathbf{J}_i\}_{i=1}^{K}$ is the camera coordinates of human joints, $\mathcal{S} = \{\mathbf{S}_i\}_{i=1}^{K}$ is the pixel coordinates of joints, $g(\mathcal{S})$ is function of 2D pixel coordinates, e.g. $g(\mathbf{S}_k) = \mathcal{H}(\mathbf{S}_k)$ is 2D heatmap function. $h(\mathbf{Z})$ is function of depth values of joints, e.g. $h(\mathbf{Z}) = \mathcal{D}$ is the depth maps function. $N(\mathbf{I})$ is function of images, which indicates the hidden feature maps of feature extraction networks, other than 2D heatmaps and depth maps. f represents the lifting module, a general transformation function, which could be modeled as fully-connected layers or convolutional neural networks with non-linear activations.

Lifting module from 2D to 3D The lifting function f in Eq. 6 could be modeled as various networks. In this work, we instantiate f as convolutional neural networks. More specifically, the module takes into previous feature maps containing 2D heatmaps and depth maps, as well as the rest hidden feature maps. Three convolutional layers of kernel size 3 x 3, stride 2 are used to downsample the resolution. Then one residual block (He et al. 2016) is used to upsample the channel dimension. A sequence of five bottleneck

blocks (He et al. 2016) are used to increase the receptive fields, then an average pooling and the final fully-connected layer are used to output the targets. We choose to use convolutional layers instead of fully-connected layers in (Martinez et al. 2017) for fewer parameters and higher speed.

Experiments

We evaluate our method and perform ablation study on the large-scale Human3.6M dataset (Ionescu et al. 2014), and show generalization ability on the in-the-wild MPI-INF-3DHP benchmark (Mehta et al. 2017a) and 2D MPII (Andriluka et al. 2014) test set. We solve the problem of estimation 3D pose in the camera space, more specifically, pevis-centered camera coordinates.

Datasets and evaluation metrics

Human3.6M is currently the largest 3D human pose benchmark, which contains 3.6 million video frames for in total 11 subjects. Accurate 3D annotations are provided, as well as the camera parameters. Following previous works, we split the dataset into training set of five subjects (S1, S2, S3, S4, S5) and test set of two subjects (S9, S11). Every 5 frames of the training set are used for training, while we test on every 64 frames of test set following (Sun et al. 2018).

We consider the evaluation metrics mean per-joint position error (MPJPE), and also reports the error after rigid alignment, denoted as P-MPJPE. Specially, we use two additional evaluation metrics in our ablation study to better analyze the behavior, which are denoted as MPJPE (X, Y) and MPJPE (Z), considering (X, Y) axes and Z axes separately. For one frame, the error metrics are calculated as follows.

$$\text{MPJPE} = \frac{1}{K}\sum_{k=1}^{K} \|(X_k, Y_k, Z_k)_{pred} - (X_k, Y_k, Z_k)_{gt}\|_2$$

$$\text{MPJPE (X, Y)} = \frac{1}{K}\sum_{k=1}^{K} \|(X_k, Y_k)_{pred} - (X_k, Y_k)_{gt}\|_2$$

$$\text{MPJPE (Z)} = \frac{1}{K}\sum_{k=1}^{K} \|(Z_k)_{pred} - (Z_k)_{gt}\|_2$$

K is the number of joints, X_k, Y_k, Z_k is the camera coordinates of joint k. For a set of frames, the error is the average over MPJPE of all frames.

MPI-INF-3DHP dataset (Mehta et al. 2017a) is a recently released 3D human pose dataset, which contains in the wild images with general backgrounds, both indoors and outdoors. It is captured with marker-less motion capture, allowing for wearing everyday apparel. It has 8 subjects performing 8 activity sets, covering more pose classes than Human3.6m (Mehta et al. 2017a). We use the original images and do not add augmented ones used in (Mehta et al. 2017a). The percentage of correct 3D Keypoints (3D PCK) and the Area under curve (AUC) as in (Mehta et al. 2017a) are used as evaluation metrics for this benchmark.

Implementation details

2D pose estimation module We use two state-of-the-art 2D human pose estimation modules as the feature extractor networks for our methods. SimpleBaselines (Xiao, Wu, and Wei 2018) and HRNet (Sun et al. 2019) are used. We conducted ablation study using SimpleBaselines with ResNet-50 as backbone, and report final results using HRNet-W32.

Training details We train all the methods for 20 epochs using Adam optimizer, with of initial learning rate of 0.001, and decreases 10 times at the 15th, 17th epochs. Rotation, synthetic occlusion (Sárándi et al. 2018) and Photo metric distortion are used as data augmentation. The input size of images is 256 x 256. The training targets are the pelvis-centered camera coordinates of human joints. Our methods could benefit from the large-scale 2D datasets, we choose to add MPII dataset into the training set. For methods trained with both the 3D datasets (Human3.6m or MPI-INF-3DHP) and 2D MPII dataset, the sampling ratio for the two sources is 1:1. Only 2D heatmaps loss is added when training on 2D annotated datasets.

Ablation study

In this section, we study various components of our methods and show the effectiveness of the proposed depth maps representation and the lifting module with explicit depth values.

Target types	Joint Depth Maps	Limb Depth Maps
MPJPE (Z)	37.9	**36.3**$_{\downarrow 1.6}$
MPJPE	50.2	**49.3**$_{\downarrow 0.9}$
P-MPJPE	41.0	**39.3**$_{\downarrow 1.7}$

Table 1: Error on Human3.6M test set using Joint Depth Maps or Limb Depth Maps as regression targets. Error metrics MPJPE (Z), MPJPE and P-MPJPE are shown in table, the lower the better. Method trained with Limb Depth Maps shows lower error on the depth dimension MPJPE (Z), and is overall better than method using Joint Depth Maps.

Effectiveness of depth map We show that our densely-generated limb depth maps helps the learning of the Z values. For estimating depth values, we experimented with joint depth maps and the limb depth maps representation as described in Eq. 1 and Eq. 2. We only change the depth maps supervision targets in the pipeline, while keeping others the same to study the difference.

To better analyze the learning process of depth values alone, we decouple the learning of (X, Y) and Z by recovering (X, Y) using estimated (x, y) from learned 2D heatmaps and camera parameters, reducing the possible influence of depth values when estimating (X, Y). Table 1 shows the results on Human3.6m validation set. Method using limb depth maps as target has 0.9 mm MPJPE and 1.7 mm P-MPJPE lower error that method using joint depth maps. Specially, limb depth maps method has 1.6 mm MPJPE (Z) lower error that using joint depth maps, which shows the effectiveness of our proposed densely-generated limb depth maps for inferring the depth values. The improvement could

come from densely supervision signal compared to joint depth maps which only have one non-zero value per joint depth map. It could also benefit from situations that joints are in bad cases (e.g. occluded) that other points on the limb have better (image) clues to infer the depth values.

Method	(a)	(b)	(c)
2D Heatmaps		✓	✓
Limb Depth Maps			✓
MPJPE (X, Y)	30.1	$27.7_{\downarrow 2.4}$	$\mathbf{25.9}_{\downarrow \mathbf{4.2}}$
MPJPE (Z)	37.5	$37.9_{\uparrow 0.4}$	$\mathbf{36.3}_{\downarrow \mathbf{1.2}}$
MPJPE	52.9	$51.7_{\downarrow 1.2}$	$\mathbf{48.8}_{\downarrow \mathbf{4.1}}$

Table 2: Error on Human3.6M test set for methods w/o 2D heatmaps or limb depth maps loss as supervision. Best results are shown in bold. The relative gains (drops) are compared to the baseline are shown in the subscript. The lower the better for MPJPE metrics. Adding limb depth maps as supervision clearly outperforms other methods and shows much lower error for (X, Y) axes.

Effectiveness of explicit depth for 3D pose estimation We show that explicitly introducing depth values contributes positively for 3D lifting process from 2D pixel coordinates. We conducted experiments by taking out the 2D heatmaps regression or limb depth maps regression components. More specifically, we choose to add or not 2D heatmaps loss or limb depth maps loss to the input of the lifting module, while keeping other the same. To better analyze the learning process, we additionally report the evaluation metric MPJPE (X, Y) and MPJPE (Z), which consider the error for (X, Y) axes and Z separately. The results are shown in Table 2.

Method (a) shows the performance of baseline direct regression method without any intermediate supervision. Method (b) adds 2D heatmaps regression supervision to the intermediate feature maps. By adding 2D heatmaps supervision, Method (b) outperform method (a) by 2.4 mm MPJPE (X, Y), which benefits from better localization of 2D joints. Method (c) is our proposed one, which takes both 2D pixel coordinates and depth values for consideration by adding 2D heatmaps and limb depth maps supervision to the intermediate feature maps. We could see that method (c) has 4.2 mm lower MPJPE (X, Y) error than method (a), and 1.8 mm lower MPJPE (X, Y) than method (b). Method (c) has 4.2 mm lower error in the X and Y axes, while 1.2 mm in the Z axis, compared to method (a). The major improvement of method (c) compared to others comes from more accurate localization of X, Y axes. This shows the effectiveness of explicitly introducing depth values into the learning process of (X, Y), which coincides with Eq. 6.

Comparison with state-of-the-art methods

We compare our method with current state-of-the-art methods. Results are shown in Table 3a and Table 3b. We could see that our method outperforms previous ones by a large margin under all settings. Equipped with 2D MPII dataset, and HRNet-W32 as feature extraction network, our method achieves the best result of 44.6 mm MPJPE, 36.3 mm P-MPJPE, which sets the new state-of-the-art performance. Note that the pipeline of our method is relatively simple compared to many previous state-of-the-art ones. Bone constrains (Zhou et al. 2017), 3D to 2D reconstructions (Habibie et al. 2019) and are used in (Habibie et al. 2019), our method is able to outperform it by a large margin. Also, our method manages to achieve on better performance than multi frame method (Pavllo et al. 2019), which utilizes temporal convolution and has a receptive field of 243 frames, while ours only look at single frame. Our method can easily plug into these techniques, which could further improve upon our methods.

Evaluation on MPI-INF-3DHP

We also test our method on MPI-INF-3DHP benchmark (Mehta et al. 2017a), which has more in-the-wild elements compared to Human3.6M. Following previous The results are shown in Table 4. We could see that our method achieves the new state-of-the-art performance under any evaluation metrics, which shows our method could generalize to in-the-wild situations. Our method is able to outperform (Habibie et al. 2019) by 1.9 3DPCK and 4.9 AUC, which is trained on the hybrid of MPI-INF-3DHP, Human3.6M 3D datasets, MPII and LSP 2D datasets, while ours is trained on MPI-INF-3DHP train set alone.

Visualization results on in-the-wild images

We test our method on in-the-wild images, where the 3D ground truth labels are not available. Figure 3 shows some visualization results on MPII test set. We could see that our method is able to produce reasonable 3D pose results on these images, where the backgrounds are much more complicated than current available 3D pose datasets. This indicates our method generalizes well for in the wild situations.

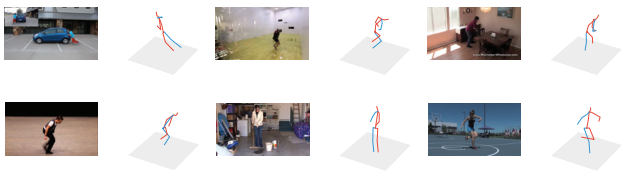

Figure 3: Visualization results of our method on MPII test set. Left: images. Right: our 3D pose predictions. Our method generalizes well on 2D in-the-wild datasets, even the 3D ground truth labels are not available.

Conclusion

In this work, we propose a framework for 3D human pose estimation by tackling the depth values representation and the lifting process from pixel coordinates to camera coordinates. We show the effectiveness of our proposed depth representation and lifting process. Our proposed framework achieves the state-of-the-art performance on two large-scale 3D human pose datasets, while keeping a clean pipeline.

	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
(Pavlakos et al. 2017)	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
(Tekin et al. 2017)	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6	70.1	107.3	69.3	70.3	74.3	51.8	63.2	69.7
(Habibie et al. 2019)	54.0	65.1	58.5	62.9	67.9	75.0	54.0	60.6	82.7	98.2	63.3	61.2	66.9	50.0	56.5	65.7
(Martinez et al. 2017)	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
(Sun et al. 2017)	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
(Fang et al. 2018)	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
(Pavlakos, Zhou, and Daniilidis 2018)	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
(Yang et al. 2018)	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
(Luvizon, Picard, and Tabia 2018)	49.2	51.6	47.6	50.5	51.8	60.3	48.5	51.7	61.5	70.9	53.7	48.9	57.9	44.4	48.9	53.2
(Pavllo et al. 2019) (single-frame)	47.1	50.6	49.0	51.8	53.6	61.4	49.4	47.4	59.3	67.4	52.4	49.5	55.3	39.5	42.7	51.8
(Lee, Lee, and Lee 2018) (†)	40.2	49.2	47.8	52.6	50.1	75.0	50.2	43.0	55.8	73.9	54.1	55.6	58.2	43.3	43.3	52.8
(Rayat Imtiaz Hossain and Little 2018) (†)	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
(Pavllo et al. 2019) (243 frames)(†)	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	**44.0**	49.0	32.8	33.9	46.8
Ours (ResNet-50)	36.9	43.9	39.5	60.4	45.3	51.6	38.1	41.9	54.1	79.9	44.4	57.6	45.2	32.2	36.5	47.3
Ours (HRNet-W32)	**34.9**	**40.8**	**37.5**	**47.2**	**41.5**	**46.6**	**35.9**	**39.5**	**52.6**	**72.5**	**42.3**	45.8	**42.0**	**31.6**	**33.8**	**43.2**

(a) Reconstruction error (MPJPE).

	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
(Habibie et al. 2019)	43.7	46.9	45.4	48.0	50.2	40.6	41.6	60.7	75.6	48.8	54.9	46.8	47.5	36.9	43.9	49.2
(Martinez et al. 2017)	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
(Sun et al. 2017)	42.1	44.3	45.0	45.4	51.5	53.0	43.2	41.3	59.3	73.3	51.0	44.0	48.0	38.3	44.8	48.3
(Fang et al. 2018)	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
(Pavlakos, Zhou, and Daniilidis 2018)	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
(Yang et al. 2018)	26.9	30.9	36.3	39.9	43.9	47.4	28.8	29.4	36.9	58.4	41.5	30.5	29.5	42.5	32.2	37.7
(Pavllo et al. 2019) (single-frame)	36.0	38.7	38.0	41.7	40.1	45.9	37.1	35.4	46.8	53.4	41.4	36.9	43.1	30.3	34.8	40.0
(Rayat Imtiaz Hossain and Little 2018) (†)	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
(Pavllo et al. 2019) (243 frames) (†)	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	**33.8**	37.8	25.6	27.3	36.5
Ours (ResNet-50)	32.1	36.2	33.9	41.2	37.4	40.6	30.7	33.4	45.0	55.0	37.4	38.8	36.8	25.7	30.6	37.3
Ours (HRNet-W32)	**29.9**	**33.6**	**31.4**	**37.1**	**33.9**	**36.8**	**28.4**	**30.7**	**42.6**	**52.2**	**35.3**	35.2	**34.0**	**24.9**	**27.9**	**34.6**

(b) Reconstruction error after rigid alignment with the ground truth (P-MPJPE).

Table 3: Reconstruction error on Human3.6M. (†) indicates use of multi-frame temporal information. Lower is better, best in bold. Some results borrowed from (Pavllo et al. 2019). Our methods outperform all single-frame methods, as well as multi-frame methods.

Method	3DPCK	AUC
(Mehta et al. 2017a)	76.5	40.8
(Mehta et al. 2017b)	76.6	40.4
(Rogez, Weinzaepfel, and Schmid 2017)	59.6	27.6
(Zhou et al. 2017)	69.2	32.5
(Mehta et al. 2018)	75.2	37.8
(Luo, Chu, and Yuille 2018)	81.8	45.2
(Kanazawa et al. 2018)	77.1	40.7
(Yang et al. 2018)	69.0	32.0
(Wandt and Rosenhahn 2019)	82.5	58.5
(Habibie et al. 2019)	91.3	57.5
Ours	**93.2**	**62.4**

Table 4: Results for the MPI-INF-3DHP test set. A higher value is better for 3D PCK and AUC. The best results are marked in bold. Some results borrowed from (Wandt and Rosenhahn 2019). Ours are trained on MPI-INF-3DHP only.

References

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B. 2014. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7291–7299.

Chen, C.-H., and Ramanan, D. 2017. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7035–7043.

Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; and Sun, J. 2018. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7103–7112.

Fang, H.-S.; Xu, Y.; Wang, W.; Liu, X.; and Zhu, S.-C. 2018. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Habibie, I.; Xu, W.; Mehta, D.; Pons-Moll, G.; and Theobalt, C. 2019. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10905–10914.

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

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2014. 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.

Kanazawa, A.; Black, M. J.; Jacobs, D. W.; and Malik, J. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7122–7131.

Lee, K.; Lee, I.; and Lee, S. 2018. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 119–135.

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

Luo, C.; Chu, X.; and Yuille, A. 2018. Orinet: A fully convolutional network for 3d human pose estimation. *arXiv preprint arXiv:1811.04989*.

Luvizon, D. C.; Picard, D.; and Tabia, H. 2018. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5137–5146.

Martinez, J.; Hossain, R.; Romero, J.; and Little, J. J. 2017. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2640–2649.

Mehta, D.; Rhodin, H.; Casas, D.; Fua, P.; Sotnychenko, O.; Xu, W.; and Theobalt, C. 2017a. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 International Conference on 3D Vision (3DV)*, 506–516. IEEE.

Mehta, D.; Sridhar, S.; Sotnychenko, O.; Rhodin, H.; Shafiei, M.; Seidel, H.-P.; Xu, W.; Casas, D.; and Theobalt, C. 2017b. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)* 36(4):44.

Mehta, D.; Sotnychenko, O.; Mueller, F.; Xu, W.; Sridhar, S.; Pons-Moll, G.; and Theobalt, C. 2018. Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, 120–130. IEEE.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, 483–499. Springer.

Pavlakos, G.; Zhou, X.; Derpanis, K. G.; and Daniilidis, K. 2017. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7025–7034.

Pavlakos, G.; Zhou, X.; and Daniilidis, K. 2018. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7307–7316.

Pavllo, D.; Feichtenhofer, C.; Grangier, D.; and Auli, M. 2019. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7753–7762.

Rayat Imtiaz Hossain, M., and Little, J. J. 2018. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 68–84.

Rogez, G.; Weinzaepfel, P.; and Schmid, C. 2017. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3433–3441.

Sárándi, I.; Linder, T.; Arras, K. O.; and Leibe, B. 2018. Synthetic occlusion augmentation with volumetric heatmaps for the 2018 eccv posetrack challenge on 3d human pose estimation. *arXiv preprint arXiv:1809.04987*.

Sun, X.; Shang, J.; Liang, S.; and Wei, Y. 2017. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, 2602–2611.

Sun, X.; Xiao, B.; Wei, F.; Liang, S.; and Wei, Y. 2018. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 529–545.

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

Tekin, B.; Katircioglu, I.; Salzmann, M.; Lepetit, V.; and Fua, P. 2016. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*.

Tekin, B.; Márquez-Neila, P.; Salzmann, M.; and Fua, P. 2017. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3941–3950.

Wandt, B., and Rosenhahn, B. 2019. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7782–7791.

Wei, S.-E.; Ramakrishna, V.; Kanade, T.; and Sheikh, Y. 2016. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4724–4732.

Xiao, B.; Wu, H.; and Wei, Y. 2018. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 466–481.

Yang, W.; Ouyang, W.; Wang, X.; Ren, J.; Li, H.; and Wang, X. 2018. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5255–5264.

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

Zhou, X.; Huang, Q.; Sun, X.; Xue, X.; and Wei, Y. 2017. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, 398–407.