This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Not All Parts Are Created Equal: 3D Pose Estimation by Modeling Bi-directional Dependencies of Body Parts

Jue Wang[1,2] Shaoli Huang*[1] Xinchao Wang[3] Dacheng Tao[1]

[1]UBTECH Sydney AI Centre, School of Computer Science, FEIT, University of Sydney, Darlington, NSW 2008, Australia

[2]University of Technology Sydney [3]Stevens Institute of Technology

jue.wang.0911@gmail.com, {shaoli.huang, dacheng.tao}@sydney.edu.au, xinchao.wang@stevens.edu

Abstract

Not all the human body parts have the same degree of freedom (DOF) due to the physiological structure. For example, the limbs may move more flexibly and freely than the torso does. Most of the existing 3D pose estimation methods, despite the very promising results achieved, treat the body joints equally and consequently often lead to larger reconstruction errors on the limbs. In this paper, we propose a progressive approach that explicitly accounts for the distinct DOFs among the body parts. We model parts with higher DOFs like the elbows, as dependent components of the corresponding parts with lower DOFs like the torso, of which the 3D locations can be more reliably estimated. Meanwhile, the high-DOF parts may in turn impose a constraint on where the low-DOF ones lie. As a result, parts with different DOFs supervise one another, yielding physically constrained and plausible pose-estimation results. To further facilitate the prediction of the high-DOF parts, we introduce a pose-attribute estimation, where the relative location of a limb joint with respect to the torso, which has the least DOF of a human body, is explicitly estimated and further fed to the joint-estimation module. The proposed approach achieves very promising results, outperforming the state of the art on several benchmarks.

1. Introduction

The unique physiological structure of a human body results in that different body parts may have different degrees of freedom (DOFs). For example, the motion range of a human wrist is significantly broader than that of a shoulder. Such distinct DOFs further lead to the varying levels of difficulties when it comes to 3D pose estimation, for which the goal is to predict the 3D locations of human body joints from one or multiple images.

Most of the existing 3D pose estimation methods [48, 47, 49, 14, 17, 7, 22, 33, 45, 21, 5, 31], despite the impressive state-of-the-art performances achieved, have overlooked such DOF distinctions among the body parts and have treated them equally during the learning process. This consequently leads to the often larger reconstruction errors on the more flexible and thus more challenging body parts, such as the limbs.

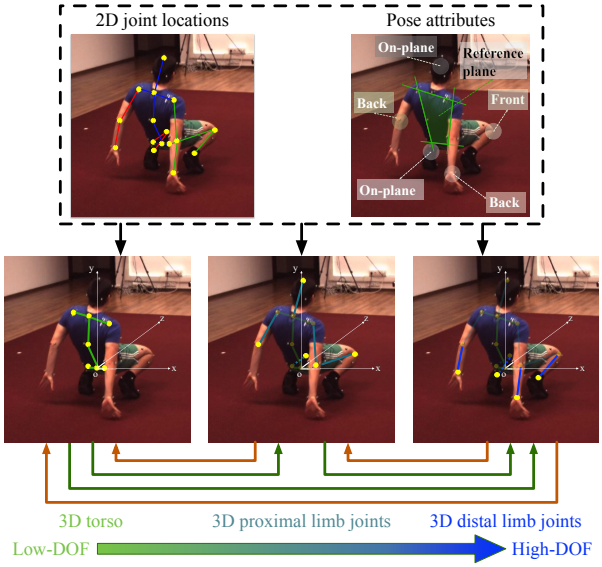

Figure 1. Illustration of the proposed approach. The 2D joint locations along with the pose attributes are first estimated using a multi-task network, and then fed as input to the 3D pose estimation network that explicitly models the bi-directional dependencies among body parts of different DOFs. Specifically, the high-DOF body parts are treated as dependent components of low-DOF ones, and in turn, provide a constraint on where the low-DOF ones lie.

In this paper, we propose a dedicated approach that explicitly utilize the DOF differences among body parts to facilitate 3D pose estimation (see Fig. 1). We categorize the body parts into three groups based on the increasing levels of DOFs: the *torso*, the *proximal limb joints* including the head, the elbows and the knees, and the *distal limb joints* including the wrists and ankles. By such categorization, we

*Corresponding author

explicitly model the location of a higher-DOF joint like an elbow, as a dependent variable of a lower-DOF one like the torso in a progressive manner, where the latter can be in most cases estimated more reliably. In turn, the locations of the higher-DOF joints may constrain those of lower-DOF ones and potentially revise the prediction errors. Such bi-directional and progressive dependencies enable that body parts of various DOFs supervise one another, yielding physically plausible 3D pose-estimation results.

To further take advantage of image evidences for the dependency modeling, we introduce *pose attribute*, which captures the relative location of a limb joint with respect to the torso, the body part with the least DOF. Each limb joint may be assigned one of three pose attributes, *front*, *back*, and *on-plane*, describing its offset to the torso. Given an input image, these attributes are estimated together with the 2D pose via a multi-task network, and are further fed to the succeeding 3D pose module that explicitly accounts for the aforementioned progressive dependencies. In other words, the estimated pose attributes, being an input to 3D pose estimation, provide an explicit and strong prior of where the limb may lie, benefiting the downstream part-dependency modeling. Unlike the regression-based depth estimation that often yields deviations, which may further propagate to the 3D pose estimation and deteriorate the results, the three-class pose-attribute prediction, as demonstrated in our experiments, turns out to be in most cases reliable, providing advantageous image cues.

Our main contributions are thus summarized as follows.

- By categorizing human body parts into three varying levels of DOFs, we explicitly model the bi-directional dependencies among the body parts, which supervise one another and together yield the physically constrained and plausible 3D pose estimation.
- We introduce for each limb joint a pose attribute, depicting the offset of the joint from the torso. Estimated together with the 2D pose via a multi-task network, the pose attribute provides an explicit prior of the joint's location and further facilitates the succeeding 3D pose estimation.

We test our approach on benchmarks including Human3.6M [11] and MPI-INF-3DHP [15], and demonstrate that it consistently achieves very encouraging results, outperforming the state of the art. Furthermore, we show that even in the absence of 3D annotations and pose-attribute ground truths, by adopting an unsupervised domain adaptation approach, our method can be readily applied to in-the-wild images and achieve promising performances.

2. Related work

We briefly review here the two main streams of 3D pose estimation methods, the one-stage approaches and the two-stage ones, and then look at the methods that explicitly utilize additional image cues. Finally we outline the differences of the proposed method with respect to the prior ones.

One-stage approach. One-stage approaches directly infer 3D human poses from input images. Tekin *et al*. [32] trained an auto-encoder to learn a latent pose representation and joint dependencies in a high-dimensional space, then adopted the decoder at the end of the convolutional neural network to infer 3D poses. Pavlakos *et al*. [22] proposed a volumetric representation for 3D joints and used a coarse-to-fine strategy to refine the prediction iteratively. Rogez *et al*. [27, 28] used ConvNets to classify each image in the appropriate pose class. Nie *et al*. [19] proposed to predict the depth on joints from global and local image features.

All the above methods require images with corresponding 3D ground truths. Due to the lack of in-the-wild images with 3D annotations, these approaches tend to produce unsatisfactory results on inputs with domain shifts. To this end, Zhou *et al*. [45] proposed a weakly-supervised approach to utilize the large-scale in-the-wild 2D pose data. Dabral *et al*. [5] improved this weakly-supervised setup by using two additional losses to restrict the predicted 3D pose structure. Yang *et al*. [41] considered the 3D pose estimator as a generator and used an adversarial learning approach to generate indistinguishable 3D poses. Sun *et al*. [31] used soft argmax to regress 2D/3D poses directly from images. Despite the success of this strategy, a main flaw of these methods lies in that they tend to fail when the height of the subject is considerably different from those in the training set, since they fixed the scale of 3D poses to construct 3D poses from 2D poses and depths.

Two-stage approach. Another widely used strategy is to divide the 3D pose estimation task into two decoupled subtasks: 2D pose detection, followed by 3D pose inference from 2D poses. These methods comprise a 2D pose detector and a subsequent optimization [48, 47, 49] or regression [4, 3, 17, 30, 36, 14, 19, 7, 12] step to estimate 3D pose . In these methods, the 2D pose and 3D pose estimation stages are separated, making these 3D pose estimators generalize well on outdoor images. The most straightforward approach is to represent 3D poses as linear combinations of models learned from training data [48, 47, 49]. This method is based on dictionary learning and has to run an optimization for each example, making it very time-consuming in both training and evaluation. Specifically, Chen *et al*. [4] and Yasin *et al*. [42] used a pose library to retrieve the nearest 3D pose given the corresponding 2D pose prediction.

Recently, with the availability of large-scale 3D pose datasets, deep-learning based 2D-to-3D pose regression methods have made significant progress. For instance, Moreno-Noguer [17] used an hourglass network to regress the 3D joints distance matrices instead of 3D poses because

they found that the distance matrix representation shows a more correlated pattern than Cartesian ones and suffer from smaller ambiguities. Notably, Martinez *et al*. [14] achieved state-of-the-art results using a simple multi-layer perceptron with residual blocks [9] to regress 3D poses directly from 2D poses. Sun *et al*. [30] re-parameterized the pose presentation to use bones instead of joints and proposed a structure-aware loss. Lee *et al*. proposed a long short-term memory (LSTM) architecture to reconstruct 3D depth from the centroid to edge joints through learning the joint interdependencies. However, as 2D-to-3D mapping is an ill-posed problem, methods along this line are prone to ambiguities in the 2D-to-3D regression at the second stage of this pipeline, if no addition image cues are utilized.

Additional image cues. The development in computer vision domain makes it possible to learn various image cues from images [37, 38, 44, 24]. The idea of pose attributes was firstly explored by Pons-Moll *et al*. [23]. In their work, they proposed an extensive set of posebits representing the boolean geometric relationships between body parts, and designed an algorithm to select useful posebits for 3D pose inference. Recently, many researchers have investigated approaches that combine 2D pose detection techniques and the power of CNN to extract supplementary information from images to enhance 3D pose estimation. Tekin *et al*. [33] proposed a two-stream network with trainable fusion to fuse 2D heat maps and image features to obtain the final 3D pose estimation. Pavlakos *et al*. [21] augmented the LSP and MPII 2D pose datasets with ordinal depth annotations, which are used as weak supervision to learn the depth of each joint. Zhou *et al*. [45] used a CNN to predict 2D joint locations and the corresponding depth, then rescaled the predictions to a pre-defined canonical skeleton. All these approaches tried to learn depth information from single images. However, an image is a two-dimensional representation itself and does not carry depth information, making it challenging to learn depth from images. Also, depth is highly sensitive to camera parameters, such as translation and rotation, making the depth prediction of human joints more difficult.

Our approach. By explicitly categorizing body parts into varying levels of DOFs, which have been largely overlooked in prior methods, the proposed approach treats the higher-DOF parts as dependent components of the lower-DOF ones, and conversely, constrains the latter using the former. Such bi-directional 3D dependency modeling is further facilitated by a dedicated and newly-introduced pose attribute estimation, which predicts the relative location of a limb joint with respect to the torso.

3. Method

Different body parts have varying levels of DOFs due to the unique physiological structure of a human body. To see this difference, we use the ground truths of the Human3.6M dataset [11], to compute, for each joint location, its standard deviation (STD), which gives us a coarse description on the motion range of the joint. We show the results in Tab. 1, where, as expected, the distal limb joints including wrist and ankle have the largest STDs, followed by the proximal limb joints including elbow and knee. The joints on the torso, like spine and hip, yield the smallest STDs.

Such DOF differences among body joints lead to the different levels of challenges in terms of pose estimation, and further result in estimation results of diverse qualities, especially those obtained by conventional methods that treat all the parts equally. For example, as shown in Tab. 6, the method of [14] produces more accurate predictions for joints on the torso and proximal joints on limbs but poorer ones for distal joints.

To this end, we categorize the body joints into three levels of DOFs, from low to high: *torso*, *proximal limb joints*, and *distal limb joints*. We then explicitly model the higher-DOF joints as dependent components of the low-DOF ones that are easier to estimate, and in turn, enforce the former to impose physical constraints on the latter. To aid the learning of this bi-directional dependency, we introduce *pose attribute* to measure the relative location of a limb joint with respect to the torso, the body part that can be in most cases reliably estimated. Unlike the regression-based depth estimation that is often prone to deviations, the proposed pose attribute estimation is taken to be a much less demanding classification problem, where one of the only three labels, *front*, *back*, and *on-plane*, is assigned to each limb joint.

More specifically, our 3D pose estimation follows a two-step strategy, as depicted in Fig. 2. In the first step, we adopt a multi-task network to estimate the 2D pose and the proposed pose attribute, both of which are together fed into another network to model the bi-directional dependency for 3D pose estimation in the second step. These two networks are connected via soft argmax layers [43, 35, 13, 31], so that the network training is end-to-end. In what follows, we provide more details on the two networks.

3.1. Multi-task network

The multi-task network, as discussed, handles simultaneously 2D pose estimation and pose attribute learning. In recent years, many network architectures have been proposed for 2D pose estimation and have achieved encouraging results [39, 18, 10, 40, 31]. Here, we adopt the state-of-the-art

Joint	**Hip**	**Spine**	**Thorax**	**Shoulder**	**Head**
STD (mm)	68.5	57.8	109	127	140
Joint	**Elbow**	**Knee**	**Wrist**	**Ankle**	**Avg.**
STD (mm)	195	188	240	227	150

Table 1. The standard deviation of the 3D locations of each joint, obtained using the ground-truth annotations of the Human3.6M training set.

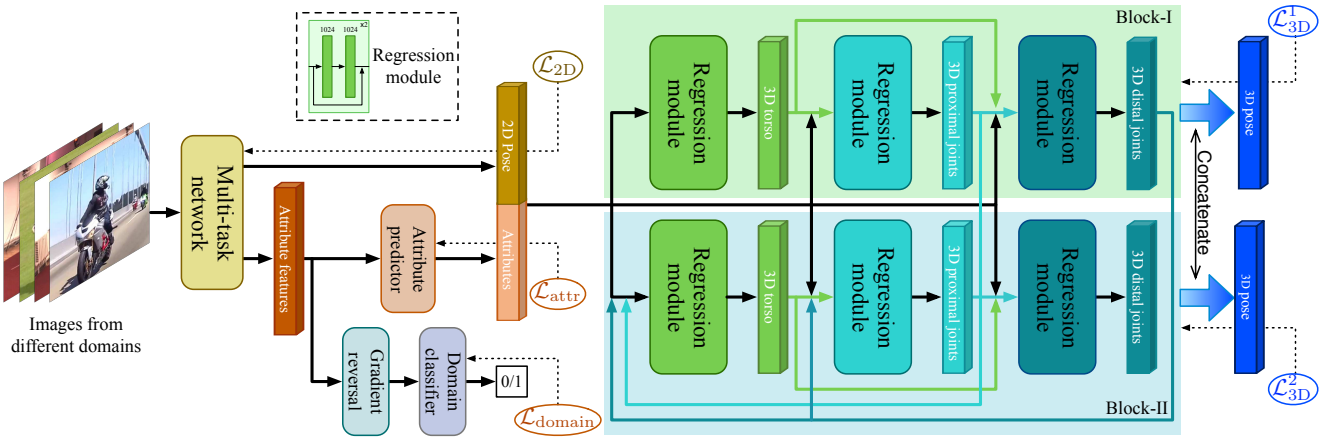

Figure 2. The network architecture of our method. It consists of two parts, a multi-task network that learns 2D poses and attributes from images and a progressive 3D pose estimation network. The multi-task network is trained on the mixture of MPII and Human3.6M datasets. As there are no 3D annotations available in MPII, we adopt an unsupervised domain adaptation method [8] to help the network learn domain-independent features for attribute prediction, so that the network can predict reasonable attributes for in-the-wild images in the absence of attribute supervision (see Section 3.1.2 for more detail). The 3D pose network takes as input the estimated 2D poses and pose attributes, and explicitly models the bi-directional dependencies among the three groups of body parts of different DOFs. The final 3D pose estimation is the concatenation of the three group predictions.

stacked hourglass backbone, as done in many other methods [14, 45, 5, 7, 21, 41], to be our multi-task architecture, following the network design proposed by Zhou *et al*. [45]. As the pose attributes are highly related to the locations of joints, and a pretrained 2D pose detector can act as a reliable joint feature extractor, it is a natural idea to reuse the feature maps in the 2D pose detector to ease pose attribute learning.

3.1.1 2D pose detection

Many previous 3D pose estimation methods [14, 7, 12] fine tune a pretrained 2D pose detector on the 3D pose dataset like Human3.6M to obtain the pose estimation results. Since the images in 3D pose datasets are captured in an indoor lab environment with several subjects, the diversity of image backgrounds, clothing, skin color and so on, is very limited compared to the 2D pose datasets. As a result, the generalization ability of the model could deteriorate after fine-tuning, limiting the application of the pose detector on real-world images.

Here, we train our 2D pose detector from scratch using a mixture of images from both 2D pose and 3D pose datasets. In each training batch, half of the examples are randomly sampled from a 2D pose training dataset, and the other from a 3D one. Through this strategy, the 2D pose detector could achieves high performance on both of the 2D and 3D pose dataset. In other words, the 2D pose detector has good generalization capability. Moreover, the mixed training strategy also helps to learn pose attributes, as shown in the experiments (see Tab. 4), possibly due to the mixed trained network could learn better human keypoint features. The mixed training strategy is essential in our method. On one hand, it helps to train a 2D pose detector with good generalization ability. On the other hand, it is also beneficial to the training of the pose attribute learning sub-network by providing better image features.

Let us use M_n to denote the ground-truth 2D pose heat map of joint n, and use $\hat{M}_n$ to denote that of the prediction. The loss function for 2D pose detection is taken to be

$$\mathcal{L}_{\text{2D}} = \frac{1}{N}\sum_{n} \text{MSE}(\hat{M}_n - M_n), \tag{1}$$

where N is the total number of joints.

3.1.2 Pose attributes learning

To ease the learning and inference of 3D limb joints, we introduce pose attribute as an additional input for 3D pose estimation. The main motivation of introducing such an attribute lies in that we aim to encode more visual cues, together with the 2D estimated poses, into the 3D estimation; meanwhile, such visual cues should be estimated reliably. To this end, we take pose attribute to be a three-class categorization of the relative location of a limb joint with respect to the torso.

Specifically, we define a torso plane to be the one where five body parts lie: left and right shoulder, left and right hip, and the pelvis. In practice, this plane is regressed using *Orthogonal Distance Regression*, where the sum of the Euclidean distances of the five points to the plane is minimized. We then compute the Euclidean distances between the obtained torso plane and the joints on the four limbs, including left and right elbow, left and right wrist, left and right knee, left and right ankle, as well as the head. Based on the derived distance, a predefined threshold, as well as

the side of the plane where the joint lies, we assign to each join one of the three labels, *front*, *back*, and *on-plane*. The joints with distances smaller than the threshold are taken to be on-plane.

Let p_i denote the ground-truth probability distribution of the pose attribute on joint i, and let $\hat{p}_i$ denote the estimated one. Also, let $\mathcal{J} = \{$l-Elbow, l-Wrist, r-Elbow, r-Wrist, l-Knee, l-Ankle, r-Knee, r-Ankle, head$\}$ denote the set of limb joints. Our model predicts all the nine attributes simultaneously with a single network, for which a multi-output cross entropy loss is adopted for training:

$$\mathcal{L}_{\text{attr}} = \frac{1}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \text{CrossEntropy}(\hat{p}_i, p_i), \tag{2}$$

where $|\mathcal{J}|$ denotes the cardinality of $\mathcal{J}$.

When training on 2D datasets without 3D annotations, however, we have no attribute supervisions available. To deal with this problem, we treat the 2D and 3D training examples as images from different domains, and adopt an unsupervised domain adaptation method [8] to help the multi-task network to generate domain-independent features for attribute prediction. A classifier is trained to distinguish the domain of the input based on the features to be fed into the attribute predictor, while the multi-task network is trained to fool the domain classifier by generating domain-independent features.

Let us denote the ground truth probability distribution of domain with q and the corresponding prediction with $\hat{q}$, the loss function for the domain classifier is taken to be

$$\mathcal{L}_{\text{domain}} = \text{CrossEntropy}(\hat{q}, q). \tag{3}$$

Specifically, we adopt a gradient reversal layer [8] to connect the multi-task network and the domain classifier. In the forward propagation, the gradient reversal layer acts as an identity function, while in the backward one, it multiplies the gradient by $-\lambda$, where $\lambda > 0$. As a result, the parameters in the multi-task network are updated in a way to *increase* the loss of the domain classifier, which means the CNN tries to learn domain-independent features. In the ideal case, the accuracy of the domain classifier is 50%, which means attribute features extracted by the multi-task network is not distinguishable at all, so it is domain-independent. An attribute predictor trained on this domain-independent features is also domain-independent. The experiment, as will be demonstrated in Tab. 4, gives strong support to the domain adaptation method above. The attribute predictor achieves an accuracy of 84.0% when using this domain adaptation method, and 82.7% without it. The attribute prediction accuracy on the validation set of MPI-INF-3DHP dataset is 70.1% without using any training data from this dataset, which also demonstrates the effectiveness of the domain adaptation method.

3.2. 3D pose estimation network

The 3D pose network takes as input the estimated 2D poses and pose attributes, and explicitly models the bi-directional dependencies among the body parts of different DOFs to produce the final 3D pose estimation. By categorizing the joints into three groups, the *torso*, the *proximal limb joints* including the head, the elbows and the knees, and the *distal limb joints* including the wrists and ankles, we allow the locations of the higher-DOF groups to be dependent variables of those of the lower-DOF ones, and in turn, constrain the latter using the former.

Specifically, such bi-directional dependency is achieved via a two-block network architecture, as depicted in Fig. 2. Each block models the body parts dependency from one of the two directions. Let us denote the 3D joint locations in the three groups as Y_1, Y_2 and Y_3. In Block-I, the locations of joints in the lowest-DOF group, $\hat{Y}_{11}$, are inferred from the image evidences learned by the multi-task network, using a basic regression component $G_{11}(\cdot;\theta_{11})$. The prediction results $\hat{Y}_{11}$, due to their low DOF, are usually plausible. The locations of joints $\hat{Y}_{12}$ and $\hat{Y}_{13}$ in the higher-DOF groups, are estimated from both the image evidences and their dependence upon the predictions of lower-DOF groups. For Block-I, we therefore have,

$$\begin{cases} \hat{Y}_{11} &= G_{11}(X;\theta_{11}), \\ \hat{Y}_{12} &= G_{12}(X, \hat{Y}_{11};\theta_{12}), \\ \hat{Y}_{13} &= G_{13}(X, \hat{Y}_{12}, \hat{Y}_{11};\theta_{13}), \end{cases} \tag{4}$$

where X denotes the image evidences, G_{ij} denotes the network module that regresses group j in block i, and θ_{ij} represents the learnable parameters in G_{ij}.

In Block-II, we enforce the derived high-DOF parts to constrain where the low-DOF ones may lie. In other words, such dependency is modeled in a reversed direction as the one in Block-I. We write,

$$\begin{cases} \hat{Y}_{21} &= G_{21}(X, \hat{Y}_{12}, \hat{Y}_{13};\theta_{21}), \\ \hat{Y}_{22} &= G_{22}(X, \hat{Y}_{21}, \hat{Y}_{13};\theta_{22}), \\ \hat{Y}_{23} &= G_{23}(X, \hat{Y}_{21}, \hat{Y}_{22};\theta_{23}), \end{cases} \tag{5}$$

where, again, $\hat{Y}_{ij}$, G_{ij}, and θ_{ij} respectively denote recovered pose locations, network module, and learnable parameters.

The final 3D pose prediction of the Block-s, $\hat{Y}_s$, is the concatenation of all the body parts. The loss function is taken to be,

$$\mathcal{L}_{\text{3D}} = \sum_{s \in \{1,2\}} |\hat{Y}_s - Y_s|. \tag{6}$$

Here we choose the L_1 loss over L_2 as the former shows consistent better performances in our experiments.

4. Experiments

In this section, we first introduce the datasets and protocols we used, then provide our implementation details, and

Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Tekin *et al*. [34]	102.4	147.2	88.8	125.3	118.0	182.7	112.4	129.2	138.9	224.9	118.4	138.8	126.3	55.1	65.8	125.0
Zhou *et al*. [48]	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5	199.2	107.4	118.1	114.2	79.4	97.7	113.0
Du *et al*. [6]	85.1	112.7	104.9	122.1	139.1	135.9	105.9	166.2	117.5	226.9	120.0	117.7	137.4	99.3	106.5	126.5
Zhou *et al*. [46]	91.8	102.4	96.7	98.8	113.4	125.2	90.0	93.8	132.2	159.0	107.0	94.4	126.0	79.0	99.0	107.3
Chen *et al*. [4]	89.9	97.6	90.0	107.9	107.3	139.2	93.6	136.1	133.1	240.1	106.7	106.2	114.1	87.0	90.6	114.2
Tome *et al*. [36]	65.0	73.5	76.8	86.4	86.3	110.7	68.9	74.8	110.2	173.9	85.0	85.8	86.3	71.4	73.1	88.4
Rogez *et al*. [28]	76.2	80.2	75.8	83.3	92.2	105.7	79.0	71.7	105.9	127.1	88.0	83.7	86.6	64.9	84.0	87.7
Pavlakos *et al*. [22]	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
Nie *et al*. [19]	90.1	88.2	85.7	95.6	103.9	103.0	92.4	90.4	117.9	136.4	98.5	94.4	90.6	86.0	89.5	97.5
Tekin *et al*. [33]	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	74.3	69.7
Zhou *et al*. [45]	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.2	66.1	51.4	63.2	55.3	64.9
Martinez *et al*. [14]	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*. [30]	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*. [7]	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
Rhodin *et al*. [26]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	66.8
Yang *et al*. [41]	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
Pavlakos *et al*. [21]	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
Lee *et al*. [12]	**43.8**	51.7	48.8	53.1	**52.2**	74.9	52.7	**44.6**	**56.9**	74.3	56.7	66.4	68.4	47.5	45.6	55.8
Dabral *et al*. [5]	46.9	53.8	**47.0**	52.8	56.9	**63.6**	**45.2**	48.2	68.0	94.0	55.7	51.6	55.4	40.3	44.3	55.5
Rogez *et al*. [29]	50.9	55.9	63.3	56.0	65.1	70.7	52.1	51.9	81.1	90.7	64.7	54.6	61.1	44.7	53.7	61.2
Ours	44.7	**48.9**	**47.0**	**49.0**	56.4	67.7	48.7	47.0	63.0	78.1	**51.1**	**50.1**	54.5	**40.1**	**43.0**	**52.6**
Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Akhter & Black [1]	199.2	177.6	161.8	197.8	176.2	186.5	195.4	167.3	160.7	173.7	177.8	181.9	176.2	198.6	192.7	181.1
Ramakrishna *et al*. [25]	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6	175.6	160.4	161.7	150.0	174.8	150.2	157.3
Zhou *et al*. [47]	99.7	95.8	87.9	116.8	108.3	107.3	93.5	95.3	109.1	137.5	106.0	102.2	106.5	110.4	115.2	106.7
Bogo *et al*. [3]	62.0	60.2	67.8	76.5	92.1	77.0	73.0	75.3	100.3	137.3	83.4	77.3	86.8	79.7	87.7	82.3
Moreno-Noguer [17]	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3	103.5	74.6	92.6	69.6	71.5	78.0	73.2	74.0
Pavlakos *et al*. [22]	47.5	50.5	48.3	49.3	50.7	55.2	46.1	48.0	61.1	78.1	51.1	48.3	52.9	41.5	46.4	51.9
Martinez *et al*. [14]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	59.4	49.2	45.0	49.5	38.0	43.1	47.7
Fang *et al*. [7]	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 *et al*. [21]	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
Lee *et al*. [12]	38.0	39.1	46.3	44.4	49.0	55.1	40.2	41.1	53.2	68.9	51.0	39.1	56.4	33.9	38.5	46.2
Dabral *et al*. [5]	**32.8**	**36.8**	42.5	**38.5**	**42.4**	49.0	**35.4**	**34.3**	53.6	66.2	46.5	**34.1**	42.3	**30.0**	39.7	42.2
Ours	33.6	38.1	**37.6**	**38.5**	43.4	48.8	36.0	35.7	51.1	63.1	**41.0**	38.6	**40.9**	30.3	**34.1**	**40.7**

Table 2. Detailed results on Human3.6M under Protocol #1 and #2. All the numbers recorded in the table refer to the mean per joint position errors (MPJPE) in millimeter. The results of all approaches are taken from the original papers. Our method outperforms all previous state-of-the-art methods, in terms of the average of the results.

next show both the quantitative and qualitative results as well as the ablation studies. Additional results can be found in our supplementary material.

4.1. Datasets and protocols

We evaluate our method on the following three popular human pose benchmarks.

Human3.6M [11]. It contains 3.6 million images and the corresponding 2D pose and 3D pose annotations captured in an indoor environment, featuring 7 subjects performing 15 everyday activities like "Eating" and "Walking". We follow the standard protocol on Human3.6M to use S1, S5, S6, S7 and S8 for training and S9 and S11 for evaluation. The evaluation metric is the mean per joint position error (MPJPE) in millimeter between the ground-truth and the prediction across all cameras and joints after aligning the depth of the root joints. We refer to this as Protocol #1. In some works, the predictions are further aligned with the ground-truth via a rigid transformation. We refer to this as Protocol #2. Following [48, 22, 45, 21], we down sampled the original videos from 50fps to 10fps to remove redundancy. We employed all camera views and trained a single model for all activities.

MPII [2]. It is the most widely used benchmark for 2D human pose estimation. It contains 25K in-the-wild images collected from YouTube videos covering a wide range of activities. It provides 2D annotations but no 3D ground truth. As a result, direct image-to-3D training is not a practical option with this dataset. We adopt this dataset for the training and testing of the multi-task network and for the qualitative evaluation of our 3D pose estimation method.

MPI-INF-3DHP [15]. It is a 3D pose dataset constructed by the Mocap system with both constrained indoor scenes and complex outdoor scenes. We only use the test split of this dataset, which contains 2929 frames from six subjects performing seven actions, to evaluate the generalization ability quantitatively and qualitatively.

4.2. Implementation details

Our method is implemented using PyTorch [20]. The training procedure of our network consists of three steps: training the multi-task network, training progressive regression network, and connecting them and fine-tuning. For the first step, the multi-task network is trained for 60 epochs. The learning rate is set to 5×10^{-4} and batch size is 12. For the second step, the 3D pose regression network is trained on predicted 2D key point positions and ground-truth pose attributes for 60 epochs. The learning rate is set

Model	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Avg.
HG	96.3	95.0	89.0	84.5	87.1	82.5	78.3	87.6
[41]	96.1	95.6	89.9	84.6	87.9	84.3	81.2	88.6
Ours	94.7	94.0	90.8	88.7	84.9	83.0	83.4	88.5

Table 3. PCKh@0.5 score on the MPII validation set. Joints are grouped by bilateral symmetry (ankles, wrists, etc). HG represents the pretrained hourglass model [18]. The 2D pose detection performance of our multi-task network is very close to that of [41], while our 3D results are much better (see Tab. 2 and Tab. 5).

to 2.5×10^{-4} and batch size is 64. For the final step, the multi-task network and the 3D pose network are connected with soft argmax layers and fine-tuned for 40 epochs. The learning rate is set to 1.0×10^{-4} and batch size is 64. The first and third steps are trained on the mixture of MPII and Human3.6M datasets. The training examples are randomly sampled from the two datasets with equal probability. Augmentation of random scale (1 ± 0.2) and random color jitter (1 ± 0.2) are used for both datasets. For the MPII dataset, random rotation ($\pm 30^\circ$) and random horizontal flipping are also used. The RMSprop optimizer is used for all the training steps. The whole training procedure takes about 2 days on two Tesla V100 GPUs with 16G memory for each.

4.3. Quantitative results

In what follows, we show our quantitative results on 2D pose estimation, on attribute prediction, on 3D pose estimation, qualitative results, and ablation studies.

4.3.1 2D pose estimation results on MPII

The accuracy of 2D pose detection is known to be crucial for 3D estimation [14]. Although our 2D detector is trained on the mixture of MPII and Human3.6m dataset, the PCKh@0.5 score of our model on the MPII validation split is very close to previous works [41] (see Tab. 3). This indicates that the performance improvement does not rely on a extremely well-trained 2D detector.

4.3.2 Performance of attribute prediction

In this section, we conduct experiments to find out the best training strategy to learn the attributes. There are three candidate training strategies, training on only Human3.6M, training on the mixture of MPII and Human3.6M, and training on the mixture with domain adaptation (DA). From Tab. 4 we can see that the mix-training strategy can significantly boost the attribute prediction accuracy. With the help of the domain adaptation, the attribute prediction is further improved.

It is worth noting that our attribute predictor also works well on the MPI-INF-3DHP dataset without using any examples from this dataset for training, which shows that our multi-task network successfully learns to transfer between domains.

Dataset	Method	Head	Elb.	Wri.	Knee	Ank.	Avg.
H36M	h36m	75.7	77.2	80.6	82.0	77.9	78.6
	mix	79.2	80.9	87.5	84.0	82.1	82.7
	mix+DA	79.4	82.6	88.4	85.9	83.6	**84.0**
MPI3D	h36m	47.5	48.4	58.7	59.6	41.4	51.1
	mix	74.6	67.1	72.5	69.2	55.3	67.7
	mix+DA	73.1	65.0	71.1	79.7	61.8	**70.1**

Table 4. The accuracy of attribute prediction on the Human3.6M (H36M) and the MPI-INF-3DHP (MPI3D) dataset. *H36m* stands for training using only Human3.6M, *mix* stands for using the mixture of Human3.6M and MPII, and *DA* stands for using the domain adaptation method discussed in Section 3.1.2. No training data from MPI-INF-3DHP have been used for training.

	[15]	[45]	[21]	[41]	ours
3DPCK	64.7	69.2	**71.9**	69.0	**71.9**
AUC	31.7	32.5	35.3	32.0	**35.8**

Table 5. 3DPCK and AUC on the MPI-INF-3DHP dataset. The results for all approaches are taken from the original papers. No training data from this dataset have been used for training.

4.3.3 3D pose estimation results on Human3.6M

We evaluate our method using the two most popular protocols (see Section 4.1) on Human3.6M. The detailed results of our method and previous state-of-the-art methods are listed in Tab. 2. Our method outperforms previous methods, in terms of the average result on all the actions.

4.3.4 3D pose estimation results on MPI-INF-3DHP

We evaluate our method on another unseen 3D human pose dataset, MPI-INF-3DHP [15], to test the cross-domain generalization ability. We follow [15, 16, 45, 41, 21] to use 3DPCK and AUC as the evaluation metrics. Comparisons with previous methods are shown in Tab. 5. Our method outperforms the prior ones on this unseen dataset, demonstrating the robustness of our method on domain shift.

4.4. Qualitative results

In Fig. 3 we show visualizations of several 3D pose predictions of our method on Human3.6M. As we may observe, our results are visually very close to the ground truths and considerably better than the baseline approach to be discussed in Section 4.5. Besides, in Fig. 4 we give the qualitative results on images in other scenes, including those from MPII and MPI-INF-3DHP, to show the robustness of our method to domain shift.

4.5. Ablation studies

To analyze the effectiveness of each component, we conduct ablation study on Human3.6M under Protocol #1. The mean per joint error is reported in Tab. 6. The notations are defined as follows:

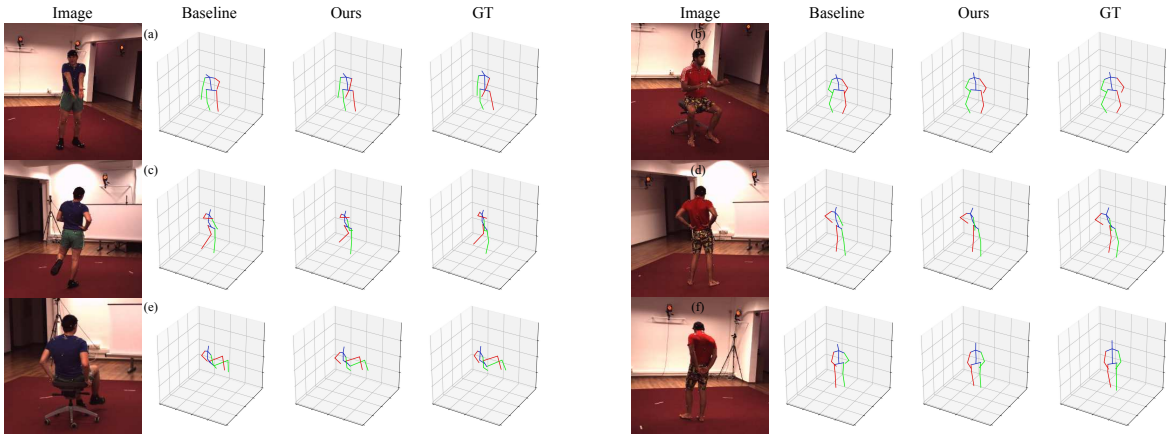

Figure 3. Qualitative results on H3.6M. Our predictions of limbs are significantly better than those of the baseline, defined in Section 4.5.

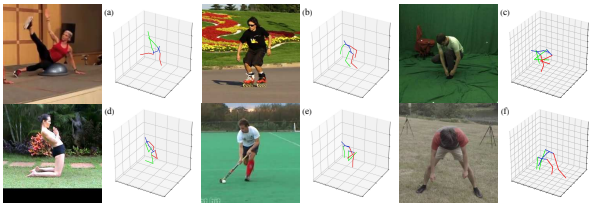

Figure 4. Qualitative results on datasets with domain shift. The first two columns are from MPII and the last is from MPI-INF-3DHP.

- **Baseline** refers to the approach that adopts the same network architecture as ours, but without modeling bi-directional dependencies among body parts and without pose attribute as input. In other words, we model only,

$$Y_n = G_n(X; \theta_n) \quad n \in \{1, 2, 3\}. \tag{7}$$

- **Progressive** refers to the bi-directional approach, as introduced in Section 3.2, but without pose attribute as input.

- **Attr** refers to using the predicted pose attributes to estimate the 3D pose.

We also re-implemented the method in [14] for comparison. Our implementation in fact yields results slightly better than those reported in the original paper.

Although the number of parameters of the baseline model and the progressive model are almost the same, the performance of the later is significantly better. From this comparative experiment we can see that the bi-directional model is indeed effective in 3D pose estimation. The proposed attributes further boosts the performance by a large margin, especially on the joints where a pose attribute is defined, proving the effectiveness the proposed pose attributes.

5. Conclusion

We propose in this paper a two-step 3D pose estimation approach that explicitly models the bi-directional dependencies among body parts of different DOFs. In the first step, we adopt a multi-task network that jointly estimates the 2D poses and the pose attributes for each limb joint, a three-class categorization that depicts the relative location between a joint and the torso plane. The pose attribute, unlike the more challenging regression-based depth estimation, provides a dependable yet informative prior of the joint locations. The predictions of 2D poses and attributes are then fed to the 3D pose estimation network, where higher-DOF parts are explicitly modeled as dependent variables of lower-DOF parts and meanwhile constrain the locations of the lower-DOF ones. In this way, body parts of different DOFs supervise and benefit one another, together yielding the encouraging results that outperform the state of the art on standard benchmarks.

Joint	Hip	Spine	Thorax	Shoulder	Head
[14]	20.7	37.6	42.5	56.5	65.3
Baseline	20.9	38.3	43.0	56.4	65.0
Progressive	21.4	37.9	42.8	56.0	63.6
Progressive + Attr	**20.4**	**36.8**	**40.6**	**52.2**	**58.4**
Joint	Elbow	Knee	Wrist	Ankle	Avg.
[14]	81.6	58.7	100.3	84.8	59.4
Baseline	80.6	56.4	98.5	81.9	58.3
Progressive	78.4	55.7	94.2	80.1	56.9
Progressive + Attr	**71.3**	**51.0**	**87.6**	**74.7**	**52.6**

Table 6. The prediction errors of [14] and our model by turning some modules off.

Acknowledgement

This research was supported by Australian Research Council Projects FL-170100117 and DP-180103424. J. W. was supported by China Scholarship Council (CSC) Grant #201603170329.

References

[1] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, 2016.

[4] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *European Conference on Computer Vision*, 2018.

[6] Yu Du, Yongkang Wong, Yonghao Liu, Feilin Han, Yilin Gui, Zhen Wang, Mohan Kankanhalli, and Weidong Geng. Marker-less 3d human motion capture with monocular image sequence and height-maps. In *European Conference on Computer Vision*, 2016.

[7] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] Shaoli Huang, Mingming Gong, and Dacheng Tao. A coarse-fine network for keypoint localization. In *The IEEE International Conference on Computer Vision*, 2017.

[11] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[12] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *European Conference on Computer Vision*, September 2018.

[13] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *The IEEE International Conference on Computer Vision*, 2017.

[15] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision, 2017 International Conference on*, pages 506–516. IEEE, 2017.

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

[17] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, 2016.

[19] Bruce Xiaohan Nie, Ping Wei, and Song-Chun Zhu. Monocular 3d human pose estimation by predicting depth on joints. In *The IEEE International Conference on Computer Vision*, 2017.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[21] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] Gerard Pons-Moll, David J Fleet, and Bodo Rosenhahn. Posebits for monocular human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2344, 2014.

[24] Jiayan Qiu, Xinchao Wang, Stephen J Maybank, and Dacheng Tao. World from blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8493–8504, 2019.

[25] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *European Conference on Computer Vision*, pages 573–586. Springer, 2012.

[26] Helge Rhodin, Jrg Sprri, Isinsu Katircioglu, Victor Constantin, Frdric Meyer, Erich Mller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

[28] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[29] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[30] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *The IEEE International Conference on Computer Vision*, 2017.

[31] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *European Conference on Computer Vision*, 2018.

[32] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. In *British Machine Vision Conference*, number CONF, 2016.

[33] Bugra Tekin, Pablo Marquez Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *The IEEE International Conference on Computer Vision*, 2017.

[34] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[35] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *The IEEE International Conference on Computer Vision*, 2017.

[36] Denis Tome, Christopher Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[37] Xinchao Wang, Engin Türetken, François Fleuret, and Pascal Fua. Tracking interacting objects optimally using integer programming. In *European Conference on Computer Vision*, pages 17–32. Springer, 2014.

[38] Xinchao Wang, Engin Türetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects using intertwined flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2312–2326, 2015.

[39] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[40] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, pages 466–481, 2018.

[41] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[42] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[43] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, 2016.

[44] Xiaoqing Yin, Xinchao Wang, Jun Yu, Maojun Zhang, Pascal Fua, and Dacheng Tao. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. In *European Conference on Computer Vision*, pages 469–484, 2018.

[45] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *The IEEE International Conference on Computer Vision*, 2017.

[46] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In *European Conference on Computer Vision*, 2016.

[47] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, and Kostas Daniilidis. Sparse representation for 3d shape estimation: A convex relaxation approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1648–1661, 2017.

[48] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[49] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):901–914, 2018.