Propagating LSTM: 3D Pose Estimation Based on Joint Interdependency

Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee(✉)

Department of Electrical Electronic Engineering,
Yonsei University, Seoul, South Korea
{kasinamooth,mayddb100,slee}@yonsei.ac.kr

Abstract. We present a novel 3D pose estimation method based on joint interdependency (JI) for acquiring 3D joints from the human pose of an RGB image. The JI incorporates the body part based structural connectivity of joints to learn the high spatial correlation of human posture on our method. Towards this goal, we propose a new long short-term memory (LSTM)-based deep learning architecture named propagating LSTM networks (p-LSTMs), where each LSTM is connected sequentially to reconstruct 3D depth from the centroid to edge joints through learning the intrinsic JI. In the first LSTM, the seed joints of 3D pose are created and reconstructed into the whole-body joints through the connected LSTMs. Utilizing the p-LSTMs, we achieve the higher accuracy of about 11.2% than state-of-the-art methods on the largest publicly available database. Importantly, we demonstrate that the JI drastically reduces the structural errors at body edges, thereby leads to a significant improvement.

Keywords: 3D human pose estimation · Joint interdependency (JI)
Long short-term memory (LSTM)
Propagating LSTM networks (p-LSTMs)

1 Introduction

Human pose estimation has been extensively studied in computer vision research area [1–6]. In general, human pose estimation can be categorized into 2D and 3D pose estimations. While the former focuses on obtaining human 2D joint positions from an image, the latter aims to acquire human 3D joint positions from an image by additionally inferring human depth information. Since various applications need human depth information including human motion capture, virtual training, augment reality, rehabilitation, and 3D graphic avatar, 3D pose estimation has become more paid attention in this research area [7–12].

Early 3D pose estimation approaches attempted to map 2D image to 3D pose using handcrafted features [13–16]. With a recent development of deep learning technology, many researchers in [17–19] have applied it to their methods to acquire 3D pose directly from an image without the handcrafted features.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11211, pp. 123–141, 2018.
https://doi.org/10.1007/978-3-030-01234-2_8

However, these direct approaches limited the input to only 3D pose data captured in laboratory environments [20,21]. Alternatively, the authors in [4,22–30] used 2D poses derived from a generalized environment, so their networks have shown a superior performance to the direct 3D pose estimation approach. Nevertheless, most of these works overlooked the joint dependency of body called structural connectivity, which might lead to a degradation in pose estimation performance.

In [31], the authors applied the structural connectivity at a whole-body level to the cost function of a network. However, the whole-body based structural connectivity makes all the joints be coupled tightly, so it has a difficulty in reflecting actual attributes in regard to joint interdependency. For instance, if a person moves the right wrist, the right elbow and shoulder are triggered to move, but the joints of the left arm may be unaffected. In other words, the joints of the intra-body part are dependently operated while the joints of the inter-body part are quite irrelevant. Based on this observation, we attempt to embed this joint interdependency in conjunction with joint connectivity into our model, which would make it easier to estimate 3D pose more accurately.

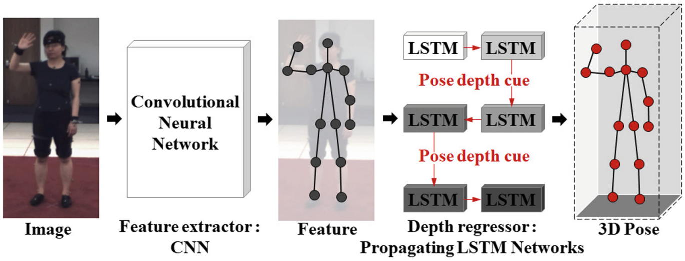

Fig. 1. Concept of the 3D pose estimation method. Convolutional neural network extracts a 2D pose from the input RGB video, which becomes a 3D pose through p-LSTMs via inferring depth cues implicitly.

In this paper, we present a novel 3D pose estimation method reflecting body part based structural connectivity as prior knowledge. Figure 1 gives an overall overview of our model. First, a 2D pose is extracted from the monocular RGB image by employing a 2D pose estimation method [2]. Second, the 3D pose is estimated using a proposed network named propagating long short term memory networks (p-LSTMs), which estimates depth information based on the 2D pose. In order to reflect the prior knowledge into p-LSTMs, we connect several LSTM networks in series. Those connected networks progressively elaborate the 3D pose while transferring the depth information called the pose depth cue. Eventually, the last LSTM network of the p-LSTMs builds the 3D pose of the whole body.

Our contributions are summarized as follows: (1) Unlike traditional approach that did not cover the joint interdependency based on actual human behavior, we develop a new model through utilizing body part based structural connectivity. In particular, to further refine the 3D pose, we adopt a multi-stage architecture in our method. (2) The effectiveness of our method is validated by extensive experiments on the largest 3D pose dataset [21]. It remarkably achieves an estimation accuracy improvement by 11.2% with competitive speed compared to the state-of-the-art methods.

2 Related Work

Estimating depth from an image is one of the most classic and challenging problems in computer vision. Many researchers have tried to reconstruct and analysis the 3D space closer to real world in a variety of areas [10,32–36]. 3D human pose estimation has to be robust against visual characteristics such as appearances, lights and motion. Early methods reconstructed human pose using a variety of invariant features such as silhouette [13], shape [14], SIFT [15], HOG [16]. Since deep learning technology can extract invariant features automatically from images, many researchers have brought this technology into 3D pose estimation [17–19]. Li *et al.* [17] applied a convolutional neural network (CNN) to directly estimate 3D pose from image. Grinciunaite *et al.* [18] exploited a 3D-CNN on sequential frames to obtain 3D pose. Although the 3D-CNN could obtain 3D pose from multiple frames, the estimations of complex 3D poses still do not demonstrate good performance. Pavlakos *et al.* [19] extended the existing 2D pose estimation method [2] to 3D. The authors used a coarse-to-fine strategy to handle the increase in dimensionality of the volumetric representation like 3D heatmap. However, the direct approaches using deep learning have a significant problem with generalization due to the lack of GT 3D pose data.

To efficiently enhance the poor performances, some approaches used 2D pose as a new invariant feature [4,22–27,29,30]. It is easier to convert the 2D pose to a 3D pose with high accuracy compared with conventional features. Moreover, currently, reliable 2D pose can be obtained owing to abundant databases. Many studies have paid attention to lifting dimension of pose from 2D to 3D. Zhou *et al.* [4] formulated an optimization problem in terms of the relationship between 2D pose and sparsity-driven 3D geometric prior, and predicted 3D pose by using an expectation-maximization algorithm. Chen *et al.* [22] and Yasin *et al.* [23] exploited the nearest-neighbor searching method to match the estimated 2D pose to a 3D pose from a large pose library. Tome *et al.* [24] proposed an iterative method which consisted of 2D pose method [1] and probabilistic 3D pose model. However, the systems, which are based on optimization and data retrieval, take a long time to obtain 3D pose and even require normalized input data.

As another attempt, many researchers in [25–27] used deep learning models to learn implicit pose structures from data when estimating 3D pose from 2D pose. Tekin *et al.* [25] extracted the 2D pose from an image by using a CNN and estimated the 3D pose by introducing an auto-encoder for 2D-to-3D estimation. This approach simply utilized an existing 2D pose estimation method by structurally connecting the auto-encoder to the CNN. Lin *et al.* [26] extracted 2D pose from an input image using the method in [1]. In addition, the LSTM was utilized to obtain the corresponding 3D pose from the extracted 2D pose.

Martinez *et al.* [27] proposed a simple model which works fast by using several fully connected networks. The 3D pose estimation performance has been greatly improved since the use of the 2D pose as invariant feature. However, these methods intended to automatically learn the relationship between 2D and 3D poses into the deep learning model without any prior knowledge of 3D human pose.

Some authors manually utilized prior knowledges such as kinematic model, body model, and structural connectivity [5,29–31]. These approaches reinforce our belief that prior knowledges are useful information to effectively train deep learning models when the dimension of pose increases from 2D to 3D. Zhou *et al.* [5] embedded a kinematic model layer into CNN. However, the parameters were hard to set due to the nonlinearity of the model. Furthermore, the method required hard assumptions such as fixed bone length and known scale. Bogo *et al.* [30] proposed an optimization process to fit the estimated 2D pose in [3] into the 3D human body model [37]. Moreno *et al.* [29] converted the input 2D pose from the joint position based vector to the Euclidean distance of joints based N-by-N matrix. Sun *et al.* [31] changed the cost function from per-joint error to per-bone(limb) error, and yet, to the best of our knowledge, the performance of the method [31] is currently highest in terms of pose estimation error.

However, the conventional methods overlooked an important notion from the perspective of interdependency of joints observed from the spatial and temporal behavior of the human body. Namely, the authors in [29,31] have exploited the structural connectivity of whole-body level as prior knowledge. Different to previous works, our novelty lies in embedding the body part based joint connectivity into the deep learning structure to reconstruct 3D pose more accurately.

3 3D Pose Estimation Method

3.1 System Architecture

Figure 2 illustrates the system architecture of our method. The proposed method consists of two deep learning models for 2D and 2D-to-3D pose estimations. The CNN extracts a 2D pose as the feature from the input RGB image in Fig. 2(b). Then, the proposed p-LSTMs, which is composed of 9 p-LSTMs serially, conducts the 2D-to-3D pose estimation stemming from the extracted 2D pose as shown in Fig. 2(d). The first 3D pose is constructed in the fully connected layers (FCs). Finally, the 3D pose is further refined by a multi-stage architecture of the 2D-to-3D pose estimation module as shown in Figs. 2(g) and (h).

3.2 Problem Statement

The main purpose of our method is to estimate the 3D human pose information from a given 2D input image. Towards this, a vast number of images and the corresponding 3D GT pose data are required. In general, the 2D human pose gives a more abstract representation of the human posture than that captured in raw image. Thus, the 2D-to-3D pose estimation by means of 2D pose is effective

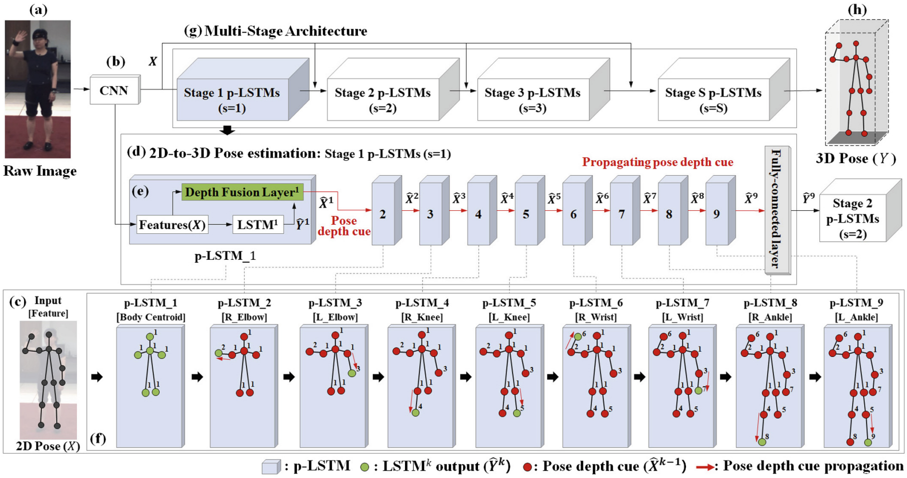

Fig. 2. System architecture of 3D pose estimation. (a) Input RGB image. (b) CNN extracts 2D pose from input data. (c) 2D pose extracted from (b). (d) Proposed p-LSTMs for extracting depth information from (c). (e) A unit of p-LSTM. (f) Procedure of constructing 2D-to-3D pose via p-LSTMs in accordance with the body part based structural connectivity of joints. (g) Multi-stage architecture. (h) Output of 3D pose. (Best seen in color and zoom.)

when estimating 3D pose from image [4,22–27,29–31]. We adopt the 2D pose estimation method in [2] as shown in Fig. 2(b). In this paper, the aim of our method is to learn a mapping function $f^* : \mathbb{R}^{2J} \rightarrow \mathbb{R}^{3J}$ through adding a depth dimension to the 2D pose with J joints. The mapping function uses $2J$ vectors for 2D pose X as input, and $3J$ vectors for 3D pose Y as output where $X = [x_1, \cdots, x_J]$, and $Y = [y_1, \cdots, y_J]$, respectively. The major objective of our method is to design the function f as a depth regressor.

3.3 Propagating LSTM Networks: p-LSTMs

We present a new deep learning model based on LSTM for estimating 3D pose from 2D pose, as shown in Fig. 2(d). In general, there is a limitation to estimate a 3D pose using single-frame 2D image only. If there are self-occluded cases in human pose, it would difficult for even human to answer the pose correctly, which significantly degrades the estimation performance. On the other hand, if multi-frame images are utilized, it should be much easier to handle the self-occluded issue. Hence, LSTM has demonstrated better performance in applications with time-correlated characteristics [26,38]. Lin *et al.* [26] only considered temporal correlation of input frames, but the proposed method includes spatial correlation as well as temporal correlation through the connection of multiple LSTM networks. Namely, in order to learn the spatial correlation of human pose, each LSTM network is sequentially connected to build a human body structure in a way of central-to-peripheral dimension extensions in accordance with natural human recognition over temporal domain.

Figure 2(e) shows the proposed p-LSTM which consists of one LSTM network and one depth fusion layer. From the 2D pose, the first p-LSTM only builds the 3D joints of the centroid part of the body, which is used as seed joints. Each p-LSTM builds its part of the 3D pose while connecting them to each other. The entire 3D pose is constructed in the order shown in Fig. 2(f). Then, the estimated 3D joints are merged into the input 2D pose in the depth fusion layer of the first p-LSTM. The merged information is propagated along the next sequentially connected LSTM networks. The final 3D pose is created by propagating the merged information, which is called the merged information pose depth cue. Finally, the propagated pose depth cue is regressed to the whole 3D pose via FC. To further refine the 3D pose, we adopt a multi-stage architecture in the p-LSTMs similar to previous works [1,24,26,39]. The pseudo code A1 shows the procedure of the algorithm for p-LSTMs.

A1: Algorithm of p-LSTMs

```
Variables
$k$: index of the p-LSTM
$K$: number of the p-LSTM
$\hat{Y^k}$: output of the $k^{th}$ LSTM network
$\hat{X^k}$: output of the $k^{th}$ depth fusion layer
$\mathrm{LSTM}^\mathrm{k}$: $k^{th}$ LSTM network
$\mathrm{Depth}^\mathrm{k}$: $k^{th}$ depth fusion layer
FC: fully connected layer
$Y_{\mathrm{Pred}}$: output of 3D pose

Input: $X$ (2D pose)
Output: $Y$ (3D pose)
for $k = 1$ to $K$
    if ($k$==1)
        $\hat{Y}^k = \mathrm{LSTM}^\mathrm{k}(\mathrm{X})$
        $\hat{X}^k = \mathrm{Depth}^\mathrm{k}(\hat{\mathrm{Y}}^\mathrm{k}, \mathrm{X})$
    else
        $\hat{Y}^k = \mathrm{LSTM}^\mathrm{k}(\hat{\mathrm{X}}^{\mathrm{k}-1})$
        $\hat{X}^k = \mathrm{Depth}^\mathrm{k}(\hat{\mathrm{Y}}^\mathrm{k}, \hat{\mathrm{X}}^{\mathrm{k}-1})$
return   $Y_{\mathrm{Pred}} = \mathrm{FC}(\hat{\mathrm{X}}^\mathrm{K})$
```

Propagating Connection: To reflect the joint interdependency (JI) into our method, the body part based structural connectivity is carefully dealt with. The movement of a body part leads to movements of its connected body parts dependently, but the other parts of the body may move independently. For example, a motion of the right elbow triggers the movement of its connected wrists and shoulders, but the other side (left part) may not be affected. In other words, even though the whole body is physically connected to each other, the motion of each body part is independent. Unlike previous studies [29, 31] which simply accounted for prior knowledge of the body, we attempt to embed the body part based structural connectivity into the deep learning structure. Since each body part has different characteristics (range of motion), it is decomposed to several LSTM blocks instead of the whole-body inference. In addition, each p-LSTM is linked to each other according to human body structure to induce dependency because each body part derived from whole-body indirectly influences each other. When the 3D pose to be estimated is based on 14 joints, 9 p-LSTMs are used for representing the human body structure as shown in Fig. 2(f). The first p-LSTM plays a role of populating the 3D joints of the body's centroid part which are utilized as seed joints. After that, the first output is generated from the first p-LSTM, which becomes an input of the next p-LSTM. In the second p-LSTM, the next neighbor parts are constructed according to the human body structure. In this way, the 9 p-LSTMs are connected and each output is propagated to the other parts along the connection.

Pose Depth Cue: From the second p-LSTM to the last p-LSTM, the p-LSTM must rely solely on the output of the preceding p-LSTM because the initial 2D pose information disappears after the first p-LSTM. The spatial correlation of the 2D pose could be useful when estimating a 3D pose. Since human recognizes the structural connectivity (spatial correlation) of pose, human can easily reconstruct 3D pose according to the change of 2D joint position. For example, when the 2D positions of the wrist and elbow joints approach each other, the 3D positions of the two joints move along depth-axis. In fact, the limb connected by the wrist and elbow joints is structurally unchanged in length. To prevent the initial 2D pose from disappearing, each p-LSTM uses the input 2D pose as ancillary data and merges it with its own output in the depth fusion layer. In the proposed method, the depth information is gradually estimated through the newly generated input feature, and the spatial correlation of the human body is learned. Thus, the merged auxiliary and input data are called pose depth cue. In other words, the pose depth cue is created by integrating the 2D with 3D poses in the depth fusion layer, as shown in Fig. 2(e) and lines 4 and 7 of A1.

Different types of the pose depth cues can be created depending on how the 2D and 3D poses are merged. **(1) Elimination and addition method:** it deletes the 2D pose and only uses some of estimated 3D pose (no auxiliary data). **(2) Concatenated method:** it simply concatenates the 2D and 3D poses. **(3) Replacement method:** it replaces some of 2D pose with some of estimated 3D pose. Figure 3 depicts the three pose depth cues in details.

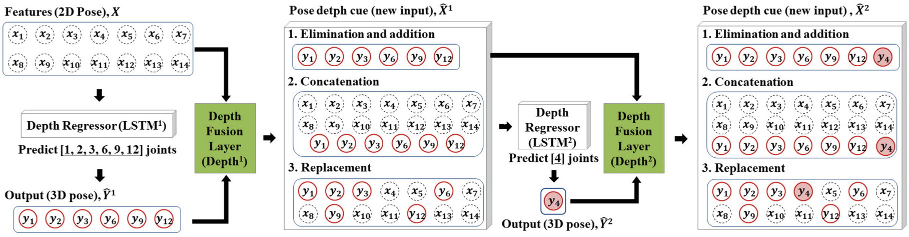
Features (2D Pose), X
x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14}
Depth Regressor (LSTM^1)
Predict [1, 2, 3, 6, 9, 12] joints
Output (3D pose), $\hat{Y}^1$
y_1 y_2 y_3 y_6 y_9 y_{12}
Depth Fusion Layer (Depth^1)
Pose detph cue (new input), $\hat{X}^1$
1. Elimination and addition
y_1 y_2 y_3 y_6 y_9 y_{12}
2. Concatenation
x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14}
y_1 y_2 y_3 y_6 y_9 y_{12}
3. Replacement
y_1 y_2 y_3 x_4 x_5 y_6 x_7
x_8 y_9 x_{10} x_{11} y_{12} x_{13} x_{14}
Depth Regressor (LSTM^2)
Predict [4] joints
y_4
Output (3D pose), $\hat{Y}^2$
Depth Fusion Layer (Depth^2)
Pose depth cue (new input), $\hat{X}^2$
1. Elimination and addition
y_1 y_2 y_3 y_6 y_9 y_{12} y_4
2. Concatenation
x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14}
y_1 y_2 y_3 y_6 y_9 y_{12} y_4
3. Replacement
y_1 y_2 y_3 y_4 x_5 y_6 x_7
x_8 y_9 x_{10} x_{11} y_{12} x_{13} x_{14}

Fig. 3. Different types of the pose depth cues. (Best seen in zoom.)

The proposed 2D-to-3D pose estimation method consists of 9 p-LSTMs as shown in Fig. 2(d), and creates the pose depth cue for each depth fusion layer of the p-LSTMs. Passing through the p-LSTMs, the input pose depth cues change gradually. Figure 2(f) shows the procedure for the final pose depth cue to become the 3D pose. Although the proposed 2D-to-3D pose method is connected to multiple LSTM networks, the learning of the method is simple because it consists of an end-to-end network. To train the proposed p-LSTMs, the basic loss function can be represented by

$$\mathcal{L}_{3D}(Y_{pred}, Y_{GT}) = \frac{1}{|J|}\sum_{j}^{J}(Y_{pred} - Y_{GT})^2, \tag{1}$$

where Y_{pred} and Y_{GT} are predicted and GT 3D poses, respectively.

3.4 Training and Testing

For training, the final loss function of our method for 3D pose estimation is

$$\mathcal{L}_{3D}(Y_{pred}, Y_{GT}) = \sum_{s}^{S}\alpha_s\left[\sum_{t}^{T}\frac{1}{|J|}\sum_{j}^{J}(Y_{pred}^{t,s} - Y_{GT}^{t,s})^2 + \lambda\sum_{k}^{K}(w_k^s)^2\right], \tag{2}$$

where S is the stage number of the proposed method, T is the length of the input image frames, α_s is the weight for each stage, λ is the regularization parameter, w_k is the weight value of the k^{th} LSTM network, and K is the number of LSTM networks. When S is greater than 2, it means that the method is repeated S times. The final loss function consists of Euclidean distance of the GT 3D joint and the predicted 3D joint, and a regularization term is added for training stability. Our method is learned using an adaptive subgradient method (Adagrad optimizer) [40]. In the testing part, the input image comes in sequentially, and our proposed model processes it to estimate the 3D pose.

4 Experiments

4.1 Implementation Details

For implementation of our method, we used the Tensorflow [41], which is an open source deep learning library. We employed the conventional CNN [2] for 2D pose estimation. The 2D pose model was pretrained on the 2D pose dataset [42] and fine-tuned on the Human3.6M [21] or HumanEVA-I [20] datasets. One stage of p-LSTMs consists of 9 LSTM blocks, 9 depth fusion layers and 2 FCs. One LSTM block consists of one LSTM cell with 100 hidden units and one FC with 150 hidden units. In addition, 2 FCs with 45 hidden units were added after the p-LSTMs. The keeping probability of dropouts was set to 0.9 and 0.6 in the first and second FCs. Finally, in order to estimate the 3D human pose from the RGB image, we unified all of the aforementioned networks into an end-to-end network structure. In the training procedure of the deep learning model, the parameters of the model were initialized to uniform distribution [–0.1, 0.1]. The decay parameter and learning rate were set to $1e-4$ and $1e-2$, respectively. The stage loss weight α_s was set to 1. The total number of proposed model parameters is 31 million, consisting of 30 million from 2D part and 1 million from p-LSTMs. It took about 2 days to train our method with 10,000 epochs on GeForce TITAN X with 12 GB memory. The training batch size was set to 64. The testing time of the proposed method takes about 33.6 ms per image (RGB-to-2D and 2D-to-3D methods take about 33 ms and 0.6 ms per image, respectively).

4.2 Datasets and Evaluation

For performance evaluation, we used two public datasets, namely HumanEva-I [20] and Human3.6M [21], which were the most widely used for performance comparison in the 3D human pose estimation research.

Human3.6M: The Human3.6M dataset consists of 3.6 million images and 3D human poses. In addition, the dataset was recorded from 4 cameras with different views. The 3D human pose data consists of 11 subjects with 15 actions. Previous works [5,18,19,22–24,26–29,31,43,44] performed the evaluation according to several different protocols. In this paper, we followed 2 major protocols for performance comparison. Protocol 1 was used to train 5 subjects (S1, S5, S6, S7, and S8) and to test 2 subjects (S9 and S11). Training and testing were performed independently and all camera views were used. This protocol was used in [5,18,19,22,24,26–29,31,44]. The original videos were down-sampled from 50 fps to 10 fps. Protocol 2 was used to train 6 subjects (S1, S5, S6, S7, S8, and S9) and to test 1 subject (S11). The original videos were down-sampled by keeping every 64^{th} frame. This protocol was used in [22–24,29,31,43]. After the predicted 3D pose and GT 3D pose were aligned with the rigid transformation used in the Procrustes method [45], the error was computed.

HumanEva-I: The HumanEva-I dataset consists of RGB video sequences and 3D human pose. The RGB video sequences were recorded using 3 cameras with different views. The 3D human pose data consists of 3 subjects with 6 actions (walking, jogging, boxing, and so on). We trained the proposed method using the training dataset, and tested the method using the validation dataset in the same protocol as [23,26,27,46–49]. In the experiment, we excluded some results where rigid alignment was performed as post processing.

Evaluation Metric: We used the mean per joint position error (MPJPE) [21] as the evaluation metric, which is the most widely used performance index of 3D human pose estimation. The MPJPE simply calculates from the 3D Euclidean distance between GT and the predicted result. The error in millimeter is measured, and the GT value is obtained using infrared sensors.

4.3 Comparison with State-of-the-art Methods

Performance Comparison on Human3.6M: In Tables 1 and 2, the notations S and T mean the number of stages and the number of input frames, respectively. We compared the performance of the proposed method with state-of-the-art previous works on the Human3.6M dataset. In all the proposed methods of

Table 1. Comparison with the state-of-the-art methods for the Human3.6M under Protocol 1. The marks *, °, and † indicate a method using rigid alignment as post processing, GT 2D pose as input, and multiple frames as input, respectively.

Method (Protocol 1)	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SitD.	Smoke	Wait	Walk	WalkD.	WalkT.	Avg.
Chen, CVPR'17 [22]	89.8	97.5	89.9	107.8	107.3	139.1	93.5	136.0	133.1	240.1	106.6	106.2	87.0	114.0	90.5	114.1
Zhou, ECCV'16 [5]	91.8	102.4	96.9	98.7	113.3	125.2	90.0	93.8	132.1	158.9	106.9	94.4	79.0	126.0	98.9	107.2
Tome, CVPR'17 [24]	64.9	73.4	76.8	86.4	86.2	110.6	68.9	74.7	110.1	173.9	84.9	85.7	71.3	86.2	73.1	88.3
Pavlakos, CVPR'17 [19]	67.3	71.9	66.7	69.0	71.9	76.9	65.0	68.3	83.6	96.5	71.7	65.8	59.1	74.8	63.2	71.9
Martinez, ICCV'17 [27]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	49.5	65.1	52.4	62.9
Sun, ICCV'17 [31]	52.8	54.8	54.2	54.3	61.8	**67.2**	53.1	53.6	71.7	86.7	61.5	**53.4**	**47.1**	**61.6**	53.4	59.1
Our p-LSTMs ($S = \mathbf{3}$, $T = \mathbf{1}$)	**43.8**	**51.7**	**48.8**	**53.1**	**52.2**	74.9	**52.7**	**44.6**	**56.9**	**74.3**	**56.7**	66.4	47.5	68.4	**45.6**	**55.8**
Zhou, ICCV'17 [28]*	54.8	60.7	58.2	71.4	62.0	**65.5**	53.8	55.5	75.2	111.5	64.1	66.0	63.2	**51.4**	55.3	64.9
Martinez, ICCV'17 [27]*	39.5	43.2	46.4	47.0	51.0	56.0	41.4	**40.6**	56.5	69.4	**49.2**	45.0	38.0	**49.5**	43.1	47.7
Our p-LSTMs ($S = \mathbf{3}$, $T = \mathbf{1}$)*	**38.0**	**39.3**	**46.3**	**44.4**	**49.0**	**55.1**	**40.2**	41.1	**53.2**	**68.9**	51.0	**39.1**	**33.9**	56.4	**38.5**	**46.2**
Moreno, CVPR'17 [29]°	53.5	50.5	65.7	62.4	56.9	80.8	60.6	50.8	55.9	79.6	63.6	61.8	59.4	68.5	62.1	62.1
Martinez, ICCV'17 [27]°	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	36.4	47.6	40.4	45.5
Our p-LSTMs ($S = \mathbf{3}$, $T = \mathbf{1}$)°	34.6	39.7	37.2	40.9	45.6	50.5	42.0	39.4	47.3	48.1	39.5	38.0	31.9	41.5	37.2	**40.9**
Grinciunaite, ECCV'16 [18]†	91	89	94	102	105	151	99	112	151	239	109	106	101	141	106	119
Lin, CVPR'17 [26]†	58.0	68.2	63.3	65.8	75.3	93.1	61.2	65.7	98.7	127.7	70.4	68.2	50.6	72.9	57.7	73.1
Hossain, Thesis [44]†	44.2	**46.7**	52.3	**49.3**	59.9	**59.4**	**47.5**	46.2	59.9	**65.6**	55.8	50.4	43.5	**52.3**	45.1	**51.9**
Our p-LSTMs ($S = \mathbf{3}$, $T = \mathbf{3}$)†	**40.2**	49.2	**47.8**	52.6	**50.1**	75.0	50.2	**43.0**	**55.8**	73.9	**54.1**	55.6	**43.3**	58.2	**43.3**	52.8
Our p-LSTMs ($S = \mathbf{3}$, $T = \mathbf{3}$)†,°	32.1	36.6	34.3	37.8	44.5	49.9	40.9	36.2	44.1	45.6	35.3	35.9	30.3	37.6	35.5	**38.4**

Table 2. Comparison with the state-of-the-art methods for the Human3.6M under Protocol 2. All methods use rigid alignment as post processing.

Method (Protocol 2)	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SitD.	Smoke	Wait	Walk	WalkD.	WalkT.	Avg.
Yasin, CVPR'16 [23]	88.4	72.5	108.5	110.2	97.1	142.5	81.6	107.2	119.0	170.8	108.2	86.9	92.1	165.7	102.0	110.1
Rogez, NIPS'16 [43]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	88.1
Chen, CVPR'17 [22]	71.6	66.6	74.7	79.0	70.0	93.2	67.5	89.3	90.7	195.6	83.4	71.1	55.7	85.8	62.5	82.7
Moreno, CVPR'17 [29]	66.0	61.6	84.5	73.7	65.2	92.5	67.1	60.8	67.2	103.4	74.7	69.5	71.4	78.0	73.2	73.9
Tome, CVPR'17 [24]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	70.7
Sun, ICCV'17 [31]	42.1	44.3	45.0	45.4	51.5	**53.0**	43.2	41.3	59.3	73.3	51.0	44.0	38.3	**48.0**	44.8	48.3
Our p-LSTMs ($S=\mathbf{3}$, $T=\mathbf{1}$)	37.4	38.9	45.6	43.8	48.5	54.6	39.9	39.2	53.0	68.5	51.5	38.4	33.2	55.8	37.8	**45.7**
Our p-LSTMs ($S=\mathbf{3}$, $T=\mathbf{3}$)	**34.9**	**35.2**	**43.2**	**42.6**	**46.2**	55.0	**37.6**	**38.8**	**50.9**	**67.3**	**48.9**	**35.2**	**31.0**	50.7	**34.6**	**43.4**

Tables 1 and 2, the replacement type of the pose depth cue is used. Tables 1 and 2 show the result of average 3D joint error (mm) w.r.t. the GT 3D joints in Protocol 1 and 2.

For a fair comparisons, Table 1 shows the results separately for the factors affecting performance such as format of input data or usage of post processing. In Table 1, the first sub-table (rows 1 to 7) show performance comparisons for single frame. Our result achieve a performance improvement of about 3.3 mm (5.9%) compared with [31]. Next sub-table is the results obtained by further calibrating the 3D pose using rigid alignment. We obtain a 1.5 mm (3.2%) lower prediction error compared with [27]. Third sub-table show the results when the 2D GT pose is used as input data. In the 3D pose estimation using 2D pose as feature, our method shows a potential performance by eliminating the influence of estimation accuracy of 2D pose methods. We achieve a gain of about 4.6 mm (11.2%) over [27]. Finally, the performances when multiple frames are used as input are shown in rows 14 to 17. The methods using multiple frames can achieve a robust 3D pose against noise such as self-occlusion using temporal correlation. A detailed description of the effects of multiple frames in the proposed method is given in Sect. 4.4. Our performance is slightly less than [44] in terms of accuracy, but the number of parameters is three times fewer than [44], which makes the computation significantly faster. For Protocol 2, our method shows the best performance except for the *photo* and the *walking with dog* scenarios including the case of using single frame. The authors in [24,43] only provided the average joint error. The results are quantitatively compared with [31], which improves the performance about 2.6 mm (5%) to 4.3 mm (9%). The *photo* scenario consists of very complex poses but the performance of our method is competitive. The proposed method outperforms all of state-of-the-art methods on average. The average error of Protocol 2 is lower than that of Protocol 1 because the deep learning based methods are effectively trained on more diverse datasets. From the experiments, it is demonstrated that the learning JI is effective from the regularization point of view.

Table 3. Comparison with the state-of-the-art methods for the HumanEva-I.

Method (HumanEVA-I)	Walking				Jogging				Boxing			
	S1	S2	S3	Avg.	S1	S2	S3	Avg.	S1	S2	S3	Avg.
Radwan, CVPR'13 [46]	75.1	99.8	93.8	89.6	79.2	89.8	99.4	89.5	-	-	-	-
Simo-Serra, CVPR'13 [47]	65.1	48.6	73.5	62.4	74.2	46.6	32.2	56.7	-	-	-	-
Kostrikov, BMVC'14 [48]	44.0	30.9	41.7	38.9	57.2	35.0	33.3	40.3	-	-	-	-
Tekin, CVPR'16 [49]	37.5	25.1	49.2	37.3	-	-	-	-	50.5	61.7	57.5	56.6
Yasin, CVPR'16 [23]	35.8	32.4	41.6	36.6	46.6	41.4	35.4	38.9	-	-	-	-
Lin, CVPR'17 [26]	26.5	20.7	38.0	28.4	41.0	29.7	29.1	33.2	**39.4**	57.8	61.2	52.8
Martinez, CVPR'17 [27]	19.7	**17.4**	46.8	28.0	26.9	18.2	18.6	21.2	-	-	-	-
Our p-LSTMs (S = 3, T = 1)	**18.6**	19.9	**30.5**	**23.0**	**25.7**	**16.8**	**17.7**	**20.1**	42.8	**48.1**	**53.4**	**48.1**

Performance Comparison on HumanEva-I: This dataset is also widely used for performance comparisons due to simple actions and fewer sequences compared to Human3.6M. For a fair comparison with previous works, we only learned and evaluated data recorded with camera 1. The number of hidden units in each LSTM network was 80. The results are summarized in Table 3. Some results of previous works were excluded because there were no results for the *jogging* and *boxing* scenarios. Our result shows the best performance for all the actions, and improves from 1.1 mm (5%) to 5 mm (17.8%) over the state-of-the-art methods. The average joint error of the *boxing* scenario higher than that of the others due to self-occlusion action. For the *jogging* scenario, a margin of performance improvement is the least.

4.4 Ablative Study (Effect of the p-LSTMs)

Tables 4 and 5 show the effect of the p-LSTMs via ablation test. Our baseline consists of one LSTM and 2 FCs and the number of hidden units in LSTM and each FC are 80 and 45, respectively.

Multi-stage Architecture: In order to improve the performance of the proposed method, a multi-stage architecture is used, which consists of concatenated multiple of p-LSTMs. Furthermore, the input of next stage consists of concatenating the initial input 2D pose and the predicted 3D pose from the current stage. Experimental results according to the multi-stage architecture are shown from the rows 1 to 7 of Table 4. The multi-stage parameter S is set to 2, which means that the structure of p-LSTMs are repeated twice. In this experiment, the more stages were configured, the longer the training took, but the better

Table 4. Results of the baseline and variants on Human3.6M under Protocol 1.

Method (Protocol 1)	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SitD.	Smoke	Wait	Walk	WalkD.	WalkT.	Avg.
Single LSTM (S=1, T=1)	97.2	109.8	96.5	95.6	98.4	134.2	93.6	94.8	134.6	145.2	96.7	106.5	117.1	106.9	100.6	108.5
Single LSTM (S=3, T=1)	90.4	98.9	94.8	91.2	95.2	133.9	90.4	91.1	132.6	142.1	95.9	98.9	104.4	105.5	99.1	104.3
Single LSTM (S=1, T=3)	91.1	96.2	93.5	92.4	95.6	133.5	84.8	88.9	131.2	139.4	94.0	100.3	98.4	99.6	90.4	102.0
Single LSTM (S=3, T=3)	86.5	93.4	90.9	88.7	92.1	129.8	80.1	86.2	127.8	135.6	92.1	88.1	92.5	93.2	86.8	97.6
p-LSTMs (S=1, T=1)	45.0	54.3	50.6	57.1	55.8	77.6	56.5	47.0	58.5	77.3	58.0	69.2	50.4	70.6	48.4	58.4
p-LSTMs (S=2, T=1)	44.4	53.1	49.7	54.6	53.8	76.8	54.9	45.1	57.3	76.0	57.6	68.0	48.2	68.9	45.7	56.9
p-LSTMs (S=3, T=1)	**43.8**	**51.7**	**48.8**	**53.1**	**52.2**	**74.9**	**52.7**	**44.6**	**56.9**	**74.3**	**56.7**	**66.4**	**47.5**	**68.4**	**45.6**	**55.8**
p-LSTMs (S=3, T=3)	**40.2**	**49.2**	**47.8**	**52.6**	**50.1**	**75.0**	**50.2**	**43.0**	**55.8**	**73.9**	**54.1**	**55.6**	43.3	**58.2**	**43.3**	**52.8**
p-LSTMs (S=3, T=5)	41.7	59.0	60.2	60.8	58.8	83.5	58.4	54.4	59.4	86.2	60.4	69.9	**39.6**	66.5	44.8	60.2
p-LSTMs (S=3, T=10)	42.5	67.6	65.3	69.2	76.3	90.4	63.8	62.5	79.6	93.9	64.1	76.3	43.2	73.5	61.9	68.8

Table 5. Joints error on Human3.6M. D, o, and i are the type of the pose depth cue, the propagation to outward direction, and the propagation to inward direction, respectively. The p-LSTMs consist of 3 stages and 1 input frame.

Method (Protocol 1)	Head	Neck	R_shld	R_elbow	R_wrist	L_shld	L_elbow	L_wrist	R_hip	R_knee	R_ankle	L_hip	L_knee	L_ankle	Avg.
Single LSTM (baseline)	79.1	39.1	88.9	103.5	130.3	90.0	105.4	140.6	79.5	98.5	197.8	75.7	90.1	200.4	108.5
p-LSTMs (D=1, o)	65.1	33.4	83.5	100.4	110.1	87.6	97.4	108.1	74.5	80.0	117.1	74.3	80.5	116.2	87.6
p-LSTMs (D=2, o)	63.6	28.6	81.5	95.9	103.8	80.4	87.3	102.6	66.1	74.3	99.6	64.1	73.0	97.7	79.8
p-LSTMs (D=3, o)	**52.4**	**16.2**	**34.6**	**56.6**	**74.3**	**34.1**	**58.9**	**74.9**	**44.2**	**63.5**	**83.1**	**48.4**	**58.6**	**81.4**	**55.8**
p-LSTMs (D=3, i)	69.6	35.4	85.1	85.5	96.3	86.6	84.9	97.5	76.8	81.7	105.8	76.9	83.1	105.3	83.6

the performance, which was improved by up to 2.6 mm (4.4%). The multi-stage architecture refined the 3D pose, which was initially estimated, and was repeated as the structure of the same method was repeated.

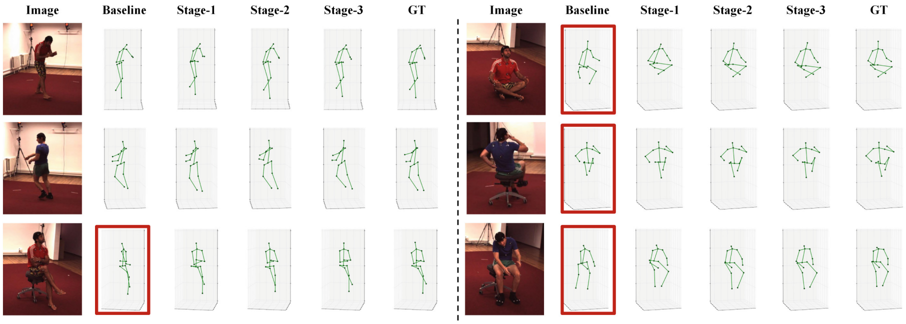

Fig. 4. Qualitative comparisons of the baseline and variants on the Human3.6M dataset. The 3D human poses are represented by the baseline, stage-1, stage-2, stage-3 of our method ($S = 3, T = 1$) and ground-truth, respectively.

Figures 4 and 5 show the qualitative results of the estimated 3D pose. In Fig. 4, the left and right figures w.r.t. the center dotted line, the reconstructed 3D pose show the effect of the multi-stage architecture. As the number of stages increases, the estimated 3D pose becomes closer to the ground truth. This multi-stage structure is very quantitatively and qualitatively effective in the 3D pose estimation. In Fig. 5, the estimated 3D pose of real-world image using the proposed model trained with Human3.6M shows qualitatively satisfactory results.

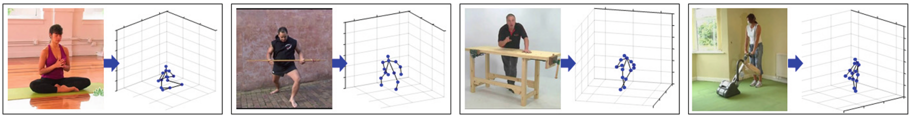

Fig. 5. Qualitative results of real-world image. (Best seen in zoom.)

Effect of Temporal Correlation in Pose Estimation: Estimating 3D pose using single frame 2D image or pose only has a limitation. If there are self-occluded cases in human pose, it would be difficult for even human to guess the pose correctly, which significantly degrades the estimation performance. On the other hand, if multiple frame images or poses are utilized, it should be much easier to handle the self-occluded issue. To reduce these errors, some authors in [18,26,44] used sequential frames as inputs to their methods to learn temporal correlation. Inspired by [26], we used sequential frames as inputs to the proposed method. In Table 4, rows 8 to 10 provide the performance according to the number of input frames. The result on row 9 is the best performance with

3 input frames. The overall performance is worst when 10 frames are used as inputs, and the best performance is shown only in the walking scenario when 5 frames are used as inputs. The performance in the walking scenario consisting of simple repetitive actions can be improved by using more frames. The results show that using a proper quantity of frames contributes to improving the performance, but using excessive frames degrades it.

How to Make the Pose Depth Cue: In the pose depth cue of Sect. 3.2, we have described the type of the pose depth cue. In Table 5, rows 2 to 4 show the performance of each joint according to the type of the pose depth cue. The elimination and addition method ($D = 1$) implies the pure connected p-LSTMs which do not use auxiliary data. The second type is created using the concatenation method, and the last type is created using the replacement method. The results show that the third type has the best performance and the first type has the worst performance. For the third type of the pose depth cue, when some 2D human pose is replaced by some expected 3D human pose, the vector of input pose depth cues will have a constant size even though they pass through p-LSTMs. This type also includes a 2D pose remaining as ancillary data. On the other hand, the first type of the pose depth cue does not use auxiliary data. This result shows the performance of p-LSTMs in a purely connected structure. This ablation study shows that the auxiliary data brings approximately 36.3% performance improvement to the proposed method. The pose depth cue of the third type is very effective in learning the spatial correlation of human poses.

How to Set the Propagating Direction: The pose depth cue is created through a depth fusion layer of a p-LSTM. The created pose depth cue is propagated sequentially to a number of connected p-LSTMs. From the propagation point of view, directions are determined after the initial seed joints are generated. Thus, the direction of the propagating pose depth cue can be divided into outward and inward directions. The outward direction is to propagate the pose depth cue from the centroid part to the edge of the body outward. On the contrary, the inward direction is a method of propagating the pose depth cue from the edge to the centroid part of the body inward. The results of the experiment are explained by the fourth and fifth row results in Table 5. The method of the outward direction is superior in performance. Since the pose depth cue is made up of a combination of some estimated 3D and 2D poses, the 3D pose estimated at the body center delivers more stable pose depth cues.

5 Conclusion

In this study, we have proposed a novel 3D pose estimation method, the p-LSTMs, where 9 LSTM networks are sequentially connected, in order to reflect the spatial information about the JI and the temporal information about the input image frames. In addition, we have defined a depth cue for the pose, and propagated this information across multiple LSTM networks. Through an ablative study, we have proved the validity of the proposed techniques such as propagating direction, pose depth cue, and multi-stage architecture. The proposed method have achieved significant improvement compared with the state-of-the-art methods on two public datasets.

In the future, we plan to investigate failure cases for further improvement. A possible direction will be to weight a frame with a reliability factor when using multiple input frames. Another direction is to adjust the parameters of the proposed method to obtain more accurate poses. Finally, we hope that our approach can provide insight for other research on 3D multiple-human, 3D object and 3D hand pose estimations.

Acknowledgement. This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00406, SIAT CCTV Cloud Platform).

References

1. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–4732 (2016)
2. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
3. Pishchulin, L., et al.: DeepCut: joint subset partition and labeling for multi person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4929–4937 (2016)
4. Zhou, X., Zhu, M., Leonardos, S., Derpanis, K.G., Daniilidis, K.: Sparseness meets deepness: 3D human pose estimation from monocular video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4966–4975 (2016)
5. Zhou, X., Sun, X., Zhang, W., Liang, S., Wei, Y.: Deep kinematic pose regression. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 186–201. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_17
6. Popa, A.I., Zanfir, M., Sminchisescu, C.: Deep multitask architecture for integrated 2D and 3D human sensing. In: Conference on Computer Vision and Pattern Recognition, vol. 1, p. 5 (2017)
7. Kim, J., Lee, I., Kim, J., Lee, S.: Implementation of an omnidirectional human motion capture system using multiple kinect sensors. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **98**(9), 2004–2008 (2015)
8. Kwon, B., et al.: Implementation of human action recognition system using multiple kinect sensors. In: Ho, Y.-S., Sang, J., Ro, Y.M., Kim, J., Wu, F. (eds.) PCM 2015. LNCS, vol. 9314, pp. 334–343. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24075-6_32
9. Kwon, B., Kim, J., Lee, K., Lee, Y.K., Park, S., Lee, S.: Implementation of a virtual training simulator based on 360 multi-view human action recognition. IEEE Access **5**, 12496–12511 (2017)
10. Meng, M., et al.: Kinect for interactive AR anatomy learning. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 277–278. IEEE (2013)

11. González-Ortega, D., Díaz-Pernas, F., Martínez-Zarzuela, M., Antón-Rodríguez, M.: A kinect-based system for cognitive rehabilitation exercises monitoring. Comput. Methods Programs Biomed. **113**(2), 620–631 (2014)
12. Tong, J., Zhou, J., Liu, L., Pan, Z., Yan, H.: Scanning 3D full human bodies using kinects. IEEE Trans. Vis. Comput. Graph. **18**(4), 643–650 (2012)
13. Agarwal, A., Triggs, B.: 3D human pose from silhouettes by relevance vector regression. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, p. II. IEEE (2004)
14. Mori, G., Malik, J.: Recovering 3D human body configurations using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **28**(7), 1052–1062 (2006)
15. Bo, L., Sminchisescu, C., Kanaujia, A., Metaxas, D.: Fast algorithms for large scale conditional 3D prediction. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
16. Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., Torr, P.H.: Randomized trees for human pose detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
17. Li, S., Chan, A.B.: 3D human pose estimation from monocular images with deep convolutional neural network. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9004, pp. 332–347. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16808-1_23
18. Grinciunaite, A., Gudi, A., Tasli, E., den Uyl, M.: Human pose estimation in space and time using 3D CNN. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 32–39. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_5
19. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3D human pose. In: Computer Vision and Pattern Recognition (CVPR) (2017)
20. Sigal, L., Balan, A.O., Black, M.J.: HumanEva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. Int. J. Comput. Vis. **87**(1), 4–27 (2010)
21. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6M: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
22. Chen, C.H., Ramanan, D.: 3D human pose estimation = 2D pose estimation + matching. In: CVPR, vol. 2, p. 6 (2017)
23. Yasin, H., Iqbal, U., Kruger, B., Weber, A., Gall, J.: A dual-source approach for 3D pose estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4948–4956 (2016)
24. Tome, D., Russell, C., Agapito, L.: Lifting from the deep: convolutional 3D pose estimation from a single image. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
25. Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., Fua, P.: Structured prediction of 3D human pose with deep neural networks. In: Richard, C. Wilson, E.R.H., Smith, W.A.P. (eds.) Proceedings of the British Machine Vision Conference (BMVC), pp. 130.1–130.11. BMVA Press, September 2016)
26. Lin, M., Lin, L., Liang, X., Wang, K., Chen, H.: Recurrent 3D pose sequence machines. In: CVPR (2017)
27. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3D human pose estimation. In: IEEE International Conference on Computer Vision, vol. 206, p. 3 (2017)

28. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Towards 3D human pose estimation in the wild: a weakly-supervised approach. In: IEEE International Conference on Computer Vision (2017)
29. Moreno-Noguer, F.: 3D human pose estimation from a single image via distance matrix regression. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1561–1570. IEEE (2017)
30. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it SMPL: automatic estimation of 3D human pose and shape from a single image. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 561–578. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_34
31. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. In: The IEEE International Conference on Computer Vision (ICCV), vol. 2 (2017)
32. Westoby, M., Brasington, J., Glasser, N., Hambrey, M., Reynolds, J.: Structure-from-motionphotogrammetry: a low-cost, effective tool for geoscience applications. Geomorphology **179**, 300–314 (2012)
33. Lee, S.H., Kang, J., Lee, S.: Enhanced particle-filtering framework for vessel segmentation and tracking. Comput. Methods Programs Biomed. **148**, 99–112 (2017)
34. Oh, H., Kim, J., Kim, J., Kim, T., Lee, S., Bovik, A.C.: Enhancement of visual comfort and sense of presence on stereoscopic 3D images. IEEE Trans. Image Process. **26**(8), 3789–3801 (2017)
35. Lee, K., Lee, S.: A new framework for measuring 2D and 3D visual information in terms of entropy. IEEE Trans. Circuits Syst. Video Technol. **26**(11), 2015–2027 (2016)
36. Oh, H., Lee, S., Bovik, A.C.: Stereoscopic 3D visual discomfort prediction: a dynamic accommodation and vergence interaction model. IEEE Trans. Image Process. **25**(2), 615–629 (2016)
37. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. ACM Trans. Graph. (TOG) **34**(6), 248 (2015)
38. Lee, I., Kim, D., Kang, S., Lee, S.: Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1012–1020. IEEE (2017)
39. Toshev, A., Szegedy, C.: DeepPose: human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653–1660 (2014)
40. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
41. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. OSDI **16**, 265–283 (2016)
42. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: new benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, pp. 3686–3693 (2014)
43. Rogez, G., Schmid, C.: Mocap-guided data augmentation for 3D pose estimation in the wild. In: Advances in Neural Information Processing Systems, pp. 3108–3116 (2016)
44. Hossain, M.R.I.: Understanding the sources of error for 3D human pose estimation from monocular images and videos. Ph.D. thesis, University of British Columbia (2017)
45. Gower, J.C.: Generalized procrustes analysis. Psychometrika **40**(1), 33–51 (1975)
46. Radwan, I., Dhall, A., Goecke, R.: Monocular image 3D human pose estimation under self-occlusion. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1888–1895 (2013)

47. Simo-Serra, E., Quattoni, A., Torras, C., Moreno-Noguer, F.: A joint model for 2D and 3D pose estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3634–3641 (2013)
48. Kostrikov, I., Gall, J.: Depth sweep regression forests for estimating 3D human pose from images. In: BMVC, vol. 1, p. 5 (2014)
49. Tekin, B., Rozantsev, A., Lepetit, V., Fua, P.: Direct prediction of 3D body poses from motion compensated sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 991–1000 (2016)