Occlusion-Aware Networks for 3D Human Pose Estimation in Video

Yu Cheng*[1], Bo Yang*[2], Bo Wang[2], Wending Yan[1], and Robby T. Tan[1,3]

[1]National University of Singapore
[2]Tencent Game AI Research Center
[3]Yale-NUS College
{e0321276,e0267911}@u.nus.edu, {brandonyang,bohawkwang}@tencent.com, robby.tan@nus.edu.sg

Abstract

Occlusion is a key problem in 3D human pose estimation from a monocular video. To address this problem, we introduce an occlusion-aware deep-learning framework. By employing estimated 2D confidence heatmaps of keypoints and an optical-flow consistency constraint, we filter out the unreliable estimations of occluded keypoints. When occlusion occurs, we have incomplete 2D keypoints and feed them to our 2D and 3D temporal convolutional networks (2D and 3D TCNs) that enforce temporal smoothness to produce a complete 3D pose. By using incomplete 2D keypoints, instead of complete but incorrect ones, our networks are less affected by the error-prone estimations of occluded keypoints. Training the occlusion-aware 3D TCN requires pairs of a 3D pose and a 2D pose with occlusion labels. As no such a dataset is available, we introduce a "Cylinder Man Model" to approximate the occupation of body parts in 3D space. By projecting the model onto a 2D plane in different viewing angles, we obtain and label the occluded keypoints, providing us plenty of training data. In addition, we use this model to create a pose regularization constraint, preferring the 2D estimations of unreliable keypoints to be occluded. Our method outperforms state-of-the-art methods on Human 3.6M and HumanEva-I datasets.

1. Introduction

Estimating 3D human poses from a monocular video is important in many applications, such as animation generation, activity recognition, human-computer interaction, and etc. Recent top-down pose estimation methods have achieved promising results [29, 13, 27, 6, 21, 28, 36]. Generally, these methods detect individual persons in each image, estimate the 2D pose within each person bounding box, and finally convert the 2D pose to a 3D pose. As humans are articulated objects, many joints or keypoints, such as those at wrist, elbow, and foot, can be invisible due to occlusion,

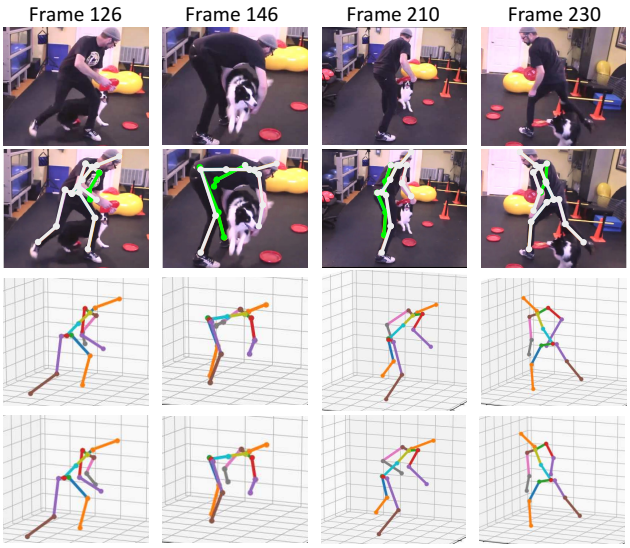

Figure 1. Comparison between 3D pose estimation results using occlusion awareness (fourth row) or not (third row). The 2D pose estimation results are shown in the second row, and the occluded and unoccluded human joints are labeled as green and white respectively (best viewed in color).

as shown in the second row in Figure 1. These methods always predict 2D locations of all keypoints belonging to a person, even though some of them are invisible or occluded. This is risky, since consequently the 3D pose estimation becomes vulnerable to error.

Researchers have shown that occlusion is a major source of errors of human pose estimation from a single image [25, 24, 39, 23], and state-of-the-art methods [34, 42, 8] still suffer from it. There have been attempts to estimate occlusion likelihoods of keypoints or body parts from an image [32, 11], penalize occluded keypoints [1], or infer multiple 3D poses [44, 18]. For video input, to address the problem, temporal information are utilized [16, 13, 17, 29]. However, they are based on an assumption that occlusion

*Both authors contributed equally.

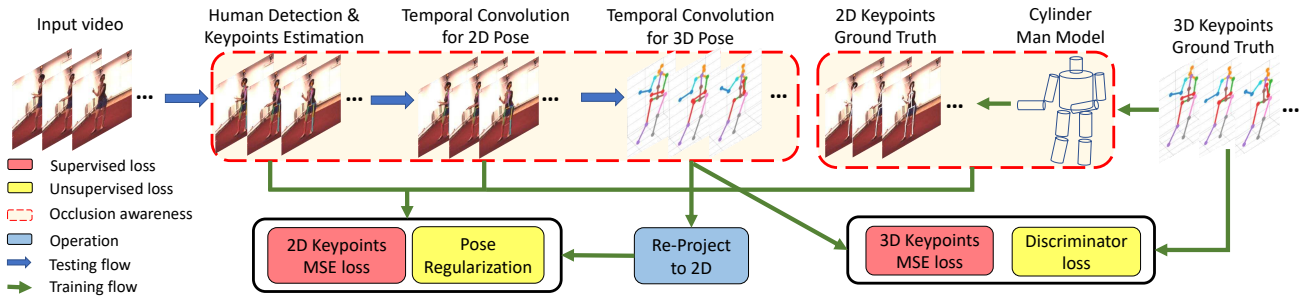

Figure 2. The framework of our approach, best viewed in color.

happens only in few independent frames. Unfortunately, in real cases, occlusion can occur persistently across multiple frames. Hence, the error is difficult to be corrected by simply doing occlusion-unaware temporal convolution as shown in the third row in Figure 1, since they have no knowledge about which keypoints may be unreliable and treat all keypoints equally.

In this paper, our goal is to estimate 3D human poses from a single video. We introduce an occlusion-aware deep-learning framework, consisting three networks, to deal explicitly with occluded keypoints, as shown in Figure 2. The first network outputs the estimated 2D locations of the keypoints for each bounding box of a person in the form of heatmaps (confidence maps) frame by frame independently. These maps are combined with optical flow to assess whether the predicted keypoints are occluded. Existing temporal based methods [13, 17, 29] use all keypoints despite some of them are inaccurate due to occlusion. In contrast, we filter out occluded ones, and then feed the possibly incomplete 2D keypoints to our second and third networks, which are both temporal convolutional networks (2D and 3D TCNs, respectively), to enforce temporal smoothness.

As our 3D TCN takes possibly incomplete 2D keypoints as input, we need pairs of a 3D pose and a 2D pose with occlusion labels during training. However, in most 3D human datasets, there are no occlusion labels available. Therefore, we introduce a "Cylinder Man Model" that enables us to project a 3D human pose onto virtual 2D planes in different viewing angles. Thus, we can obtain and label keypoints that are occluded by the person himself, providing plenty of training data.

Due to limited 3D joint ground-truths, a few recent methods utilized 2D pose data to train their 3D pose network by employing a 3D-to-2D projection loss [26, 29]. However, these methods simply ignore occluded keypoints when computing the loss, leading to possible erroneous solutions, since those keypoints may be estimated as unoccluded, contradicting the occlusion ground-truth labels. Therefore, we introduce a pose regularization term to penalize such violations, with the help of our "Cylinder Man Model".

Our whole framework can be trained end-to-end in a semi-supervised way. The fourth row in Figure 1 shows the effectiveness of our method under a long-term 2D occlusion scenario; the occluded arms could be correctly estimated. As a summary, our contributions are as follows:

- We introduce a 3D pose estimation framework with explicit occlusion handling.
- We propose a novel "Cylinder Man Model" for automatic data augmentation of paired 3D pose and occluded 2D pose, and for pose regularization of occluded keypoints.
- We introduce a fully integrated framework of 2D pose and 3D pose estimations that can be trained end-to-end and in a semi-supervised way.

2. Related Work

In recent years, while deep learning-based 2D human pose estimation methods have showed significant progress [38, 37, 25, 40, 4, 3], 3D pose estimation remains challenging mainly due to occlusion and depth ambiguity. Some methods use the camera array system (a set of RGB and depth sensors) to track accurate 3D body motion [9, 15]. Due to the high demand of pose estimation for wild videos, many recent methods focused on data captured by monocular RGB cameras [2, 45].

Getting accurate and reliable 3D joints from a single image is intractable [17, 13, 29]. Recently, temporal information is used to provide reliable 3D estimation. Lee et al. [17] use LSTM to learn joint inter-dependency for 3D human pose estimation. Hossain et al. [13] use an RNN model to enforce the motion smoothness. However they assume high frame rate and slow motion, which limit the effectiveness of the method in wild videos. Pavllo et al. [29] propose a temporal-convolution method to generate 3D poses from 2D keypoint sequences. However, they require the estimation of all 2D keypoints in every frame and assume prediction errors are temporally non-continuous and independent, which do not hold in most of the occlusion cases. Charles et al. [5] also detected the occluded keypoints and removed them for temporal propagation; however, we adopted TCN to achieve larger temporal perceptive field than their optical flow approach.

As human pose datasets with 3D joint ground-truth are rare, to avoid overfitting, a few methods adopt a semi-supervised approach, which usually projects the estimated 3D joints onto the 2D image space and compare the results with 2D ground-truths for loss computation [26, 29]. This allows the use of 2D datasets without 3D joint ground-truths for training. However, none of them take the missing (occluded) keypoints into account, causing their networks to learn inaccurately. A few methods tackle the occlusion problem by regularizing the spatial configuration [8, 7], or performing adversarial data augmentation to improve the detection accuracy for occluded cases [30]. Unfortunately, none of these utilize temporal information, making the prediction unstable.

Unlike existing temporal based methods [26, 13, 29], our temporal convolutional networks explicitly exclude occluded and thus unreliable keypoint predictions. Moreover, we introduce a novel "Cylinder Man Model" to generate pairs of virtual 3D joints and 2D keypoints with explicit occlusion labels, that are essential for training our networks. In addition, instead of ignoring occluded keypoints, we design a pose regularization scheme by adding occlusion constraints in the loss function.

3. Occlusion-Aware 3D Pose Estimation

Figure 2 shows an overview of our framework. Given an input video, we apply a human detector, such as Mask R-CNN [12], to each frame, normalize each detected human bounding box to a fixed size while keeping the width/height ratio, and feed it to our first network, a stacked hourglass network [25], which estimates the 2D keypoints in the form of heatmaps (or confidence maps). Subsequently, our second network (2D TCN) improves the accuracy of the estimated 2D keypoints, and feed them further to our third network (3D TCN) to obtain the final 3D pose. Our framework is end-to-end, for both training and testing.

If there are multiple persons in the input video, we employ PoseFlow Tracker [41] to avoid identity shift. We assume that the scene is not too crowded, so that the tracker is less likely to cause identity switch. Tracking multiple persons in crowded scenes under various poses is a complex problem, which is beyond the scope of this paper.

3.1. Two-Dimensional Pose Estimation

Given a bounding box containing a person, our first network outputs a set of heatmaps, expressed as $\{\tilde{M}_i\}$, where $i \in [1, K]$ and K is the number of predefined keypoints. The network processes the bounding box frame-by-frame individually, and is trained using the following loss:

$$L_{2D}^{S} = \sum_{i=1}^{K} ||M_i - \tilde{M}_i||_2^2, \tag{1}$$

where M_i is the ground-truth heatmap for keypoint i, and is defined as all zero for occluded keypoints and a single peak with Gaussian smoothness for non-occluded ones as in [25]. A sigmoid function is used in the output layer to force each value in the heatmaps within the range of $[0, 1]$. The values represent the confidence scores of the keypoint estimation.

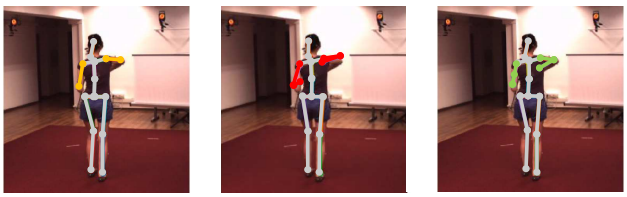

Figure 3. Comparison of final 3D results between filling occluded keypoints in 2D TCN (middle) and 3D TCN (right). The left image shows the initial incomplete 2D estimation. We highlight the joints whose estimations are different in the second and third columns for clear visualization.

For each heatmap $\tilde{M}_i$, we choose the peak response point $\tilde{p}_i$ with a confidence score of C_i as a candidate for the ith keypoint. Our method is expected to produce low C_i for occluded keypoints. To further improve the occlusion estimation, we apply optical flow (e.g., [33]) to $\tilde{p}_i$, and record the flow vector as $\vec{o_i}$. Our first network also processes the next frame, and the location difference of keypoint i in the neighboring frames is defined as $\vec{d_i}$. The difference between $\vec{o_i}$ and $\vec{d_i}$ is further used to measure the reliability of $\tilde{p}_i$. Hence, the final confidence score for $\tilde{p}_i$ is defined as:

$$C_i^* = C_i \exp(-\frac{||\vec{o_i} - \vec{d_i}||_2^2}{2\sigma^2}), \tag{2}$$

where σ is a standard deviation and is fixed to 0.1 in our case. If C_i^* is smaller than a threshold b, $\tilde{p}_i$ is labeled as an occluded keypoint.

To exploit temporal smoothness, we concatenate the coordinates of all 2D keypoints to form a $2K$ long vector, $\tilde{X}$, and feed all such vectors in the temporal window to a 2D dilated temporal convolutional network (2D TCN), $f(\cdot)$. Unlike [29], we remove the occluded keypoints by setting both their values in the vector and ground-truths to zero. The loss of the 2D TCN is formulated as:

$$L_{2D}^{T} = ||C_b^T(f(C_b^T\tilde{X}) - X)||_2^2, \tag{3}$$

where X is the concatenated ground-truth keypoint coordinate vector, and C_b is the binarized confidence score vector according to a threshold b, indicating the reliability labels of keypoints.

Note that, in our method, we do not intend to complete the missing keypoints in the 2D TCN. Our experiments in Table 2 show that leaving the prediction of the missing (occluded) keypoints to the 3D TCN provides better performance. The reason is that the temporal smoothness in 3D is more stable than in 2D where distortions can occur. Figure 3 shows an example. We see that filling missing keypoints in 2D TCN may lead to inaccurate localization of keypoints, while 3D TCN produces more precise estimations.

3.2. Three-Dimensional Pose Estimation

Having obtained the temporally smoothed yet possibly incomplete 2D keypoints, we feed them into our 3D TCN, which outputs the estimated 3D joint coordinates for all keypoints, represented as $\{\tilde{P}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)\}$, including those predicted as occluded keypoints in the early stage.

When the 3D joint ground-truths are available, our 3D TCN employs the MSE loss based on 3D joints expressed as:

$$L_{MSE} = \sum_i ||\tilde{P}_i - P_i||_2^2, \tag{4}$$

where P_i is the 3D joint ground-truth, and $\tilde{P}_i$ is the corresponding predicted 3D joint by the 3D TCN. When 3D ground-truths are not available, we project the results back to 2D assuming orthogonal projection, and calculate the loss as:

$$L_{proj} = \sum_i v_i ||p_i - \tilde{p}_i||_2^2, \tag{5}$$

where $p_i = (x_i, y_i)$ is the ground-truth of a 2D keypoint, and $\tilde{p}_i = (\tilde{x}_i, \tilde{y}_i)$ is a keypoint resulted from the projection of the corresponding 3D joint. $v_i \in \{0, 1\}$ is the occlusion label of keypoint i.

In addition, we also add a symmetry constraint to limit the bone lengths to be the same between a person's left and right part, and is defined as $L_{Sym} = \sum_{(i,j)\in E}(||\tilde{P}_i - \tilde{P}_j||_2 - ||\tilde{P}_{\hat{i}} - \tilde{P}_{\hat{j}}||_2)^2$, where E is the set of all neighboring keypoints forming a bone and $\hat{i}$ indicates the index of the symmetric part of keypoint i.

As human joints have several constraints, only part of the poses in the whole 3D pose space are anthropometrically valid. Similar to [43, 7], we also adopt the concept of adversarial learning. A discriminator is trained to assess the correctness of the estimated 3D joints by minimizing a loss function as $L_{dis} = -\sum_j (u_j \log q_j + (1-u_j)\log(1-q_j))$, where j is the index of a 3D pose, u_j is 1 and 0 for ground truth and generated 3D poses respectively, and $q_j \in [0, 1]$ is the output of the discriminator network. The loss for the 3D pose estimation module is then defined as:

$$\begin{aligned} L_{3D} =& \zeta \sum_i ||P_i - \tilde{P}_i||_2^2 + (1-\zeta) \sum_i v_i ||p_i - \tilde{p}_i||_2^2 \\ &+ \alpha L_{Sym} + \beta L_{Dis}, \end{aligned} \tag{6}$$

where $\zeta \in \{0, 1\}$ indicates whether the 3D ground-truth is available, and α and β are weighting factors to balance the influences of symmetric loss L_{Sym} and discriminative loss L_{Dis}, and are fixed to 0.2 and 0.1 in our experiments.

4. Cylinder Man Model

Training the 3D TCN requires pairs of a 3D joint ground-truth and a 2D keypoint with occlusion label. However, the existing 3D human pose datasets (e.g., [14, 22]) have no occlusion labels, and the amount of the 3D data is limited. Hence, we introduce a "Cylinder Man Model" to generate occlusion labels for 3D data and perform data augmentation. We also use the model for pose regularization of occluded keypoints when 3D ground-truths are unavailable.

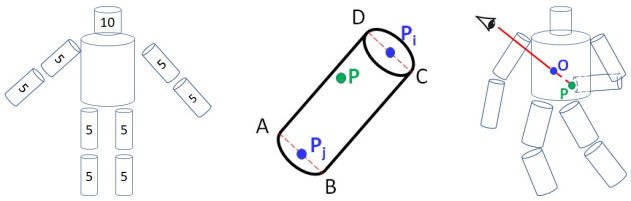

Figure 4. Illustration for the "Cylinder Man Model" for occlusion reasoning. See text for details.

4.1. Model Definition

As shown in the left of Figure 4, we divide a 3D human into ten parts: head, torso, two upper arms, two lower arms, two thighs, and two lower legs. Given any 3D skeleton, either from the ground-truth or our networks, we use a cylinder to approximate the 3D shape of each of the ten parts. The radius of the head is defined as $10cm$, and the radius of each limb is defined as $5cm$ as labeled in Figure 4 left. The height of a cylinder is defined as the distance between keypoints defining that part. The radius of the torso is not pre-defined but is set to the distance between the neck and the shoulder. Such approximation works well in our framework, and is validated by the experiments.

In our model, each cylinder is formulated by $C_{ij} = \{r_{ij}, P_i, P_j\}$, where r_{ij} is the radius and P_i, P_j are the 3D joints defining the centers of the top and bottom parts of the cylinder as visualized in the middle of Figure 4.

To calculate whether a point P is occluded by C_{ij}, we first map them to the 2D space assuming orthogonal projection. The vertical cross section of the cylinder, $ABCD$, maps to a rectangle $A'B'C'D'$ as shown in the middle of Figure 4. As r_{ij} is small with respect to the height of the cylinder, i.e., the length of a bone, we only check whether P is occluded by $ABCD$ when projecting to the 2D plane. If the projected P is not inside the rectangle $A'B'C'D'$ in 2D space, it is not occluded. Otherwise, we calculate the norm of the plane $ABCD$ in 3D space as $\overrightarrow{n_{ij}} = \overrightarrow{P_jP_j} \times \overrightarrow{P_jA}$. Note that, $-\overrightarrow{n_{ij}}$ is also the norm vector. We choose the one pointing toward the camera, i.e., z coordinate is negative.

The visibility of point P is then calculated by

$$V_P = \prod_{(i,j)\in E} [(P - P_i) \cdot n_{ij} > 0], \tag{7}$$

where E is the set of all neighboring keypoints forming a bone, and $[\cdot]$ denotes the Iverson bracket, returning 1 if the proposition is true, and 0 otherwise. In order to assure differentiable, we use sigmoid function to approximate this operation. In the example shown in Figure 4 right, if the view angle is from behind the person, the keypoint P will be occluded by the body cylinder at point O.

Other human body model like SMPL [19] could provide more detailed representation of human shape, but it requires extra computational cost for checking occlusion. Cylinder based approximation is suitable for our tasks.

4.2. Pose Data Augmentation

Existing 3D datasets provide 3D coordinates of human joints and captured 2D images in different view angles. To provide occlusion ground-truths, we first expand the 3D skeleton to a "Cylinder Man" using the above model. Based on the provided camera parameters, we can estimate the current camera's viewing angle in the world coordinate. Thus, we could predict the visibility of each keypoint in the corresponding image using Eq. 7.

While the above process creates some data for training our 3D TCN, the data are still insufficient due to limited number of captured images. Hence, we create a set of virtual cameras around the humans in a 3D dataset to increase our training data. We normalize the 3D skeleton ground-truth with respect to the body center, and thus we could ignore the camera translation but only consider the rotation operation. The rotation angles around x and z axes are limited to 0.2π with a sampling step of 0.02π to avoid reversing a human upside down. The rotation around the y axis is randomly selected within $[-\pi, \pi]$. Thus, we generate 100 virtual view angles for every sample, and use Eq. 7 to estimate the occlusion of each keypoint to produce pairs of 3D pose and 2D pose with occlusion labels. As our Cylinder Man Model only calculates self-occlusions, to further include inter-object occlusion cases, we randomly mask out some keypoints assuming that they are occluded by some virtual occluders. These additional data obviously improve the diversity of our training set.

4.3. Pose Regularization

In our framework, occluded keypoints are filtered out before feeding the keypoints to our 3D TCN. This means that we estimate the 3D joints of the missing keypoints from the other reliable ones. However, there are many possible paths in the 3D space that could fill the missing keypoints. One example is shown in the second row of Figure 5. Surely, when 3D joint ground-truths are available, we can use them to train the 3D TCN to estimate the correct path. However, we do not always have 3D joint ground-truths as mentioned before. Hence, we introduce a pose regularization constraint.

Given an estimated 3D pose, we first build its "Cylinder Man Model", and then calculate the visibility label $\tilde{v}_i \in \{0, 1\}$ for each estimated keypoint $\tilde{p}_i$ in the 2D space using Eq. 7. If a missing keypoint is occluded, we have a reasonable explanation for its failure of detection or unreliability. If it is not occluded, it is less likely to be missed by the 2D keypoint estimator and should be penalized as:

$$L_{reg} = (1 - \zeta) \sum_{i \in Occ} \tilde{v}_i, \tag{8}$$

where Occ is the set of unreliable keypoints, predicted by the method in Section 3.1. With this regularization term, we prefer to find a 3D pose configuration where the unreliable keypoints are occluded. An example is shown in Figure 5. The left wrist of the person is occluded in Frame 120 & 190, and is classified as an unreliable keypoint by Equ. 2. Without the pose regularization constraint, the estimated locations of the left wrist are not occluded as shown in the second row in Figure 5. After adding our proposed regularization, the framework pushes the unreliable keypoints to be occluded, producing correct results as shown in the last row in Figure 5.

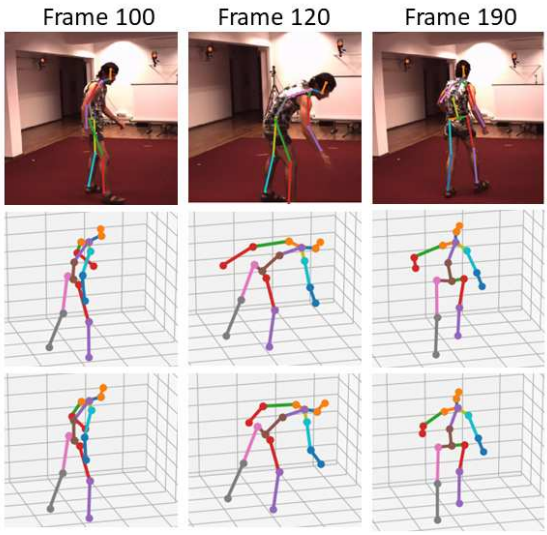

Figure 5. Example of the effectiveness of pose regularization. The second row shows wrong 3D estimation without the regularization term. The third row shows that the constraint fixes the error.

Our whole system is trained end to end by minimizing the loss as:

$$L = L_{2D} + w_1 L_{3D} + w_2 L_{reg}, \tag{9}$$

where w_1 and w_2 are weighting factors, and are fixed to 1.0 and 0.1, respectively .

5. Experiments

5.1. Experimental Settings

Datasets. Two widely used human pose estimation datasets, Human3.6M [14] and HumanEva-I [31], are used for performance evaluation.

Human3.6M is a large 3D human pose dataset. It has 3.6 million images including eleven actors performing daily-life activities, and seven actors are annotated. The 3D ground-truth is provided by the Mocap system and intrinsic and extrinsic camera parameters are known. Similar to previous work [13, 29, 27, 43], we use subjects 1, 5, 6, 7, 8 for training, and the subjects 9 and 11 for evaluation.

HumanEva-I is a relatively smaller dataset. Following the typical protocol [21, 13, 29], we use the same data division to train one model for all three actions (Walk, Jog, Box), and use the remaining data for testing.

Evaluation protocols. We use two common evaluation protocols in our experiments. *Protocol #1* refers to the Mean Per Joint Position Error (MPJPE) which is the mil-

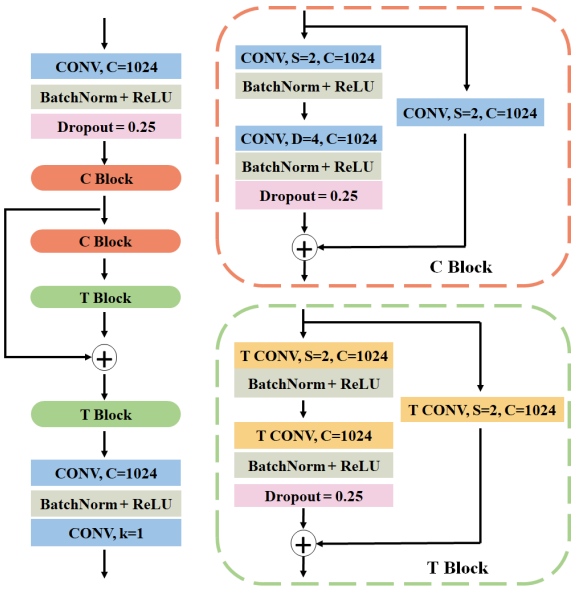

Figure 6. Illustration of our temporal convolutional network structure. *CONV* and *T CONV* stand for convolutional and transpose convolutional operations. S, C, and D stand for stride, channel and dilation rate, respectively. The kernel size for all blocks are set to 5.

limeters between the ground-truth and the predicted keypoints. *Protocol #2*, often called P-MPJPE, refers to the same error after applying alignment between the predicted keypoints and the ground-truth.

5.2. Implementation Details

We adopt Mask-RCNN [12] for human detection and use a ResNet-101 backbone. The Stacked Hourglass Network [25] is used as 2D pose detector structure and is initialized with weights pre-trained on COCO dataset.

We use the same network structures for our two TCNs, each of which has two convolutional blocks (C Blocks) and two transposed convolutional blocks (T Blocks) as shown in Figure. 6. Short connections are used to incorporate different time scale and dilation. The structure for 2D TCN and 3D TCN are the same except the output channel of the last convolutional layer. The discriminator for checking the validation of a 3D pose is composed of three 1D convolutional layers followed by one fully connected layer which outputs the final discrimination score.

We use Adam Optimizer with a learning rate of 0.001 for the first 100,000 iterations, and 0.0001 for another 30,000 iterations. We use a batch-size of 128 and perform random data augmentation as mentioned in Section 4.2.

5.3. Hyper-Parameter Sensitivity Analysis

There are two important hyper-parameters in our framework: the sequence length for our TCNs and the threshold for reliable keypoints. We test the performance on Human3.6M dataset, *Protocol #1* and *#2* for comparison. The

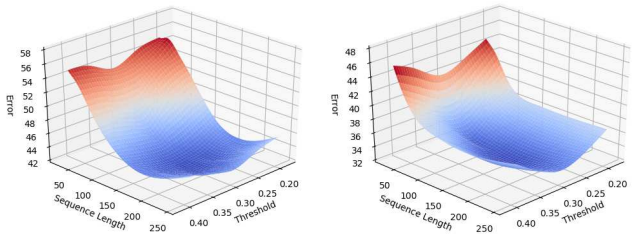

Figure 7. The estimation error (in mm) using *Protocol #1* and *#2* under different hyper-parameter settings.

Method	*Protocol #1*	*Protocol #2*
Seq=16, t=0.30	55.4	41.2
Seq=32, t=0.30	51.8	38.1
Seq=64, t=0.30	47.0	34.6
Seq=128, t=0.30	42.9	32.8
Seq=256, t=0.30	44.1	34.0
Seq=128, t=0.20	45.7	36.1
Seq=128, t=0.25	43.3	34.8
Seq=128, t=0.30	42.9	32.8
Seq=128, t=0.35	43.1	34.1
Seq=128, t=0.40	44.2	35.7

Table 1. Hyper-parameter sensitivity testing based on estimation errors on Human 3.6M under *Protocol #1* and *#2*.

3D TCN	2D TCN	Occ Aware	Sym	Adv	Pos Reg	Data Aug	P #1	P #2
✓							54.0	42.1
✓	✓						51.7	40.5
✓	✓	✓					46.3	35.4
✓	✓	✓	✓				45.8	34.8
✓	✓	✓	✓	✓			45.1	34.3
✓	✓	✓	✓	✓	✓		44.8	34.1
✓	✓	✓	✓	✓	✓	✓	42.9	32.8

Table 2. The effectiveness of different components: 3D TCN, 2D TCN, Occlusion Awareness, Symmetry Constraint, Adversarial Learning, Pose Regularization, Data Augmentation. We conduct the evaluation based on Human 3.6M *Protocol #1* (P #1) and *Protocol #2* (P #2).

models are tested under different settings, and the results are shown in Figure 7 and Table 1.

We find that under each protocol, there is an obvious valley in the error surface in Figure 7. The curvature around the valley is small, and even the second and the third best settings in Table 1 still outperform the state-of-the-art results, indicating that our approach is not sensitive to these hyper-parameters. The error drops with the increase of sequence length until 256. This means that more temporal information would benefit the pose estimation, but temporally far away poses may not provide much useful information and the over-length duplicate padding at sequence boundaries may be detrimental to the performance. Moreover, the error reaches the valley around threshold value of 0.3. Too small threshold weakens the effectiveness of suppression of unreliable keypoints, and more erroneous keypoints will be used in the later TCN modules; too large threshold leads to excessively information removal, leaving little useful information for estimation of all keypoints. In later experiments,

Method	Direct	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Pavlakos et al. [28] CVPR'17	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
Zhou et al. [46] ICCV'17	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.1	66.0	51.4	63.2	55.3	64.9
Martinez et al. [21] ICCV'17	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. [35] ICCV'17	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. [10] AAAI'18	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
Yang et al. [43] CVPR'18	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. [27] CVPR'18	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
Luvizon et al. [20] CVPR'18	49.2	51.6	47.6	50.5	51.8	60.3	48.5	51.7	61.5	70.9	53.7	48.9	57.9	44.4	48.9	53.2
Lee et al. [17] ECCV'18	40.2	49.2	47.8	52.6	50.1	75.0	50.2	43.0	55.8	73.9	54.1	55.6	58.2	43.3	43.3	52.8
Hossain & Little [13] ECCV'18	44.2	46.7	52.3	49.3	59.9	59.4	47.5	46.2	59.9	65.6	55.8	50.4	52.3	43.5	45.1	51.9
Pavllo et al. [29] CVPR'19	45.2	46.7	**43.3**	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	**44.0**	49.0	**32.8**	33.9	46.8
Our result	**38.3**	**41.3**	46.1	**40.1**	**41.6**	**51.9**	**41.8**	**40.9**	**51.5**	**58.4**	**42.2**	44.6	**41.7**	33.7	**30.1**	**42.9**

Table 3. Quantitative evaluation using MPJPE in millimeter between estimated pose and the ground-truth on Human3.6M under *Protocol #1*, no rigid alignment or transform applied in post-processing. Best in bold, second best underlined.

Method	Direct	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Moreno-Noguer et al. [23] CVPR'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. [28] CVPR'17	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	51.9
Martinez et al. [21] ICCV'17	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Sun et al. [35] ICCV'17	42.1	44.3	45.0	45.4	51.5	53.0	43.2	41.3	59.3	73.3	51.0	44.0	48.0	38.3	44.8	48.3
Fang et al. [10] AAAI'18	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. [27] CVPR'18	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
Yang et al. [43] CVPR'18	**26.9**	30.9	36.3	39.9	43.9	47.4	**28.8**	29.4	**36.9**	58.4	41.5	**30.5**	**29.5**	42.5	32.2	37.7
Lee et al. [17] ECCV'18	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
Hossain & Little [13] ECCV'18	36.9	37.9	42.8	40.3	46.8	46.7	37.7	36.5	48.9	52.6	45.6	39.6	43.5	35.2	38.5	42.0
Pavllo et al. [29] CVPR'19	34.1	36.1	**34.4**	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	**25.6**	**27.3**	36.5
Our result	28.7	**30.3**	35.1	**31.6**	**30.2**	**36.8**	31.5	**29.3**	41.3	**45.9**	**33.1**	34.0	31.4	26.1	27.8	**32.8**

Table 4. Quantitative evaluation using P-MPJPE in millimeter between estimated pose and the ground-truth on Human3.6M under *Protocol #2*. Procrustes alignment to the ground-truth is used in post-processing. Best in bold, second best underlined.

we fix the sequence length to 128 and the threshold to 0.3.

5.4. Ablation Studies

To evaluate the effectiveness of each component in our framework, we perform several ablative experiments on Human3.6M dataset, and the results are shown in Table 2. The "3D TCN" baseline method is to feed complete 2D estimated keypoints, no matter occluded or not, directly to the 3D TCN for final 3D pose estimation. Then, we gradually enable more modules, including 2D TCN, occlusion awareness, pose regularization, and data augmentation. Note that the occlusion awareness is not a separate module like others, but is integrated into 2D keypoints estimation, 2D TCN, and 3D TCN modules.

From Table 2, we see that all our modules contribute obviously to the final performance. The biggest improvement comes from our occlusion awareness module. This validates our assumption that using incomplete 2D keypoints instead of completed but incorrect ones benefits the estimation accuracy. Adding pose regularization reduces the error about about $1.5mm$ and $1.3mm$ under *Protocol #1* and *#2* respectively, showing that the occlusion constraints for missing keypoints are helpful. Our virtual view angle data augmentation scheme increases the diversity of our training pool, further reducing the error by about $1.9mm$ and $1.3mm$ under *Protocol #1* and *#2* respectively.

5.5. Quantitative Results

We evaluate our whole system on two public data sets and compared with state-of-the-art methods. The results on Human3.6M under *Protocol #1* and *#2* are shown in Table 3 and Table 4 respectively. Under *Protocol #1* and *#2*, our method outperforms the previous best results [29] by about both 4mm on average, which are about 8.3% and 10.1% error reduction rates, while Pavllo et al. [29] reduced the errors by about 9.8% and 3.2%, compared with state-of-the-art then. Note that in Table 4, although Yang et al [43] has lower errors in five actions, their results are unstable, leading to much higher overall error than ours. It is noteworthy that an improvement of $3 \sim 4$ mm is an averaged performance; among the top 10K joints with largest errors in H3.6M dataset, our occlusion-aware module significantly reduced the average error from 713mm to 382mm. These experiments demonstrate that by using our occlusion-aware framework, we could better deal with occluded human joints, and recover them from other confident ones.

We also evaluate our approach on HumanEva-I [31] dataset, and the results are shown in Table 5. Our method outperforms the state-of-the-art [29] by 9.5%, which is a solid improvement considering the performance on this relatively small dataset is almost saturated.

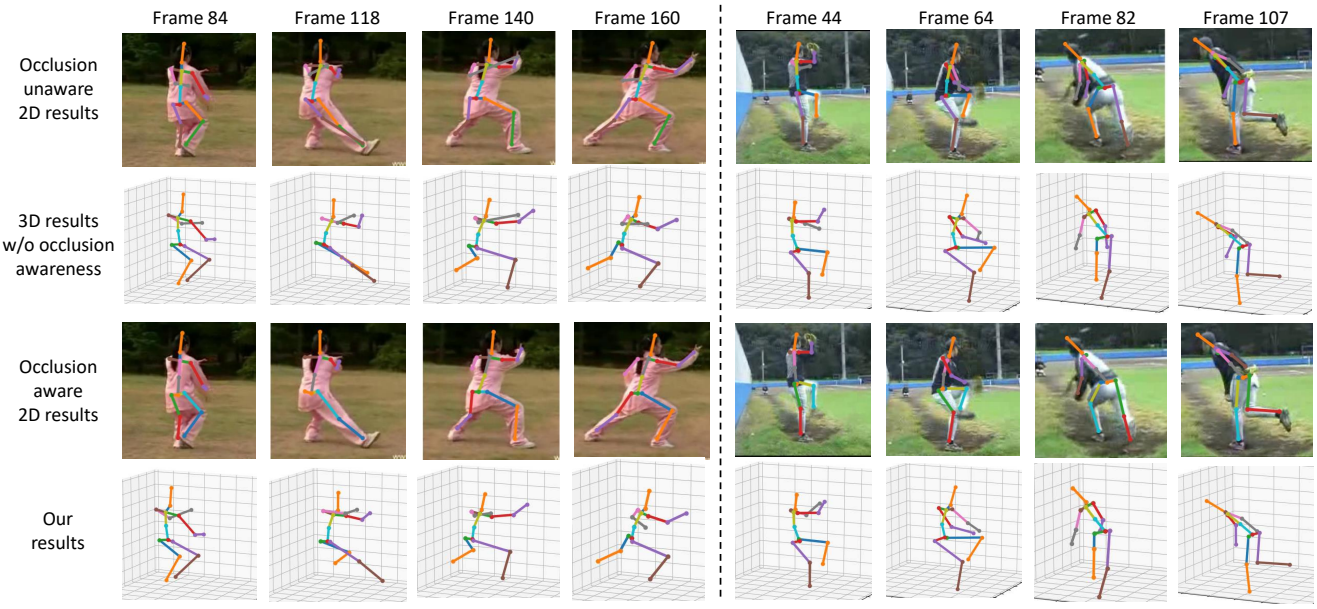

Figure 8. Examples of results from our whole framework as well as disabling the occlusion-aware module.

Method	Walking			Jogging			Avg
Pavlakos et al. [28] CVPR'17	22.3	19.5	29.7	28.9	21.9	23.8	24.3
Martinez et al. [21] ICCV'17	19.7	17.4	46.8	26.9	18.2	18.6	24.6
Pavlakos et al. [27] CVPR'18 *	18.8	12.7	29.2	23.5	15.4	14.5	18.3
Lee et al. [17] ECCV'18	18.6	19.9	30.5	25.7	16.8	17.7	21.5
Hossain et al. [13] ECCV'18	19.1	13.6	43.9	23.2	16.9	15.5	22.0
Pavllo et al. [29] CVPR'19	13.4	10.2	27.2	**17.1**	13.1	13.8	15.8
Our result	**11.7**	**10.1**	**22.8**	18.7	**11.4**	**11.0**	**14.3**

Table 5. Evaluation on HumanEva-I dataset under *Protocol #2*. **Legend:** (*) uses extra depth annotations for ordinal supervision. Best in bold, second best underlined.

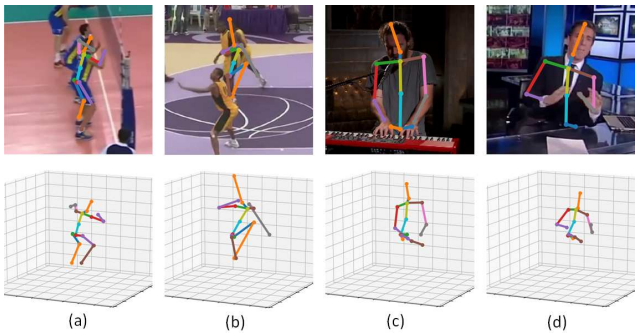

Figure 9. Failure cases caused by (a) multi-person overlapping, (b) detection or tracking error, and (c)(d) long-term heavy occlusion.

5.6. Qualitative Results

We show some example results in Figure 8. When occlusion occurs, the 2D estimations of keypoints are often incorrect, such as the left arm and leg in Frame 118 & 140, and the right arm in Frame 82 & 107 in the first row. Without occlusion awareness modules, such erroneous keypoint detections are treated the same as other reliable ones, leading to possible wrong 3D pose estimation as shown in the second row. However, our approach removes those unreliable 2D keypoints as shown in the third row, and only uses the reliable ones to produces more accurate and stable 3D estimation results as shown in the fourth row.

5.7. Limitations and Future Work

Although our framework outperforms state-of-the-art methods on public datasets, there are still several unsolved problems, and some failure examples are shown in Figure 9. Like other top down based human pose estimation methods, we assume the bounding boxes after detection and tracking are mostly correct. If the bounding box deviates too much from the ground-truth, our pose estimation may fail. Also, if two or more humans are very close, our approach may fail to distinguish the keypoints from different persons. Moreover, our "Cylinder Man Model" is defined to estimate the self-occlusions, but could not directly deal with occlusions by other objects. Finally, our method could not deal with long time heavy occlusion. In this case, little or no temporal information is available to recover the heavily occluded keypoints. Solving these issues will be our future work.

6. Conclusion

We propose an occlusion-aware framework for estimation of human 3D poses from an input video. The suppression of unreliable 2D keypoints estimation reduces the risk of accumulative error. Our "Cylinder Man Model" is effective at improving the diversity of our training data and regularizing the occluded keypoints. Our approach improves the estimation accuracy on Human3.6M dataset by about 10% and on HumanEva-I dataset by about 9.5%.

References

[1] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *International Conference on Automatic Face and Gesture Recognition (FG)*. IEEE, 2017.

[2] Federica Bogo, Michael J Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2300–2308, 2015.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017.

[5] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] 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 (CVPR)*, pages 7035–7043, 2017.

[7] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1212–1221, 2017.

[8] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1831–1840, 2017.

[9] Ahmed Elhayek, Edilson de Aguiar, Arjun Jain, Jonathan Tompson, Leonid Pishchulin, Micha Andriluka, Chris Bregler, Bernt Schiele, and Christian Theobalt. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3810–3818, 2015.

[10] 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.

[11] Golnaz Ghiasi, Yi Yang, Deva Ramanan, and Charless C. Fowlkes. Parsing occluded people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2961–2969, 2017.

[13] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 69–86. Springer, 2018.

[14] 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 (TPAMI)*, 36(7):1325–1339, jul 2014.

[15] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8320–8329, 2018.

[16] Isinsu Katircioglu, Bugra Tekin, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Learning latent representations of 3d human pose with deep neural networks. *International Journal of Computer Vision (IJCV)*, 126(12):1326–1341, 2018.

[17] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018.

[18] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248:1–248:16, Oct. 2015.

[20] 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 (CVPR)*, pages 5137–5146, 2018.

[21] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2640–2649, 2017.

[22] 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 (3DV), 2017 Fifth International Conference on*, 2017.

[23] 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 (CVPR)*, pages 2823–2832, 2017.

[24] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2277–2287, 2017.

[25] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016.

[26] Bruce Xiaohan Nie, Ping Wei, and Song-Chun Zhu. Monocular 3d human pose estimation by predicting depth on joints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3467–3475. IEEE, 2017.

[27] 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 (CVPR)*, pages 7307–7316, 2018.

[28] 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 (CVPR)*, pages 7025–7034, 2017.

[29] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2226–2234, 2018.

[31] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision (IJCV)*, 87(1-2):4, 2010.

[32] Leonid Sigal and Michael J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2041–2048, New York, NY, June 2006.

[33] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2602–2611, 2017.

[36] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3941–3950, 2017.

[37] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems (NIPS)*, pages 1799–1807, 2014.

[38] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1653–1660, 2014.

[39] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3332–3341, 2017.

[40] 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 (CVPR)*, pages 4724–4732, 2016.

[41] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[42] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1281–1290, 2017.

[43] 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 (CVPR)*, pages 5255–5264, 2018.

[44] Qi Ye and Tae-Kyun Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 817–834, 2018.

[45] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7287–7296, 2018.

[46] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 398–407, 2017.