# Exploiting temporal information for 3D pose estimation

Mir Rayat Imtiaz Hossain
University of British Columbia
Vancouver, BC ,Canada
rayat137@cs.ubc.ca

James J. Little
University of British Columbia
Vancouver, BC ,Canada
little@cs.ubc.ca

## Abstract

*In this work, we address the problem of 3D human pose estimation from a sequence of 2D human poses. Although the recent success of deep networks has led many state-of-the-art methods for 3D pose estimation to train deep networks end-to-end to predict from images directly, the top-performing approaches have shown the effectiveness of dividing the task of 3D pose estimation into two steps: using a state-of-the-art 2D pose estimator to estimate the 2D pose from images and then mapping them into 3D space. They also showed that a low-dimensional representation like 2D locations of a set of joints can be discriminative enough to estimate 3D pose with high accuracy. However, estimation of 3D pose for individual frames leads to temporally incoherent estimates due to independent error in each frame causing jitter. Therefore, in this work we utilize the temporal information across a sequence of 2D joint locations to estimate a sequence of 3D poses. We designed a simple sequence-to-sequence network composed of layer-normalized LSTM units with shortcut connections connecting the input to the output on the decoder side and imposed temporal smoothness constraint during training. We found that the knowledge of temporal consistency improves the best reported result by approximately* $17.5\%$ *and helps our network to recover temporally consistent 3D poses over a sequence of images even when the 2D pose detector fails.*

## 1. Introduction

The task of estimating 3D human pose from 2D representations like monocular images or videos is an open research problem among the Computer Vision and Graphics community for a long time. An understanding of human posture and limb articulation is important for high level computer vision tasks such as human action or activity recognition, sports analysis, augmented and virtual reality. A 2D representation of human pose, which is considered to be much easier to estimate, can be used for these tasks. However, 2D poses can be ambiguous because of occlusion and foreshortening. Additionally the poses which are totally different can appear to be similar in 2D because of the way they are projected as shown in Figure 1. The depth information in 3D representation of human pose makes it free from such ambiguities and hence can improve performance for the higher level tasks. Moreover, the 3D pose can be very useful in computer animation, where the articulated pose of a person in 3D can be used to accurately model human posture and movement. However, 3D pose estimation is an ill-posed problem because of the inherent ambiguity in back-projecting a 2D view of an object to the 3D space maintaining its structure. Since the 3D pose of a person can be projected in an infinite number of ways on a 2D plane, the mapping from a 2D pose to 3D is not unique. Moreover, obtaining a dataset for 3D pose is difficult and expensive. Unlike the 2D pose datasets where the users can manually label the keypoints by mouse clicks, 3D pose datasets require a complicated laboratory setup with motion capture sensors and cameras. Hence, there is a lack of motion capture dataset for images in-the-wild.

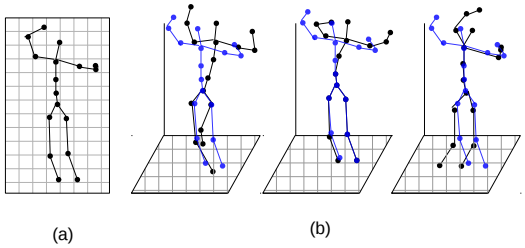


Figure 1. **(a)** 2D position of joints, **(b)** Different 3D pose interpretations of the same 2D pose. Blue points represent the ground truth 3D locations of joints while the black points indicate other possible 3D interpretations. All these 3D poses project to exactly same 2D pose depending on the position and orientation of the camera projecting them onto 2D plane.

Over the years, different techniques have been used to address the problem of 3D pose estimation. Earlier methods used to focus on extracting features, invariant to factors such

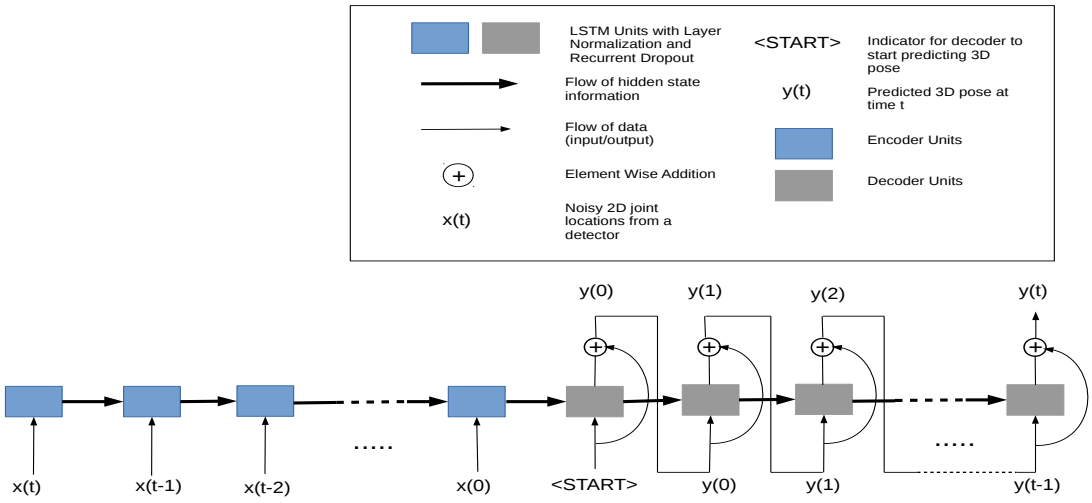


Figure 2. Our model. It is a sequence-to-sequence network [37] with residual connections on the decoder side. The encoder encodes the information of a sequence of 2D poses of length t in its final hidden state. The final hidden state of the encoder is used to initialize the hidden state of decoder. $\langle START \rangle$ symbol tells the decoder to start predicting 3D pose from the last hidden state of the encoder. Note that the input sequence is reversed as suggested by Sutskever *et al*. [37]. The decoder essentially learns to predict the 3D pose at time $(t)$ given the 3D pose at time $(t-1)$. The residual connections help the decoder to learn the perturbation from the previous time step.

as background scenes, lighting, and skin color from images and mapping them into 3D human pose [1, 27, 7, 35]. With the success of deep networks, recent methods tend to focus on training a deep convolutional neural networks (CNN) end-to-end to estimate 3D poses from images directly [39, 32, 20, 24, 47, 25, 29, 22, 31, 36, 40]. Some approaches divided the 3D pose estimation task into first predicting the joint locations in 2D using 2D pose estimators [45, 28] and then back-projecting them to estimate the 3D joint locations [33, 48, 2, 8, 26, 23]. Martinez *et al*. [23] achieved state-of-the-art results on the 3D pose estimation task by building a simple fully connected network with shortcut connections which predicts 3D pose from noisy estimates of 2D joint locations. Their results suggest the effectiveness of decoupling the task of 3D pose estimation where 2D pose estimator abstracts the complexities in the image. Following Martinez *et al*. [23], we adopt the decoupled approach to 3D pose estimation. However, predicting 3D pose for each frame individually can lead to jitter in videos because the errors in each frame are independent of others. Therefore, we designed a sequence-to-sequence network [37] with shortcut connections on the decoder side [14] that predicts a sequence of temporally consistent 3D poses given a sequence of 2D poses. Each unit of our network is a Long Short-Term Memory (LSTM) [15] unit with layer normalization [5] and recurrent dropout [46]. We also imposed a temporal smoothness constraint on the predicted 3D poses during training to ensure that our predictions are smooth over a sequence.

Our network achieves the state-of-the-art result on the Human3.6M dataset improving the previous best result by approximately $7.5\%$. We also obtained the lowest error for every action in Human3.6M dataset [16]. Moreover, we observed that our network was able to predict meaningful 3D poses on Youtube videos, even when the detections from the 2D pose detector were extremely noisy or meaningless which shows the effectiveness of using temporal information. In short our contributions in this work are:

- Designing an efficient sequence-to-sequence network that achieves the state-of-the-art results for every action class of Human3.6M dataset [16] and can be trained very fast.
- Exploiting the ability of sequence-to-sequence networks to memorize the events in the past, to predict a temporally consistent sequence of 3D poses.
- Effectively imposing temporal consistency constraint on the predicted 3D poses during training so that the errors in the predictions are distributed smoothly over the sequence.

We will release our code publicly upon acceptance.

## 2. Related Work

**Representation of 3D pose** Both model-based and model-free representations of 3D human pose have been

used in the past. The most common model-based representation is a skeleton defined by a kinematic tree of a set of joints, parameterized by the offset and rotational parameters of each joint relative to its parent. Several 3D pose methods have used this representation [6, 30, 8, 47]. Others model 3D pose as a sparse linear combination of an over-complete dictionary of basis poses [2, 48, 33]. However, we have chosen a model-free representation of 3D pose, where a 3D pose is simply a set of 3D joint locations relative to the root node like several recent approaches [23, 26, 20, 24]. This representation is much simpler and low-dimensional.

**Estimating 3D pose from 2D joints** Lee and Chen [19] were the first to infer 3D joint locations from their 2D projections given the bone lengths using a binary decision tree where each branch corresponds to two possible states of a joint relative to its parent. Jiang [17] used the 2D joint locations to estimate a set of hypothesis 3D poses using Taylor's algorithm [38] and used them to query a large database of motion capture data to find the nearest neighbor. Gupta *et al*. [13] and Chen and Ramanan [9] also used this idea of using the detected 2D pose to query a large database of exemplar poses to find the nearest nearest neighbor 3D pose. Another common approach to estimating 3D joint locations given the 2D pose is to separate the camera pose variability from the intrinsic deformation of the human body, the latter of which is modeled by learning an over-complete dictionary of basis 3D poses from a large database of motion capture data [33, 48, 8, 2, 44]. A valid 3D pose is defined by a sparse linear combination of the bases and by transforming the points using transformation matrix representing camera extrinsic parameters. Moreno-Nouguer [26] used the pairwise distance matrix of 2D joints to learn a distance matrix for 3D joints, which they found invariant up to a rigid similarity transform with the ground truth 3D and used multi-dimensional scaling (MDS) with pose-priors to rule out the ambiguities. Martinez *et al*. [23] designed a simple fully connected network with shortcut connections every two linear layers to estimate 3D joint locations relative to the root node in the camera coordinate space and obtained the state-of-the-art result. Their work motivated us to follow the idea of decoupling the task of 3D pose estimation and design a network to map the 2D pose detections into 3D pose.

**Deep network based methods** With the success of deep networks, many have designed networks that can be trained end-to-end to predict 3D poses from images directly [32, 20, 39, 31, 24, 36, 47, 43, 34, 42]. Li *et al*. [20] and Park *et al*. [31] designed CNNs to jointly predict 2D and 3D poses. Mehta *et al*. [24] and Sun *et al*. [36] used transfer learning to transfer the knowledge learned for 2D human pose estimation to the task of 3D pose estimation. Pavlakos *et al*. [32] extended the stacked-hourglass network [28] originally designed to predict 2D heatmaps of each joint to make it predict 3D volumetric heatmaps. Tome *et al*. [42] also extended a 2D pose estimator called Convolutional Pose Machine (CPM) [45] to make it predict 3D pose. Rogesz and Schmid [34] and Varol *et al*. [43] augmented the training data with synthetic images and trained CNNs to predict 3D poses from real images.

**Using temporal information** Since estimating poses for each frame individually leads to incoherent and jittery predictions over a sequence, many approaches tried to exploit the temporal information [4, 41, 48, 10, 25]. Andriluka *et al*. [4] used tracking-by-detection to associate 2D poses detected in each frame individually and used them to retrieve 3D pose. Tekin *et al*. [41] used a CNN to first align bounding boxes of successive frames so that the person in the image is always at the center of the box and then extracted 3D HOG features densely over the spatio-temporal volume from which they regress the 3D pose of the central frame. Mehta *et al*. [25] implemented a real-time system for 3D pose estimation which applies temporal filtering across 2D and 3D poses from previous frames to predict a temporally consistent 3D pose. In this vein, we got the motivation of exploiting the temporal information over a sequence so that the noise in 3D pose prediction is distributed smoothly across each frame thereby reducing jitter.

## 3. Our Approach

**Network Design** We designed a sequence-to-sequence network with LSTM units and residual connections on the decoder side to predict a temporally coherent sequence of 3D poses given a sequence of 2D joint locations. Figure 2 shows the architecture of our network. The motivation behind using a sequence-to-sequence network comes from its application on the task of Neural Machine Translation (NMT) by Sutskever *et al*. [37], where their model translates a sentence in one language to a sentence in another language e.g. English to French. In a language translation model, the input and output sentences can have different lengths. Our case is analogous to the NMT but is simpler because our input and output sequences always have the same length.

The encoder side of our network takes a sequence of 2D poses and encodes them in a fixed size high dimensional vector in the hidden state of its final LSTM unit. Since the LSTMs are excellent in memorizing events and information from the past, the encoded vector stores the 2D pose information of all the frames. The initial state of the decoder is initialized by the final state of the encoder. A $\langle START \rangle$ token is passed as initial input to the decoder, which in our case is a vector of ones, telling it to start decoding. Given a 3D pose estimate $y_t$ at a time step $t$ each decoder unit predicts the 3D pose for next time step $y_{t+1}$.

| Protocol #1 | Direct. | Discuss | Eating | Greet | Phone | Photo | Pose | Purch. | Sitting | SitingD | Smoke | Wait | WalkD | Walk | WalkT | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LinKDE [16] (SA) | 132.7 | 183.6 | 132.3 | 164.4 | 162.1 | 205.9 | 150.6 | 171.3 | 151.6 | 243.0 | 162.1 | 170.7 | 177.1 | 96.6 | 127.9 | 162.1 |
| Li *et al*. [21] (MA) | – | 136.9 | 96.9 | 124.7 | – | 168.7 | – | – | – | – | – | – | 132.2 | 70.0 | – | – |
| Tekin *et al*. [41] (SA) | 102.4 | 147.2 | 88.8 | 125.3 | 118.0 | 182.7 | 112.4 | 129.2 | 138.9 | 224.9 | 118.4 | 138.8 | 126.3 | 55.1 | 65.8 | 125.0 |
| Zhou *et al*. [48] (MA) | 87.4 | 109.3 | 87.1 | 103.2 | 116.2 | 143.3 | 106.9 | 99.8 | 124.5 | 199.2 | 107.4 | 118.1 | 114.2 | 79.4 | 97.7 | 113.0 |
| Tekin *et al*. [39] (SA) | – | 129.1 | 91.4 | 121.7 | – | 162.2 | – | – | – | – | – | – | 130.5 | 65.8 | – | – |
| Ghezelghieh *et al*. [11] (SA) | 80.3 | 80.4 | 78.1 | 89.7 | – | – | – | – | – | – | – | – | – | 95.1 | 82.2 | – |
| Du *et al*. [10] (SA) | 85.1 | 112.7 | 104.9 | 122.1 | 139.1 | 135.9 | 105.9 | 166.2 | 117.5 | 226.9 | 120.0 | 117.7 | 137.4 | 99.3 | 106.5 | 126.5 |
| Park *et al*. [31] (SA) | 100.3 | 116.2 | 90.0 | 116.5 | 115.3 | 149.5 | 117.6 | 106.9 | 137.2 | 190.8 | 105.8 | 125.1 | 131.9 | 62.6 | 96.2 | 117.3 |
| Zhou *et al*. [47] (MA) | 91.8 | 102.4 | 96.7 | 98.8 | 113.4 | 125.2 | 90.0 | 93.8 | 132.2 | 159.0 | 107.0 | 94.4 | 126.0 | 79.0 | 99.0 | 107.3 |
| Nie *et al*. [29] (MA) | 90.1 | 88.2 | 85.7 | 95.6 | 103.9 | 103.0 | 92.4 | 90.4 | 117.9 | 136.4 | 98.5 | 94.4 | 90.6 | 86.0 | 89.5 | 97.5 |
| Rogez *et al*. [24] (MA) | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 88.1 |
| Mehta *et al*. [24] (MA) | 57.5 | 68.6 | 59.6 | 67.3 | 78.1 | 82.4 | 56.9 | 69.1 | 100.0 | 117.5 | 69.4 | 68.0 | 76.5 | 55.2 | 61.4 | 72.9 |
| Mehta *et al*. [25] (MA) | 62.6 | 78.1 | 63.4 | 72.5 | 88.3 | 93.8 | 63.1 | 74.8 | 106.6 | 138.7 | 78.8 | 73.9 | 82.0 | 55.8 | 59.6 | 80.5 |
| Lin *et al*. [22] (MA) | 58.0 | 68.2 | 63.3 | 65.8 | 75.3 | 93.1 | 61.2 | 65.7 | 98.7 | 127.7 | 70.4 | 68.2 | 72.9 | 50.6 | 57.7 | 73.1 |
| Tome *et al*. [42] (MA) | 65.0 | 73.5 | 76.8 | 86.4 | 86.3 | 110.7 | 68.9 | 74.8 | 110.2 | 173.9 | 84.9 | 85.8 | 86.3 | 71.4 | 73.1 | 88.4 |
| Tekin *et al*. [40] | 54.2 | 61.4 | 60.2 | 61.2 | 79.4 | 78.3 | 63.1 | 81.6 | 70.1 | 107.3 | 69.3 | 70.3 | 74.3 | 51.8 | 63.2 | 69.7 |
| Pavlakos *et al*. [32] (MA) | 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 |
| Martinez *et al*. [23] (MA) | 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 |
| **Our network (MA)** | **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** |
| Martinez *et al*. [23] (GT) (MA) | 37.7 | 44.4 | 40.3 | 42.1 | 48.2 | 54.9 | 44.4 | 42.1 | 54.6 | 58.0 | 45.1 | 46.4 | 47.6 | 36.4 | 40.4 | 45.5 |
| **Our network (GT) (MA)** | *35.2* | *40.8* | *37.2* | *37.4* | *43.2* | *44.0* | *38.9* | *35.6* | *42.3* | *44.6* | *39.7* | *39.7* | *40.2* | *32.8* | *35.5* | *39.2* |

Table 1. Results showing the errors action-wise on Human3.6M [16] under Protocol #1 (no rigid alignment or similarity transform applied in post-processing). **Note that our results reported here are for sequence of length 5**. SA indicates that a model was trained for each action, and MA indicates that a single model was trained for all actions. GT indicates that the network was trained on ground truth 2D pose. The bold-faced numbers represent the best result while underlined numbers represent the second best.

Note that the order of input sequence is reversed as recommended by Sutskever *et al*. [37]. The shortcut connections on the decoder side causes each decoder unit to estimate the amount of perturbation in the 3D pose from the previous frame instead of having to estimate the actual 3D pose for each frame. As suggested by He *et al*. [14], such a mapping is easier to learn for the network.

We use Layer normalization [5] and recurrent dropout [46] to regularize our network. Ba *et al*. [5] came up with the idea of layer normalization which estimates the normalization statistics (mean and standard deviation) from the summed inputs to the recurrent neurons of hidden layer on a *single* training example to regularize the RNN units. Similarly, Zaremba *et al*. [46] proposed the idea of applying dropout only on the non-recurrent connections of the network with a certain probability $p$ while always keeping the recurrent connections intact because they are necessary for the recurrent units to remember the information from the past.

**Loss function** Given a sequence of 2D joint locations as input, our network predicts a sequence of 3D joint locations relative to the root node (central hip). We predict each 3D pose in the camera coordinate space instead of predicting them in an arbitrary global frame as suggested by Martinez *et al*. [23] who found it easier to regress the 3D coordinates of the joints in the camera coordinate space.

We impose a temporal smoothness constraint on the predicted 3D joint locations to ensure that the prediction of each joint in one frame does not differ too much from its previous frame. Hence we added the L2 norm of the first order derivative on the 3D joint locations with respect to time to our loss function during training.

Empirically we found that certain joints are more difficult to estimate accurately e.g. wrist, ankle, elbow compared to others. To address this issue, we partitioned the joints into three disjoint sets $\mathbf{torso\_head}$, $\mathbf{limb\_mid}$ and $\mathbf{limb\_terminal}$ based on their contribution to overall error. We observed that the joints connected to the torso and the head e.g. hips, shoulders, neck are always predicted with high accuracy compared to those joints belonging to the limbs and therefore put them in the set $\mathbf{torso\_head}$. The joints of the limbs are always more difficult to predict due to their high range of motion. To our observation, the terminal joints of the limbs i.e. wrists and ankles are more difficult to predict accurately than the knees and elbows. Therefore, we put the knees and the elbows in the set $\mathbf{limb\_mid}$ and the terminal joints in the set $\mathbf{limb\_terminal}$. We multiply the derivatives of each set of joints with different scalar values, with the highest value being assigned to the derivatives of the set of terminal joints to penalize them more.

Therefore our loss function consists of the sum of two separate terms: Mean Squared Error (MSE) of $N$ different sequences of 3D joint locations; and the mean of L2 norm of the first order derivative of $N$ sequences of 3D joint locations with respect to time, where the joints are divided into three disjoint sets.

The MSE over $N$ sequences, each of $T$ time-steps, of 3D joint locations is given by,

$$\mathbf{L}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left\| \hat{\mathbf{Y}}_{\mathbf{i,t}} - \mathbf{Y}_{\mathbf{i,t}} \right\|_2^2. \qquad (1)$$

| Protocol #2 | Direct. | Discuss | Eating | Greet | Phone | Photo | Pose | Purch. | Sitting | SitingD | Smoke | Wait | WalkD | Walk | WalkT | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Akhter & Black [2]* (MA) 14j | 199.2 | 177.6 | 161.8 | 197.8 | 176.2 | 186.5 | 195.4 | 167.3 | 160.7 | 173.7 | 177.8 | 181.9 | 176.2 | 198.6 | 192.7 | 181.1 |
| Ramakrishna *et al*. [33]* (MA) 14j | 137.4 | 149.3 | 141.6 | 154.3 | 157.7 | 158.9 | 141.8 | 158.1 | 168.6 | 175.6 | 160.4 | 161.7 | 150.0 | 174.8 | 150.2 | 157.3 |
| Zhou *et al*. [48]* (MA) 14j | 99.7 | 95.8 | 87.9 | 116.8 | 108.3 | 107.3 | 93.5 | 95.3 | 109.1 | 137.5 | 106.0 | 102.2 | 106.5 | 110.4 | 115.2 | 106.7 |
| Rogez *et al*. [24] (MA) | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 87.3 |
| Nie *et al*. [29] (MA) | 62.8 | 69.2 | 79.6 | 78.8 | 80.8 | 86.9 | 72.5 | 73.9 | 96.1 | 106.9 | 88.0 | 70.7 | 76.5 | 71.9 | 76.5 | 79.5 |
| Mehta *et al*. [24] (MA) 14j | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 54.6 |
| Bogo *et al*. [8] (MA) 14j | 62.0 | 60.2 | 67.8 | 76.5 | 92.1 | 77.0 | 73.0 | 75.3 | 100.3 | 137.3 | 83.4 | 77.3 | 86.8 | 79.7 | 87.7 | 82.3 |
| Moreno-Noguer [26] (MA) 14j | 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 |
| Tekin *et al*. [40] (MA) 17j | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 50.1 |
| Pavlakos *et al*. [32] (MA) 17j | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | 51.9 |
| Martinez *et al*. [23] (MA) 17j | 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 |
| **Our network (MA) 17j** | **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** |

Table 2. Results showing the errors action-wise on Human3.6M [16] dataset under protocol #2 (rigid alignment in post-processing). **Note that the results reported here are for sequence of length 5**. The 14j annotation indicates that the body model considers 14 body joints while 17j means considers 17 body joints. (SA) annotation indicates per-action model while (MA) indicates single model used for all actions. The bold-faced numbers represent the best result while underlined numbers represent the second best. The results of the methods are obtained from the original papers, except for (*), which were obtained from [8].

Here, $\hat{\mathbf{Y}}$ denotes the estimated 3D joint locations while $\mathbf{Y}$ denotes 3D ground truth.

The mean of L2 norm of the first order derivative of $N$ sequences of 3D joint locations, each of length $T$, with respect to time is given by,

$$\left\|\nabla_{\mathbf{t}}\hat{\mathbf{Y}}\right\|_2^2 = \frac{1}{N(T-1)}\sum_{i=1}^{N}\sum_{t=2}^{T}\left\{\eta\left\|\hat{\mathbf{Y}}_{\mathbf{i,t}}^{\mathbf{TH}} - \hat{\mathbf{Y}}_{\mathbf{i,t-1}}^{\mathbf{TH}}\right\|_2^2 + \rho\left\|\hat{\mathbf{Y}}_{\mathbf{i,t}}^{\mathbf{LM}} - \hat{\mathbf{Y}}_{\mathbf{i,t-1}}^{\mathbf{LM}}\right\|_2^2 + \tau\left\|\hat{\mathbf{Y}}_{\mathbf{i,t}}^{\mathbf{LT}} - \hat{\mathbf{Y}}_{\mathbf{i,t-1}}^{\mathbf{LT}}\right\|_2^2\right\}. \quad (2)$$

In the above equation, $\hat{\mathbf{Y}}^{\mathbf{TH}}$, $\hat{\mathbf{Y}}^{\mathbf{LM}}$ and $\hat{\mathbf{Y}}^{\mathbf{LT}}$ denotes the predicted 3D locations of joints belonging to the sets $\mathbf{torso\_head}$, $\mathbf{limb\_mid}$ and $\mathbf{limb\_terminal}$ respectively. The $\eta$, $\rho$ and $\tau$ are scalar hyper-parameters to control the significance of the derivatives of 3D locations of each of the three set of joints. A higher weight is assigned to the set of joints which are generally predicted with higher error.

The overall loss function for our network is given as,

$$\mathbf{L} = \min_{\hat{\mathbf{Y}}} \alpha\mathbf{L}(\hat{\mathbf{Y}}, \mathbf{Y}) + \beta\left\|\nabla_{\mathbf{t}}\hat{\mathbf{Y}}\right\|_2^2. \quad (3)$$

Here $\alpha$ and $\beta$ are scalar hyper-parameters regulating the importance of each of the two terms in the loss function.

## 4. Experimental Evaluation

**Datasets and protocols** We perform quantitative evaluation on Human 3.6M [16] dataset, which to the best of our knowledge is the largest publicly available dataset for human 3D pose estimation. The dataset contains 3.6 million images of 7 different professional actors performing 15 everyday activities like walking, eating, sitting, making a phone call. The dataset consists of 2D and 3D joint locations for each corresponding image. Each video is captured using 4 different calibrated high resolution cameras. They also used 10 different motion capture cameras and 1 time of flight sensor to accurately capture the motion of the actors in 3D. In addition to 2D and 3D pose ground truth, the dataset also provides ground truth for bounding boxes, the camera parameters, the body proportion of all the actors and high resolution body scans or meshes of each actor. For qualitative evaluation, we used the some videos from Youtube and Human3.6M dataset.

We follow the standard protocols of Human3.6M dataset used in the literature. We used subjects 1, 5, 6, 7, and 8 for training, and subjects 9 and 11 for testing and the error is evaluated on the predicted 3D pose without any transformation. We refer this as protocol #1. Another common approach used by many to evaluate their methods is to align the predicted 3D pose with the ground truth using a rigid body transformation. We refer this as protocol #2. We use the average error per joint in millimeters between the estimated and the ground truth 3D pose relative to the root node as the error metric.

**2D detections** We fine-tuned a model of stacked-hourglass network [28], initially trained on MPII dataset [3] (a benchmark dataset for 2D pose estimation), on the images of Human3.6M dataset to obtain 2D pose estimations for each image. We used the bounding box information provided with the dataset to crop a $440 \times 440$ region across a person and resized the cropped image to $256 \times 256$. We fine-tuned the network for 250 iterations and used a batch size of 3 and a learning rate of $2.5e-4$.

**Data pre-processing** We normalized the 3D ground truth poses, the noisy 2D pose estimates from stacked-hourglass network and the 2D ground truth [28] by subtracting the mean and dividing by standard deviation. We do not predict the 3D location of the root joint i.e. central hip joint and hence zero center the 3D joint locations relative to the global position of the root node. To obtain the ground truth

| | Moreno-Nouguer [26] | Martinez *et al*. [23] | *Ours* |
|---|---|---|---|
| GT/GT | 62.17 | 37.10 | **31.67** |
| GT/GT + $\mathcal{N}(0, 5)$ | 67.11 | 46.65 | **37.46** |
| GT/GT + $\mathcal{N}(0, 10)$ | 79.12 | 52.84 | **49.41** |
| GT/GT + $\mathcal{N}(0, 15)$ | 96.08 | **59.97** | 61.80 |
| GT/GT + $\mathcal{N}(0, 20)$ | 115.55 | **70.24** | 73.65 |
| GT/SH [28] | – | **60.52** | 62.43 |

Table 3. Performance of our system trained with ground truth 2D pose of Human3.6M [16] dataset and tested with different levels of additive Gaussian noise **(Top)** and on 2D pose predictions from stacked-hourglass [28] pose detector **(Bottom)** under protocol #2.

| | error (mm) | $\Delta$ |
|---|---|---|
| Ours | 51.9 | – |
| w/o temporal consistency constraint | 52.7 | 0.8 |
| w/o recurrent dropout | 58.3 | 6.4 |
| w/o layer normalized LSTM | 61.1 | 9.2 |
| w/o layer norm and recurrent dropout | 59.5 | 7.6 |
| w/o residual connections | 102.4 | 50.5 |

Table 4. Ablative and hyperparameter sensitivity analysis.

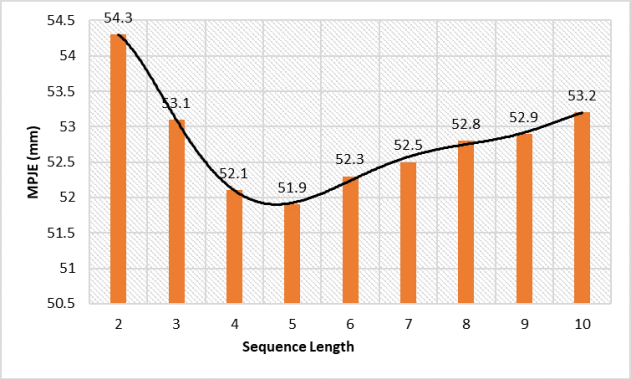


Figure 3. Mean Per Joint Error(MPJE) in mm of our network for different sequence lengths.

3D poses in camera coordinate space, an inverse rigid body transformation is applied on the the ground truth 3D poses in global coordinate space using the given camera parameters.

To generate both training and test sequences, we translated a sliding window of length $T$ by one frame. Hence there is an overlap between the sequences. This gives us more data to train on, which is always an advantage for deep learning systems. During test time, we initially predict the first $T$ frames of the sequence and as the window is slid in an overlapping manner, we predict only the $T$-th frame until we reach end of the sequence.

**Training details** We trained our network for 100 epochs, where each epoch makes a complete pass over the entire Human 3.6M dataset. We used the Adam [18] optimizer for training the network with a learning rate of $1e-5$ which is decayed exponentially per iteration. The weights of the LSTM units are initialized by Xavier uniform initializer [12]. We used a minibatch batch size of 32 i.e. 32 sequences. For most of our experiments we used a sequence length of 5, because it allows faster training with high accuracy. We experimented with different sequence lengths and found sequence length 4, 5 and 6 to generally give better results, which we will discuss in detail in the results section. We trained a single model for all the action classes. Our code is implemented in Tensorflow. We empirically set the hyper-parameter values $\alpha$ and $\beta$ of our loss function to 1 and 5 respectively. Similarly the three hyper-parameters of the temporal consistency constraint $\eta, \rho$ and $\tau$, are set to $1, 2.5$ and $4$ respectively. A single training step for sequences of length 5 takes only 34 ms approximately, while a forward pass takes only about 16ms on NVIDIA Titan X GPU. Therefore on average, our network takes only about 3.2ms to predict 3D pose per frame.

### 4.1. Quantitative results

**Evaluation on estimated 2D pose** As mentioned before, we used a sequence length of 5 to perform both qualitative and quantitative evaluation of our network. The results on Human3.6M dataset [16] under protocol #1 are shown in Table 1. From the table we observe that our model achieves the lowest error for every action class under protocol #1, unlike many of the previous state-of-the-art methods. Note that we train a single model for all the action classes unlike many other methods which trained a model for each action class. Our network significantly improves the state-of-the-art result of Martinez *et al*. [23] by approximately $17.5\%$ (by 11 mm).

The results under protocol #2, which aligns the predictions to the ground truth using a rigid body similarity transform before computing the error, is reported in Table 2. Our network improves the reported state-of-the-art results by $11.9\%$ (by $5.7$ mm) and achieves the lowest error for each action in protocol #2 as well.

From the results, we observe that exploiting temporal information across multiple sequences is indeed useful. It significantly improves the overall accuracy of the estimates of 3D joint locations, especially on actions like *phone* and *sitting down* on which most of the previous methods did not perform well due to heavy occlusion.

**Evaluation on 2D ground truth** As suggested by Martinez *et al*. [23], we also found that the more accurate the 2D joint locations are, the better are the estimates for 3D pose. We trained our model on ground truth 2D poses for a sequence length of 5. The results under protocol #1 are reported in Table 1. As seen from the table, our model improves the lower bound error of Martinez *et al*. [23] by al-

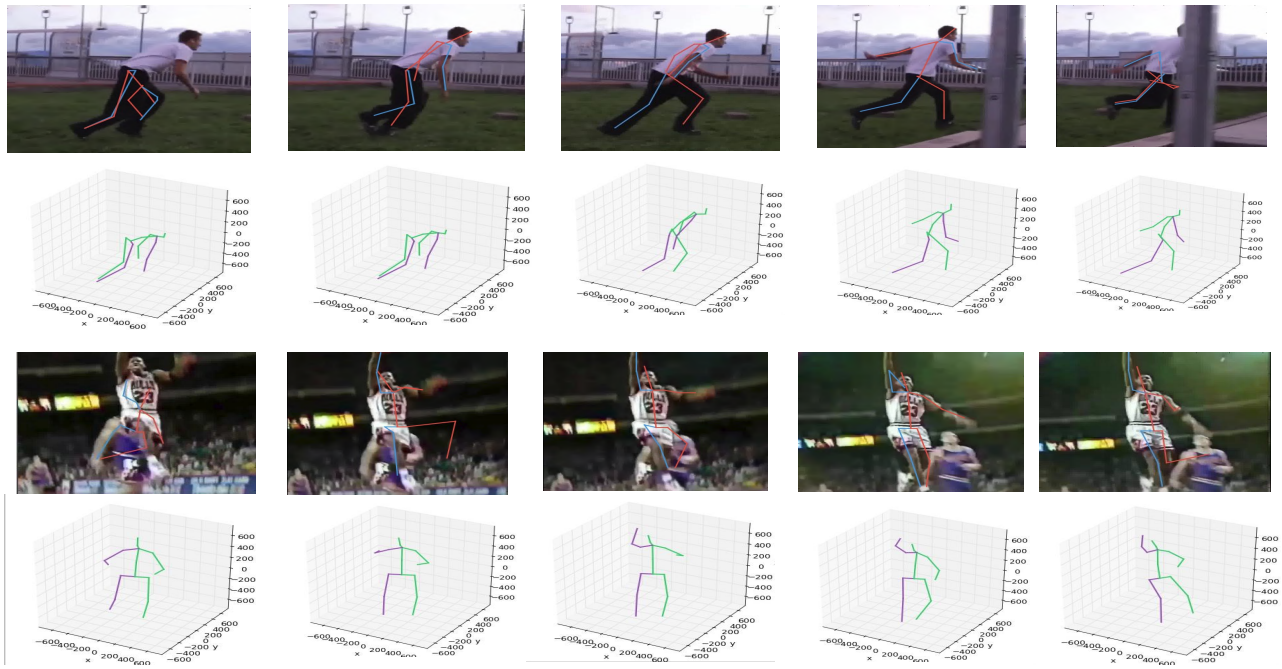

Figure 4. Qualitative results on Youtube videos

most 13.8%.

The results on ground truth 2D joint input for protocol #2 are reported in Table 3. When there is no noise in 2D joint locations, our network performs better than the models by Martinez *et al*. [23] and Moreno-Nouguer [26]. These results suggest that the information of temporal consistency from previous frames is a valuable cue for the task of estimating 3D pose even when the detections are noise free.

**Robustness to noise** We carried out some experiments to test the tolerance of our model to different levels of noise in the input data by training our network on 2D ground truth poses and testing on inputs corrupted by different levels of Gaussian noise and from the detections of stacked-hourglass 2D pose detector [28] pre-trained on MPII so that we can compare with Martinez *et al*. [23]. Table 3 shows how our final model compares against the models by Moreno-Nouguer [26] and Martinez *et al*. [23]. Our network is significantly more robust than Moreno-Nouguer's model [26]. When compared against Martinez *et al*. [23] our network performs better when level of input noise is low i.e. standard deviation less than or equal to 10. However, for higher levels of noise and for stacked-hourglass detections (which has an average error of 15 pixels) our network performs slightly worse than Martinez *et al*. [23]. We would like to attribute the cause of this to the temporal smoothness constraint imposed during training which distributes the error of individual frames over the entire sequence. However, its usefulness can be observed in the qualitative results (See Figure 4 and Figure 5).

**Ablative analysis** To show the usefulness of each component and design decision of our network, we perform an ablative analysis. We follow protocol #1 for performing ablative analysis and trained a single model for all the actions. The results are reported in Table 4. We observe that the biggest improvement in result is due the the residual connections on the decoder side, which agrees with the hypothesis of He *et al*. [14]. Removing the residual connections massively increases the error by 50.5 mm. When we do not apply layer normalization on LSTM units, the error increases by 9.2 mm. On the other hand when dropout is not performed, the error raises by 6.4 mm. When both layer normalization and recurrent dropout are not used the results get worse by 7.6 mm. Although the temporal consistency constraint may seem to have less impact (only 0.8 mm) quantitatively on the performance of our network, it ensures that the predictions over a sequence is smooth and temporally consistent which is apparent from our qualitative results discussed as seen in Figure 4 and Figure 5.

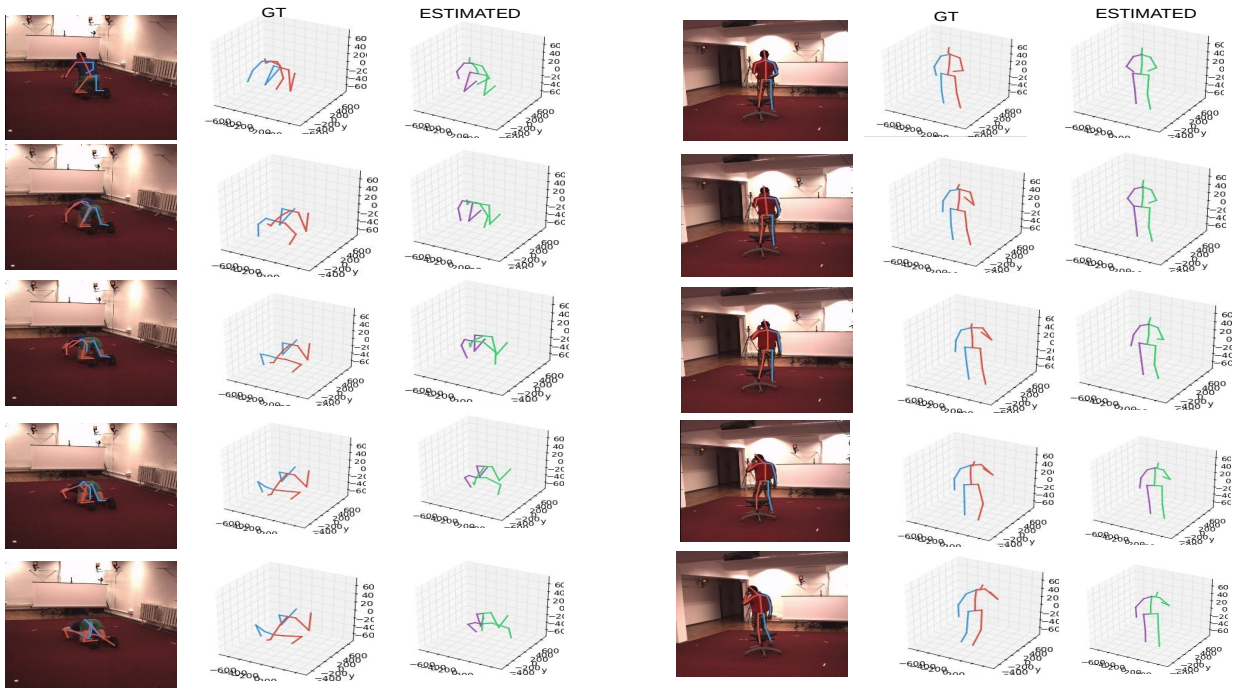


Figure 5. Qualitative results on Human3.6M videos. The images on the **left** are for subject 11 and action *sitting down*. On the **right** the images are for subject 9 and action *phoning*. 3D poses in the center is the ground truth and on the right is the estimated 3D pose.

**Performance on different sequence lengths** The results reported so far have been for input and output sequences of length 5. We carried out experiments to see how our network performs for different sequence lengths ranging from 2 to 10. The results are shown in Figure 3. As it can be seen, the performance of our network remains stable for sequences of varying lengths. Particularly the best results were obtained for length 4, 5 and 6. However, we chose sequence length 5 for carrying out our experiments as a compromise between training time and accuracy.

### 4.2. Qualitative Analysis

We provide qualitative results on some vidoes of Human3.6M and Youtube. The bounding box for each person in the Youtube video is labeled manually and for Human3.6M the ground truth bounding box is used. The 2D poses are detected using the stacked-hourglass model fine-tuned on Human3.6M data. The qualitative result for Youtube vidoes is shown in Figure 4 and for Human3.6M in Figure 5. The real advantage of using temporal smoothness constraint during training is apparent in these figures. For Figure 4, we can see that even when the 2D pose estimator breaks or generates extremely noisy detections, our system can recover temporally coherent 3D poses by exploiting the temporal consistency information. A similar trend can also be found for Human3.6M videos in Figure 5, particularly for the action *sitting down* of subject 11. We have provided more qualitative results in the supplementary material.

## 5. Conclusion

Both the quantitative and qualitative results for our network show the effectiveness of exploiting temporal information over multiple sequences to estimate 3D poses which are temporally smooth. Our network achieved the best accuracy on all of the 15 action classes in Human3.6M dataset [16]. Particularly, most of the previous methods struggled with actions which have a high degree of occlusion like *taking photo*, *talking on the phone*, *sitting* and *sitting down*. Our network has significantly better results on these actions. Additionally we found that our network is reasonably robust to noisy 2D poses. Although the contribution of temporal smoothness constraint is not apparent in the ablative analysis in Table 4, its effectiveness is clearly visible in the qualitative results particularly on challenging Youtube vidoes (see Figure 4).

Our network effectively demonstrates the power of using temporal information which we achieved using a simple sequence-to-sequence network which can be trained efficiently in a reasonably quick time. Also our network makes predictions at 3ms per frame on average which suggests that given the 2D pose detector is real time, our network can be applied in real time scenarios.

## References


[1] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *CVPR*, 2004.

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

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[4] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 623–630. IEEE, 2010.

[5] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[6] C. Barron and I. A. Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *CVIU*, 81(3):269–284, 2001.

[7] L. F. Bo, C. Sminchisescu, A. Kanaujia, and D. N. Metaxas. Fast algorithms for large scale conditional 3D prediction. In *CVPR*, pages 1–8, 2008.

[8] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[9] C.-H. Chen and D. Ramanan. 3d human pose estimation= 2d pose estimation+ matching. *arXiv preprint arXiv:1612.06524*, 2016.

[10] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-less 3d human motion capture with monocular image sequence and height-maps. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[11] M. F. Ghezelghieh, R. Kasturi, and S. Sarkar. Learning camera viewpoint using cnn to improve 3d body pose estimation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 685–693. IEEE, 2016.

[12] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[13] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham. 3D Pose from Motion for Cross-view Action Recognition via Non-linear Circulant Temporal Encoding. In *CVPR*, 2014.

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

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[17] H. Jiang. 3d human pose reconstruction using millions of exemplars. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1674–1677. IEEE, 2010.

[18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] H. J. Lee and Z. Chen. Determination of 3D human body postures from a single view. *Computer Vision, Graphics and Image Processing*, 30:148–168, 1985.

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

[21] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *ICCV*, 2015.

[22] M. Lin, L. Lin, X. Liang, K. Wang, and H. Chen. Recurrent 3d pose sequence machines. In *CVPR*, 2017.

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

[24] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation using transfer learning and improved cnn supervision. *arXiv preprint arXiv:1611.09813*, 2016.

[25] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *arXiv preprint arXiv:1705.01583*, 2017.

[26] F. Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *CVPR*, 2017.

[27] G. Mori and J. Malik. Recovering 3D human body configurations using shape contexts. *TPAMI*, 28(7):1052–1062, July 2006.

[28] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[29] B. X. Nie, P. Wei, and S.-C. Zhu. Monocular 3d human pose estimation by predicting depth on joints. 2017.

[30] V. Parameswaran and R. Chellappa. View independent human body pose estimation from a single perspective image. In *CVPR*, 2004.

[31] S. Park, J. Hwang, and N. Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *Computer Vision–ECCV 2016 Workshops*, pages 156–169. Springer, 2016.

[32] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR*, 2017.

[33] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3d human pose from 2d image landmarks. *Computer Vision–ECCV 2012*, pages 573–586, 2012.

[34] G. Rogez and C. Schmid. Mocap-guided data augmentation for 3D pose estimation in the wild. In *NIPS*, 2016.

[35] G. Shakhnarovich, P. A. Viola, and T. J. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, 2003.

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

[37] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

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

[39] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. In *BMVC*, 2016.

[40] B. Tekin, P. Marquez Neila, M. Salzmann, and P. Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *International Conference on Computer Vision (ICCV)*, number EPFL-CONF-230311, 2017.

[41] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 991–1000, 2016.

[42] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *arXiv preprint arXiv:1701.00295*, 2017.

[43] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[44] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao. Robust estimation of 3d human poses from a single image. In *CVPR*, 2014.

[45] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[46] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[47] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. In *Computer Vision–ECCV 2016 Workshops*, pages 186–201. Springer, 2016.

[48] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4966–4975, 2016.