Monocular 3D Human Pose Estimation by Predicting Depth on Joints

Bruce Xiaohan Nie[1*], Ping Wei[2,1*], and Song-Chun Zhu[1]
[1]Center for Vision, Cognition, Learning, and Autonomy, UCLA, USA
[2]Xi'an Jiaotong University, China
xiaohan.nie@gmail.com, pingwei@xjtu.edu.cn, sczhu@stat.ucla.edu

Abstract

This paper aims at estimating full-body 3D human poses from monocular images of which the biggest challenge is the inherent ambiguity introduced by lifting the 2D pose into 3D space. We propose a novel framework focusing on reducing this ambiguity by predicting the depth of human joints based on 2D human joint locations and body part images. Our approach is built on a two-level hierarchy of Long Short-Term Memory (LSTM) Networks which can be trained end-to-end. The first level consists of two components: 1) a skeleton-LSTM which learns the depth information from global human skeleton features; 2) a patch-LSTM which utilizes the local image evidence around joint locations. The both networks have tree structure defined on the kinematic relation of human skeleton, thus the information at different joints is broadcast through the whole skeleton in a top-down fashion. The two networks are first pre-trained separately on different data sources and then aggregated in the second layer for final depth prediction. The empirical evaluation on Human3.6M and HHOI dataset demonstrates the advantage of combining global 2D skeleton and local image patches for depth prediction, and our superior quantitative and qualitative performance relative to state-of-the-art methods.

1. Introduction

1.1. Motivation and Objective

This paper aims at reconstructing full-body 3D human poses from a single RGB image. Specifically we want to localize the human joints in 3D space, as shown in Fig. 1. Estimating 3D human pose is a classic task in computer vision and serves as a key component in many human related practical applications such as intelligent surveillance, human-robot interaction, human activity analysis, human attention recognition,etc. There are some existing works which estimate 3D poses in constrained environment from depth images [40, 26] or RGB images captured simultaneously at multiple viewpoints [2, 10]. Different from them, we focus on recognizing 3D pose directly from a single RGB image which is easier to be captured from general environment.

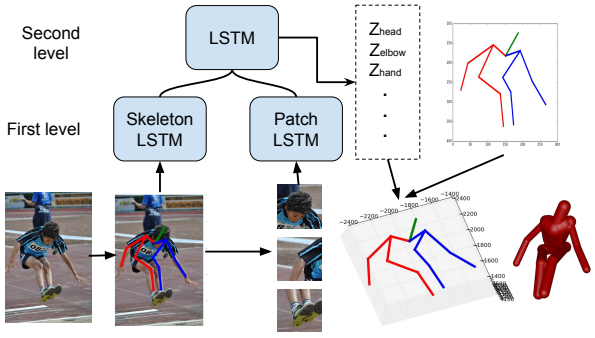

Figure 1. Our two-level hierarchy of LSTM for 3D pose estimation. The skeleton-LSTM and patch-LSTM captures information from global 2D skeleton and local image patches respectively at the first level. The global and local features are integrated in the second level to predict the depth on joints. The 3D pose is recovered by attaching depth values onto the 2D pose. We render the 3D pose for better visualization.

Estimating 3D human poses from a single RGB image is a challenging problem due to two main reasons: 1) the target person in the image always exhibits large appearance and geometric variation because of different clothes, postures, illuminations, camera viewpoints and so on. The highly articulated human pose also brings about heavy self-occlusions. 2) even the ground-truth 2D pose is given, recovering the 3D pose is inherently ambiguous since that there are infinite 3D poses which can be projected onto the same 2D pose when the depth information is unknown.

One inspiration of our work is the huge progress of 2D human pose estimation made by recent works based on deep architectures [33, 32, 17, 37, 3]. In those works, the appearance and geometric variation are implicitly modeled in feed-forward computations in networks with hierarchical deep structure. The self-occlusion is also addressed well by filters from different layers capturing features at different s-

*Bruce Xiaohan Nie and Ping Wei are co-first authors.

cales. Another inspiration of our work is the effectiveness that deep CNN has demonstrated in depth map prediction and segmentation from monocular image [8, 35, 16, 36] instead of stereo images. Most of those approaches directly predict the pixel-wise depth map using deep networks and some of them build markov random fields on the output of deep networks. The largest benefit is that they are not bothered by designing geometric priors or hand-crafted features, and most models can be trained end-to-end using back-propagation. Based on the two above inspirations, in this paper, we propose a novel framework to address the challenge of lifting 2D pose to 3D pose by predicting the depth of joints from two cues: global 2D joint locations and local body part images. The 2D joint locations are predicted from off-the-shelf pose estimation methods.

1.2. Method Overview

Our approach is built on a two-level hierarchy of LSTM networks to predict the depth on human joints and then recover 3D full-body human pose. The first level of our model contains two key components: 1) the skeleton-LSTM network which takes the predicted 2D joint locations to estimate depth of joints. This is based on the assumption that the global human depth information such as global scale and rough depth can be inferred from the correlation between 2D skeleton and 3D pose. This global skeleton feature can help to remove the physically implausible 3D joint configuration and predict depth with considerable accuracy; 2) the patch-LSTM network which takes the local image patches of body parts as input to predict depth. This network addresses the correlation between human part appearance and depth information. To better model the kinematic relation of human skeletons, the two recurrent networks have tree-structures similar to the models in [34, 19, 24]. During training, the features at different joints are broadcasted through the whole skeleton and in testing the depth are predicted for each joint in top-down fashion. The skeleton-LSTM is first pre-trained on 2D-3D pose pairs without any image so infinite training examples can be generated by projecting 3D poses onto 2D space under arbitrary viewpoint. The patch-LSTM is pre-trained on human body patches extracted around 2D joints. To increase appearance variation and reduce overfitting we employ multi-task learning on the combination of two data sources: the MoCap data with the task of depth regression and in-the-wild pose data with the task of 2D pose regression. The two networks are aggregated in the second layer and finetuned together for final depth prediction. We evaluate our method extensively on Human3.6M dataset [11] using two protocols. To test the generalization ability, we test our method on HHOI dataset using the model trained on Human3.6M dataset. The results demonstrate that we achieve better performance over state of the art quantitatively and qualitatively.

1.3. Related Works and Our Contributions

Monocular 3D human pose estimation. With the success of deep networks on a wide range of computer vision tasks and especially 2D human pose estimation, the 3D pose estimation from monocular image using deep networks [14, 15, 23, 39, 41] have received lots of attentions recently. Some approaches [14, 15] directly predict the 3D pose from images so their training and testing are restricted to the 3D MoCap data in a constrained environment, and some methods [22, 4] reconstruct 3D poses only from the 2D landmarks which are from other off-the-shelf methods. Li et al. [14] applies a deep network to regress 3D pose and detect 2D body parts simultaneously. In this method there is no explicit constraint to guarantee that the predicted 3D pose can be projected to the detected 2D part locations. Li et al. [15] learn the common embedding space for both image and 3D pose using a deep network. The matching score of pose and image is computed as the dot product between their embedding vectors. Some methods [39, 41] use two different data sources for training 2D pose estimator and 3D pose predictor. The 3D poses are recovered by minimizing the 3D-2D projection error. The benefit is that their 2D pose estimators can be trained from another data source instead of the 3D Mocap data which is captured in a constrained environment. Zhou et al. [41] predict 3D poses from a video sequence by using temporal information. The 3D pose estimation is conducted via an EM type algorithm over the entire sequence and the 2D joint uncertainties are marginalized out during inference. Yasin et al. [39] propose a dual-source approach to combine 2D pose estimation with 3D pose retrieval. The first data source only contains images with 2D pose annotations for training 2D pose estimator and the second source consists of 3D MoCap data for 3D pose retrieval. Our approach is similar to those works in that we also use two data sources for 3D pose prediction: The Mocap data and in-the-wild 2D pose data, however, we do not assume that the 3D pose is conditional independent of image given 2D pose. The cues from global 2D pose and local image patches are jointly modeled in our two-level network. The 2D pose images are used for the auxiliary task of 2D pose regression to compensate the lack of appearance variation of Mocap data. Another work worth mention is [13] which use regression forest to infer the depth information and estimate 3D joint location probabilities from image patches. The independent joint probabilities are used with the pictorial structure model to infer the full skeleton. We also infer joint depth from image patches, however, our deep network is built on the pictorial structure model and can be trained end-to-end.

Depth estimation from monocular image. Predicting depth from monocular image is a long-standing problem and recent approaches [8, 35, 16, 5, 25] using deep neural networks have made a great progress in this area. Eigen

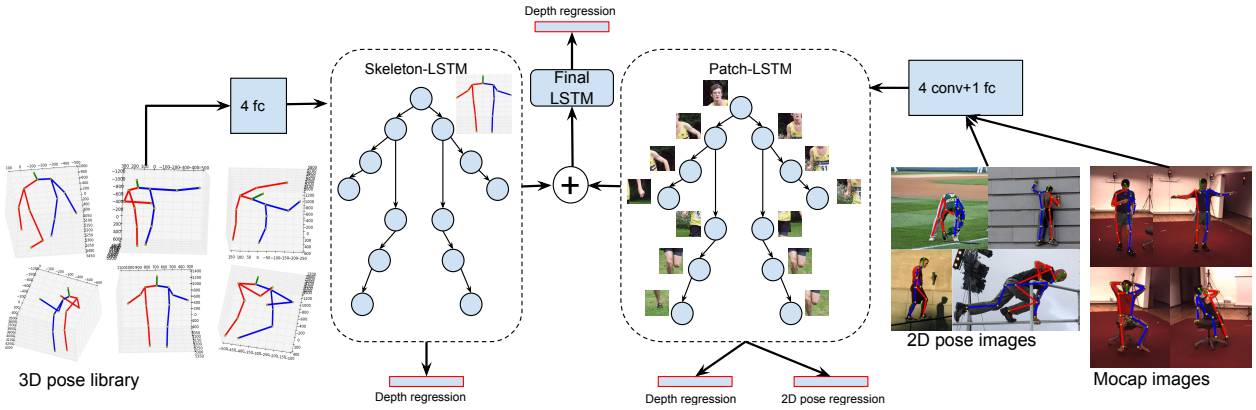

Figure 2. Overview of our model structure and training process. In the first level, the skeleton-LSTM is pre-trained with 2D-3D pose pairs to predict depth from global skeleton features. The patch-LSTM predicts depth from local image patch evidence of body parts. The tree-structure of two networks are defined on the kinematic relation of human joints, so the state of current joint is composed of the hidden states of its parents and the input feature of itself. The two networks are integrated into another LSTM at the second level for end-to-end training. To reduce overfitting of patch-LSTM, we borrow in-the-wild 2D pose images and train the network with multi-task loss: the depth prediction loss and 2D pose regression loss.

et al. [8] apply a multi-scale network to directly regress depth. The coarse-scale network is learned to predict overall depth map structure and the fine-scale network refines the rough depth map using local image evidence. Wang et al. [35] proposed a unified framework for joint learning of depth estimation and sematic segmentation. A deep CNN is trained to do both tasks and a hierarchical CRF is applied in inference to get fine-level details. Liu [16] learn a continuous CRF and a deep CNN jointly. The unary and binary potentials are from the deep CNN. They formulate the depth prediction as a MAP problem and provide closed-form solutions. Chen [5] train a deep network with ranking loss to produce pixel-wise depth using only annotations of relative depth. In this work, we integrate the global and local information from 2D skeleton and local image patches to infer the depth of human joints and our objective function considers both absolute and relative depth loss based on the tree-structured human skeleton.

The contribution of our approach is three-fold:

i) We explore the ability of deep network for predicting the depth of human joints and then recover 3D pose. Our framework is more flexible than others because complex optimization is not needed and model can be trained end-to-end.

ii) We incorporate both global 2D skeleton features and local image patch features in a two-level LSTM network and the tree-structure topology of our model naturally represents the kinematic relation of human skeleton. The features at different joints are aggregated in top-down fashion.

iii) The extensive experiments demonstrate the superior quantitative and qualitative performance of our work relative to other state of the art methods.

2. Models

In this section, we will first describe the relationship between 3D pose estimation and depth prediction, and then introduce our model and its corresponding formulations.

2.1. Recover 3D Pose by Depth Prediction

The 3D human pose is represented by a set of locations of human joints in 3D space. We use $W \in \mathbb{R}^{3\times N}$ to denote the 3D pose in the world's coordinate system where N is the number of joints. Each 3D joint location in W is denoted by the 3D coordinate $w_i = [X_i, Y_i, Z_i]$, $i \in 1, ..., N$. The 2D pose is defined in the same way as $S \in \mathbb{R}^{2\times N}$ and each 2D joint is denoted as $s_i = [x_i, y_i]$. The relationship between each 3D joint and 2D joint can be described as a perspective projection:

$$z_i \cdot \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = f \cdot [R|T] \cdot \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix}, i \in 1, ..., N \qquad (1)$$

where $R \in \mathbb{R}^{3\times 3}$ denotes the camera rotation matrix, f denotes focal length and z_i denotes depth of joint i. Note that in Eq. 1 there is no weak perspective assumption about the relationship between 3D pose and 2D pose. Given 2D joint locations $[x_i, y_i]$ and focal length f we need the depth value for each joint z_i, global rotation R and translation T to recover all 3D joint locations. Since there are infinite combinations of transformation matrix $[R|T]$ and world coordinate $[X_i, Y_i, Z_i]$ which can produce the same $[x_i, y_i]$ and z_i with unknown camera position, therefore in this work we focus on predicting $z = [z_1, ..., z_N]$ to recover the 3D pose in the camera's coordinate system $\hat{W} = [R|T] \cdot [W|\mathbf{1}]^T$.

In order to predict the depth, we define the joint distribution of depth z, 2D pose S and image I:

$$P(z, S, I) = P(z|S, I) \cdot P(S|I) \cdot P(I), \quad (2)$$

where $P(S|I)$ represents the 2D pose estimation from single image I which can be handled by any off-the-shelf 2D pose estimation method. The separate estimation of depth and 2D pose allows $P(S|I)$ to be modeled by any complex method. Any improvement made in $P(S|I)$ can be immediately plugged into $P(z, S, I)$ and re-training of the whole system is not needed. $P(z|S, I)$ is modeled as a two-level hierarchy of LSTM which utilizes the 2D pose S and image evidence I in the first level, and integrates two networks in the second level for final depth prediction. The details of our model are described below.

2.2. Components of our Model

To take advantage of the global skeleton feature and local image feature to predict depth, we use a deep structure of LSTMs with two levels. As shown in Fig. 2, the first level consists of two recurrent networks: a skeleton-LSTM stacked with a 2D pose encoding network which takes the predicted 2D pose S as input and a patch-LSTM stacked with image patch encoding network which takes the local image patches $I(s_i), i \in [1, ..., N]$ around 2D joints as input. The hidden states of the two networks at each joint are max pooled and forwarded to the LSTM at the second level to predict the final real valued depth d_i for each joint.

Inspired by those graphical model based pose estimation methods [38, 24, 20, 18], we represent human pose as a tree structure based on the kinematic relation of skeleton. The articulated relation are better represented and the correlation of features at parent joint and child joint are better captured within tree structure than the flat or sequential structure. Similar to the framework of [29], we adapt the tree-structured LSTM for modeling human pose and integrating global and local features. The aggregated contextual information are propagated efficiently through the edges between joints. In experiments we evaluate different choices of model structure and demonstrate the empirical strength of tree-structure over the flat or sequential model.

The three tree-structured LSTMs in our model share the same equation and only differ in the input data. At joint j, the state of the LSTM unit is composed of the current input feature x_j, all hidden states h_k and memory states c_k from its parents, and the output is the hidden state h_j and memory state c_j which is forwarded to the child joint:

$$(h_j, c_j) = LSTM(x_j, \{h_k\}_{k \in N(j)}, \{c_k\}_{k \in N(j)}, W, U) \quad (3)$$

where W and U are weight matrices. To obtain a fixed dimension of the input feature, the hidden states from all parents of joint j are mean pooled:

$$\bar{h}_j = \Big(\sum\nolimits_{k \in N(j)} h_k\Big) \Big/ |N(j)| \quad (4)$$

$\bar{h}_j$ is used to compute LSTM gates of joint j as below:

$$\begin{aligned} i_j &= \sigma(W^i x_j + U^i \bar{h}_j + b^i) \\ f_{jk} &= \sigma(W^f x_j + U^f h_k + b^f) \\ o_j &= \sigma(W^o x_j + U^o \bar{h}_j + b^o) \\ \tilde{c}_j &= tanh(W^c x_j + U^c \bar{h}_j + b^c) \\ c_j &= i_j \odot \tilde{c}_j + \sum\nolimits_{k \in N(j)} (f_{jk} \odot c_k) \\ h_j &= o_j \odot tanh(c_j) \end{aligned} \quad (5)$$

where i_j is the input gate, f_{jk} is the forget gate of parent k, o_j is the output gate, σ denotes the sigmoid function and $\odot$ denotes the element-wise multiplication. Note that our LSTM has different forget gates for different parent joint and the multiplication of each f_{jk} and c_k indicates the influence of parent k on current joint j.

2D pose encoding. As shown on the left of Fig. 2, the skeleton-LSTM utilize the global information from 2D skeleton S to predict the depth. In order to have a better representation of the 2D skeleton, we apply a multi-layer perceptron network shared by all joints to extract the global pose feature. The input feature of the skeleton-LSTM at joint j is $x_j^s = MP(\hat{S}_j)$ where $\hat{S}_j$ is the normalized 2D pose by subtracting each joint location by the current joint location $[x_j, y_j]$. The structure of the multi-layer perceptron is visualized in Fig. 3. It is trained together with the skeleton-LSTM.

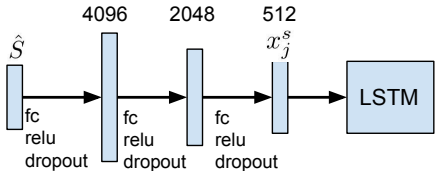

Figure 3. The multi-layer perceptron network for 2D pose encoding.

Image patch encoding. As shown on the right of Fig. 2, the patch-LSTM utilizes the local information from image patches of body parts for depth prediction. The input of LSTM unit at joint j is the encoded feature of the corresponding image patch around that joint. We use $x_j^p = CNN(I(x_j, y_j))$ to denote the input feature which is the last layer of a small ConvNet shared by all joints. The structure of the ConvNet is visualized in Fig. 4.

For the final LSTM at the second layer, the input feature at joint j is the element-wise max pooling of hidden states from skeleton-LSTM and patch-LSTM: $x_j = max(h_j^s, h_j^p)$. The real-value depth in log space $log(z_j)$ at each joint is predicted by attaching another fully-connected layer on the hidden state h_j: $log(z_j) = \sigma(W^z h_j + b^z)$.

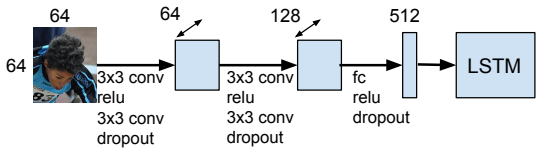

Figure 4. The convolutional network for image patch encoding.

3. Learning

The model weights that we need to learn include the weights of three LSTMs, and the weights of the 2D pose encoding network and image patch encoding network. The learning process consists of three phrases:

1) The skeleton-LSTM and skeleton encoding network are first pre-trained from Mocap data using the 2D-3D pose pairs with depth prediction loss. The RGB images are not needed and infinite 2D-3D pose pairs can be generated by projecting each 3D pose into 2D poses under different camera viewpoints.

2) The patch-LSTM and image encoding network are first pre-trained on RGB images from both MoCap dataset and in-the-wild 2D pose dataset with multi-task loss. Although the 2D pose dataset does not have depth data, they act as a regulariser with loss function of 2D pose regression.

3) The last step is to combine the two LSTMs in the second layer for end-to-end training.

3.1. Objective Function

The loss functions for depth regression at the above three phrases share the same formulation but use different input feature and hyper parameters. Inspired by [8], we define the loss based on both relative error and absolute error:

$$\begin{aligned} d_i &= log(z_i) - log(z_i^*) \\ d_{ij} &= (log(z_i) - log(z_j)) - (log(z_i^*) - log(z_j^*)) \\ L(z, z^*) &= \lambda \frac{1}{n} \sum_{i=1}^{n} d_i^2 + \beta \frac{1}{|E|} \sum_{(i,j) \in E} d_{ij}^2 \end{aligned} \tag{6}$$

where z is the vector of all depth values on joints, n is the number of joints and E denotes the set of edges in the tree structure. The first term of $L(z, z^*)$ is the mean squared error which enforces the absolute depth at each joint to be correct and the second term penalizes the difference of relative depth between each parent-child pairs. Instead of considering all pairwise depth relations in [8], we focus on the parent-child depth relations represented by edges in the tree structure of our model. The hyper parameters λ and β control the balance between absolute depth loss and relative depth loss. We set different λ and β for training skeleton-LSTM and patch-LSTM since they are good at minimizing different losses with different features.

3.2. Multi-task learning for patch-LSTM

As mentioned in Section 3.1, the patch-LSTM needs to be trained on image data with depth values of joints, and the images from Mocap data are captured from a highly constrained environment with small appearance variation which may lead to severe over-fitting. To decrease over-fitting, we argument training data using in-the-wild images with annotations of 2D poses from public pose datasets. Although the 2D pose datasets do not have depth, we apply the multi-task learning [7] to combine it with Mocap dataset in the same network. Specifically, we add another fully-connect layer on top of the hidden state of LSTM to regress the 2D joint locations which are normalized following [33]. The overall training loss is the sum of depth prediction loss which only operates on Mocap data and 2D pose regression loss which operates on both Mocap data and 2D pose data.

4. Results

Datasets. For empirical evaluation of our 3D pose estimation we use two datasets: Human3.6M dataset (H3.6M) [11] and UCLA Human-Human-Object Interaction Dataset (HHOI) [27]. The Human3.6M dataset is a large-scale dataset which includes accurate 3.6 million 3D human poses captured by Mocap system. It also includes synchronized videos and projected 2D poses from 4 cameras so the 2D-3D pose pairs are available. There are total 11 actors performing 15 actions such as Sitting, Waiting and Walking. This dataset is captured in a controlled indoor environment. The HHOI dataset contains human interactions captured by MS Kinect v2 sensor. It includes 3 types of human-human interactions: shake hands, high five and pull up and 2 types of human-object-human interactions: throw and catch, hand over a cup. There are 8 actors performing 23.6 instances per interaction on average. The data is collected in a common office with clutter background. For in-the-wild 2D pose dataset, we use the MPII-LSP-extended dataset [21] which is a combination of the extend LSP [12] and the MPII dataset [1]. After flipping each image horizontally, we get a total of 80000 images with 2D pose annotations.

Implementation details. We use the public deep learning library Keras [6] to implement our method. The training and testing are conducted on a single NVIDIA Titan X (Pascal) GPU. To train the skeleton-LSTM, we use the 2D-3D pose pairs down-sampled at 5 fps from Human3.6M dataset. Each 3D pose is projected onto 2D poses under 4 camera viewpoints. The 2D pose encoding network is stacked with skeleton LSTM for joint training with parameter $\lambda = 0.5$ and $\beta = 0.5$. To train the patch-LSTM, we use image frames down-sampled at 25 fps from Human3.6M and all images from MPII-LSP-extended dataset. The image patch encoding network is stacked with patch-LSTM for joint training with parameter $\lambda = 0.2$ and $\beta = 0.8$.

Methods	Direct	Discuss	Eat	Greet	Phone	Pose	Purchase	Sit	SitDown	Smoke	Photo	Wait	Walk	WalkDog	WalkTo	Mean
Yasin[39]	88.4	72.5	108.5	110.2	97.1	81.6	107.2	119.0	170.8	108.2	142.5	86.9	92.1	165.7	102.0	110.2
Gall[13]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	115.7
Rogez[23]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	88.1
our(s)	70.8	71.0	81.0	83.2	87.6	73.3	80.7	103.4	121.7	95.1	91.2	80.8	**71.8**	89.3	**73.0**	84.9
our(p)	93.5	88.0	116.7	105.4	111.3	80.0	99.7	136.7	173.2	111.5	117.6	86.9	89.1	118.8	97.5	108.4
our(s+p)	**62.8**	**69.2**	**79.6**	**78.8**	**80.8**	**72.5**	**73.9**	**96.1**	**106.9**	**88.0**	**86.9**	**70.7**	71.9	**76.5**	73.2	**79.5**

Table 1. Quantitative comparison of mean per joint errors (mm) on Human3.6M dataset (Protocol 1).

Methods	Direct	Discuss	Eat	Greet	Phone	Pose	Purchase	Sit	SitDown	Smoke	Photo	Wait	Walk	WalkDog	WalkTo	Mean
Tekin[30]	102.4	158.5	87.9	126.8	118.4	185.0	114.7	107.6	136.2	205.7	118.2	146.7	128.1	**65.9**	**77.2**	125.3
Zhou[41]	**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
Rogez[23]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	121.2
our(s+p)	90.1	**88.2**	**85.7**	**95.6**	**103.9**	**92.4**	**90.4**	117.9	136.4	**98.5**	**103.0**	**94.4**	**86.0**	90.6	89.5	**97.5**

Table 2. Quantitative comparison of mean per joint errors (mm) on Human3.6M dataset (Protocol 2).

After separate training of the two networks, we finally combine them with the final LSTM for end to end training using $\lambda = \beta = 0.5$. RMSprop [31] is used for mini-batch gradient descent and the learning rate is 0.00001 for all networks. The batch size is 128 for skeleton-LSTM and 64 for others.

Baseline. In addition to comparing our final system with state of the art methods, we also use two variations of our method as baselines : 1) To isolate the impact of image feature, we only keep the skeleton-LSTM and the 2D pose encoding network and train them jointly to predict the depth and then recover 3D pose. This baseline is denoted as 'ours(s)'; 2) We only keep patch-LSTM and image patch encoding network and it is denoted as 'ours(p)'.

4.1. Evaluation on Human3.6M Dataset

We compare our results with state of the art approaches in 3D pose estimation on Human3.6M dataset. We follow the evaluation protocol in [39]. The image frames and poses from subject S1, S5, S6, S7, S8 and S9 are used for training and S11 for testing. The testing data from S11 is down-sampled at 64fps and some poses without synchronized images are removed so the total testing set has 3612 poses. The training set has around 1.8 million 2D/3D poses with synchronized images. The 3D pose error is measured by the mean per joint position error (MPJPE) [11] at 13 joints up to a rigid transformation. We refer to this protocol by P1.

The quantitative results are presented in Table 1. The method 'our(s)' and 'our(p)' are two method variations and 'our(s+p)' is our final system. In all model variations, we apply the pre-trained off-the-shelf 2D pose estimator from [9] to detect 2D poses without any re-training or fine-tuning because it is easy for the model to overfit the Human3.6M dataset which is captured in a highly constrained environment with limited appearance variation.

Table 1 shows that our model variation 'our(s)' outperforms other approaches which demonstrates the powerfulness of predicting depth from only 2D pose. The human 2d pose can be seen as a strong cue to estimate the corresponding 3D pose. Although there are some ambiguities in the perspective projection, with only 2D pose features our model already captures helpful information to predict the depth of joint. This result also indicates that predicting the joint depth is more robust than predicting the whole 3D pose.

Our method variation 'our(p)' achieves similar results with [13] which also uses image patches to predict 3D joint locations. To train the patch-LSTM, we focus on the pairwise relative losses as shown in Eq. 6 because it is hard to predict the absolute depth with resized body part images. After integrating the skeleton-LSTM and patch-LSTM we further decrease the error to 79.5mm which outperforms the second best result by 9.8%.

We also report results for protocol 2 (P2) which is employed in [41, 30, 23]. The 2D/3D poses and image frames of subject S1, S5, S6, S7 and S8 are used for training and S7, S9 are used for testing. The estimated 3D pose and ground truth pose are aligned with their root locations and MPJPE is calculated without rigid transformation. The results of our final system and state-of-the-art approaches are presented in Table 2. Our method clearly outperforms the second best result [41] by 13.72% even though they use temporal information to help 3D pose estimation.

4.2. Evaluation on HHOI Dataset

To evaluate how our method can be generalized to data from a totally different environment, we train model on Human3.6M dataset and test it on HHOI dataset which is captured with Kinect in a casual environment. We pick 13 joints defined by Kinect and use mean per joint error as the evaluation metric. Each action instance is down-sampled at 10fps for efficient computation and both persons in each action are evaluated. We still use the focal length from Human3.6M to recover 3D poses and the poses are compared

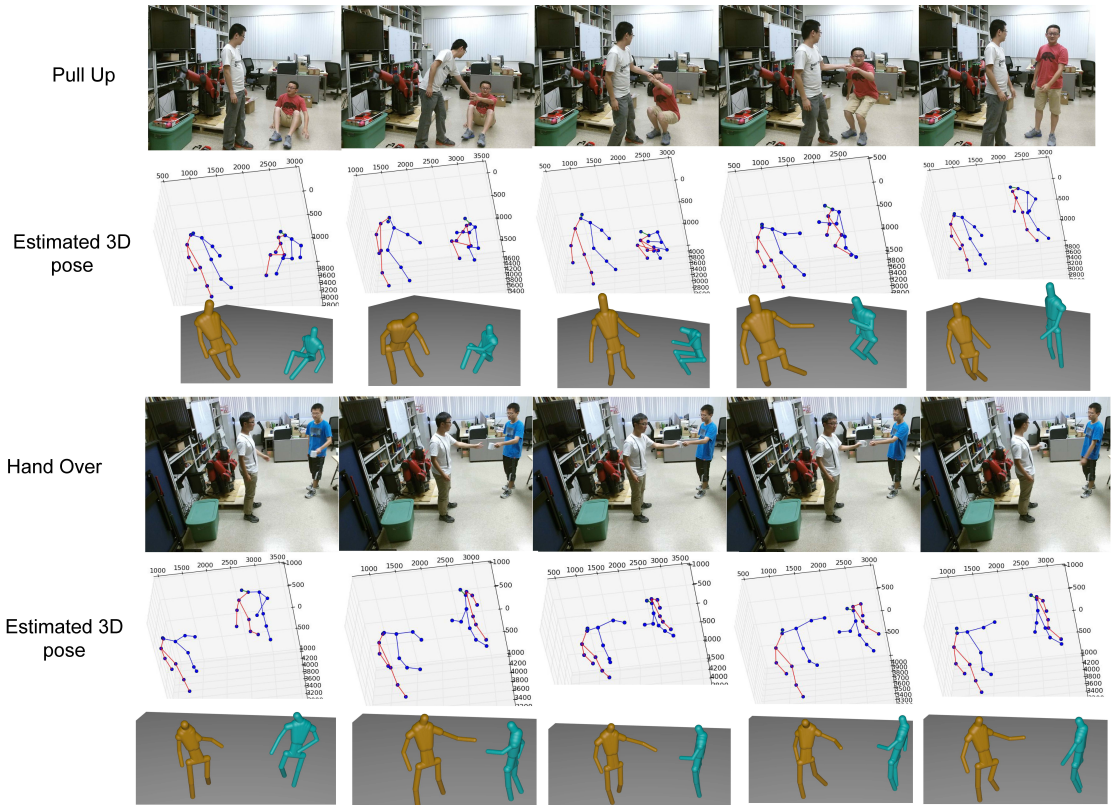

Figure 5. Qualitative results from HHOI dataset. We visualize ten frames and their estimated 3D poses from action 'Pull Up' and 'Hand Over'. Besides the original results, we show pose rendering results for better visualization.

up to a rigid transformation and also scale transformation. The method of [9] is used to produce 2D poses. Some qualitative results are presented in Fig. 5. For better visualization of 3D pose, we do pose rendering using the code released from [42]. The two poses at each frame are recovered independently so their relative depth may not be correct. We regress the relative mean depth between two persons using the 2d distance on y axis between two persons' feet.

There is no public code for recent methods compared in Human3.6M dataset so we implement another baseline 'Nearest' which match the predicted 2D pose with 2D poses from Human3.6M and select the depth from the 3D pose paired with the nearest 2D pose as the predicted depth. Note that the Kinect may produce unreasonable 3D poses because of occlusions and the evaluation with those poses cannot reflect true performance of compared methods, thus we looked at each action video and carefully select some of them for quantitative comparison. Specifically we keep all videos from 'PullUp' and 'HandOver', and a few videos from 'HighFive' and 'ShakeHands'. We select the smaller error calculated among the predicted pose and its flipped one due to the left-right confusion of Kinect. The quantitative results are summarized in Table 3. The action 'Pull Up' gets the biggest error among all actions due to the large pose variation. Our final model outperforms other baselines in three actions.

Method	PullUp	HandOver	HighFive	ShakeHands
Nearest	161.2	126.2	117.3	129.6
our(s)	139.8	105.2	98.4	**113.1**
our(p)	132.4	102.5	103.0	129.0
our(s+p)	**124.8**	**101.9**	**96.1**	118.6

Table 3. Quantitative comparison of mean per joint errors (mm) on HHOI dataset.

4.3. Diagnostic Experiments

To better justify the contribution of each component of our method, we do several diagnostic comparisons in the following. The Human3.6M and protocol 1 is used for all comparisons.

Effect of 2D poses. We first evaluate our method on 3D pose estimation when ground truth 2D poses are given and compare it with [39]. The results are presented in Table 4 (a) and indicate the potential of improvement when a more accurate 2D pose estimator is available.

We also consider the effect of performance of 2D pose estimation. To generate 2D poses with different errors, we

(a)

Method	Error
Yasin et al. [39]	70.3
our(s)	46.3
our(p)	79.3
our(s+p)	**42.9**

(b)

Method	No scale	scale
our(s)	84.9	80.6
our(p)	108.4	105.4
our(s+p)	79.5	74.0

Table 4. Quantitative comparison of mean per joint errors (mm) on Human3.6M (a) given ground truth 2D poses; (b) with and without scale transformation.

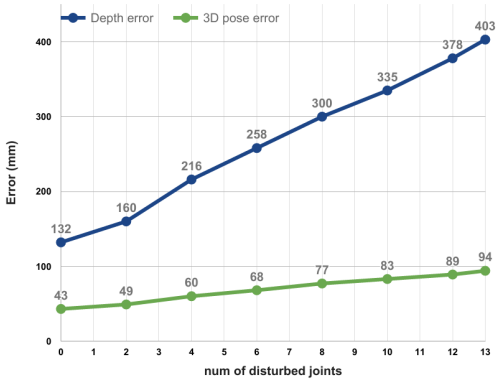

Figure 6. Depth and 3D pose error with different number of disturbed joints.

add disturbance to locations of different number of joints. Specifically, for each testing 2D pose, we randomly choose certain number of joints and add a uniform random noise in the range $[0, e], e = 0.1 \cdot max(h, w)$, where h and w are the height and width of the pose respectively. The absolute depth error and 3D pose error are calculated at each number of disturbed joints. The results are visualized in Fig. 6. Although the absolute depth error increases quickly with the error of 2D pose estimation, the 3D pose error increases slowly which indicates that the relative depth relations are not effected too much by the disturbed 2D pose.

Scale transformation. In general, it is impossible to estimate the global scale of the person from monocular image so we evaluate different variations of our model with a scale transformation. The results are presented in Table 4 (b) which show that there is a consistent improvement on all model variations when the scale transformation is allowed.

Different model structures. We consider the effect of model structure on the 3D pose estimation performance when only 2D skeleton features are used. We compare the following structures with the loss function defined in Eq. 6:

-**ske-econding**. We remove the tree-structure LSTM network and only keep the 2D pose encoding network. In this setting, the effect of explicit modeling of relations between joints are removed.

Method	Error
ske-encoding	113.0
ske-seq	89.0
ske-tree	**84.9**
whole-vgg	169.8
patch-seq	118.6
patch-tree	**108.4**

Table 5. Comparison between different model structures.

-**ske-seq**. We change the tree structure of the skeleton LSTM to a sequential chain structure with a fixed order of joints. It is impossible to evaluate all permutations of joints so we choose the order which is more similar to the tree structure: head-left limb-right limb.

-**ske-tree**. The skeleton-LSTM used in our final system.

We also evaluate the effect of the model structure when only body part image features are used. We remove 2D pose features and compare the three model variations:

-**whole-vgg**. We apply the VGG model [28] to predict the depth from the cropped image of the whole person instead of body part.

-**patch-seq**. It has the same sequential structure as ske-seq.

-**patch-tree**. The patch-LSTM used in our final system.

The results are shown in Table 5. The method with LSTM network boost performance a lot on both skeleton features (84.9 vs 113.0) and image patch features (108.4 vs 169.8) which demonstrates that the modeling of relationships between joints are essential for predicting depth. The comparison between sequential chain structure and tree structure demonstrates that the latter is more appropriate for modeling human skeleton than the former on both features.

5. Conclusions

In this paper, we propose a framework to predict the depth of human joints and recover the 3D pose. Our approach is built on a two level hierarchy of LSTM by utilizing two cues: the 2D skeleton feature which is captured by skeleton-LSTM and image feature of body part which is captured by patch-LSTM. The whole framework can be trained end to end and it allows any off-the-shelf 2D pose estimator to be plugged in. The experiments demonstrate our better performance qualitatively and quantitatively. In the future, we plan to extend this work to video-domain and combine it with 3D scene reconstruction by considering temporal constraints and person-object relations.

Acknowledgment

This research was supported by grants DARPA XAI project N66001-17-2-4029, ONR MURI project N00014-16-1-2007, NSF IIS-1423305, and NSFC 61503297.

References

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

[2] A.Yao, G. J, and L. V. Gool. Coupled action recognition and pose estimation from multiple views. *IJCV*, 100(1):16–37, 2012.

[3] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[4] C.-H. Chen and D. Ramanan. 3d human pose estimation = 2d pose estimation + matching. In *CVPR*, 2017.

[5] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *NIPS*, 2016.

[6] F. Chollet. https://github.com/fchollet/keras. 2015.

[7] R. Collobert and J. Weston. Unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, 2008.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[9] S. X. Haoshu Fang and C. Lu. RMPE: Regional multi-person pose estimation. *arXiv preprint arXiv:1612.00137*, 2016.

[10] M. Hofmann and D. M. Gavrila. Multi-view 3d human pose estimation in complex environment. *IJCV*, 96(1):103–124, 2012.

[11] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 36(7):1325–1339, 2014.

[12] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, 2011.

[13] I. Kostrikov and J. Gall. Depth sweep regression forests for estimating 3d human pose from images. In *BMVC*, 2015.

[14] S. Li and A. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014.

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

[16] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015.

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

[18] B. Nie, C. Xiong, and S. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[19] S. Park, X. Nie, and S.-C. Zhu. Attribute and-or grammar for joint parsing of human pose, parts and attributes. *IEEE TPAMI*, (99), 2017.

[20] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *ICCV*, 2013.

[21] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016.

[22] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *ECCV*, 2012.

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

[24] B. Rothrock, S. Park, and S. Zhu. Integrating Grammar and Segmentation for Human Pose Estimation. In *CVPR*, 2013.

[25] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[26] J. Shotton, A. W. Fitzgibbo, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[27] T. Shu, M. S. Ryoo, and S. Zhu. Learning social affordance for human-robot interaction. In *IJCAI*, 2016.

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv technical report*, 2014.

[29] K. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

[30] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*, 2016.

[31] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. In *Coursera: Neural networks for machine learning*.

[32] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[33] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[34] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu. Cross-view action modeling, learning, and recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014.

[35] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015.

[36] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, pages 3395–3402, 2015.

[37] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *ECCV*, 2016.

[38] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *TPAMI*, 35(12):2878–2890, 2012.

[39] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall. A dualsource approach for 3d pose estimation from a single image. In *CVPR*, 2016.

[40] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3d pose estimation from a single depth image. In *ICCV*, 2011.

[41] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *CVPR*, 2016.

[42] Y. Zhu, C. Jiang, Y. Zhao, D. Terzopoulos, and S.-C. Zhu. Inferring forces and learning human utilities from videos. In *CVPR*, 2016.