

Recurrent 3D Pose Sequence Machines

Mude Lin, Liang Lin,* Xiaodan Liang, Keze Wang, Hui Cheng
School of Data and Computer Science, Sun Yat-sen University

linmude@foxmail.com, linliang@ieee.org, xiaodanl@cs.cmu.edu,
wangkeze@mail2.sysu.edu.cn, chenghui9@mail.sysu.edu.cn

Abstract

3D human articulated pose recovery from monocular image sequences is very challenging due to the diverse appearances, viewpoints, occlusions, and also the human 3D pose is inherently ambiguous from the monocular imagery. It is thus critical to exploit rich spatial and temporal long-range dependencies among body joints for accurate 3D pose sequence prediction. Existing approaches usually manually design some elaborate prior terms and human body kinematic constraints for capturing structures, which are often insufficient to exploit all intrinsic structures and not scalable for all scenarios. In contrast, this paper presents a Recurrent 3D Pose Sequence Machine (RPSM) to automatically learn the image-dependent structural constraint and sequence-dependent temporal context by using a multi-stage sequential refinement. At each stage, our RPSM is composed of three modules to predict the 3D pose sequences based on the previously learned 2D pose representations and 3D poses: (i) a 2D pose module extracting the image-dependent pose representations, (ii) a 3D pose recurrent module regressing 3D poses and (iii) a feature adaption module serving as a bridge between module (i) and (ii) to enable the representation transformation from 2D to 3D domain. These three modules are then assembled into a sequential prediction framework to refine the predicted poses with multiple recurrent stages. Extensive evaluations on the Human3.6M dataset and HumanEva-I dataset show that our RPSM outperforms all state-of-the-art approaches for 3D pose estimation.

1. Introduction

Though quite challenging, recovering the 3D full-body human pose from a monocular RGB image sequence has re-

*Corresponding author is Liang Lin. This work was supported by State Key Development Program under Grant 2016YFB1001004, NSFC-Shenzhen Robotics Projects(U1613211), and the Fundamental Research Funds for the Central Universities, and Guangdong Science and Technology Program under Grant 201510010126.

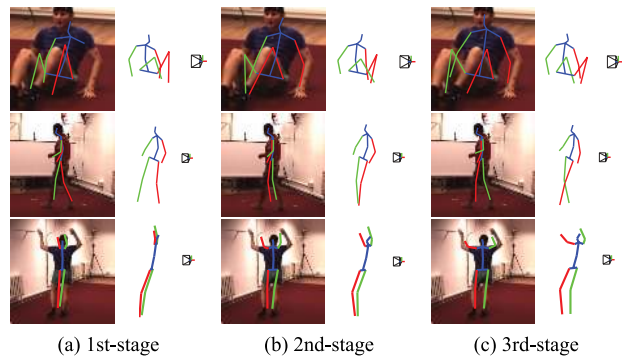


Figure 1: Some visual results of our approach (RPSM) on Human3.6M dataset. The estimated 3D skeletons are re-projected into the images and shown by themselves from the side view (next to the images). The figures from left to right correspond to the estimated 3D poses generated by the 1st-stage, 2nd-stage and 3rd-stage of RPSM, respectively. We can observe that the predicted human 3D joints are progressively corrected along with the multi-stage sequential learning. Best viewed in color.

cently attracted a lot of research interests due to its huge potentials on high-level applications, which includes human-computer interaction [10], surveillance [14], video browsing/indexing [6] and virtual reality [23].

Besides the challenges shared with 2D image pose estimation (e.g., large variation in human appearance, arbitrary camera viewpoints and obstructed visibilities due to external entities and self-occlusions), 3D articulated pose recovery from monocular imagery is much more difficult since 3D pose is inherently ambiguous from a geometric perspective [40], as shown in Fig. 1. To resolve all these issues, a preferable way is to investigate how to simultaneously enforce 2D spatial relationship, 3D geometry constraint and temporal consistency within one single model.

Recently, notable successes have been achieved for 2D pose estimation based on 2D part models coupled with 2D deformation priors, e.g., [35, 37], and the deep learning techniques, e.g., [32, 29, 34, 36]. However, these meth-

ods have not explored the 3D pose geometry that is crucial for 3D pose estimation. There has been some limited attempts on combining the image-based 2D part detectors, 3D geometric pose priors and temporal models for generating 3D poses [2, 39, 41, 30]. They mainly follow two kinds of pipelines: the first [41, 20] resorts to the model-based 3D pose reconstruction by using external 3D pose gallery, while the second pipeline [4, 40] focuses on elaborately designing human body kinematic constraints with the model training. These separate techniques and prior knowledge make their models very sophisticated. Hence, validating the effectiveness of their each component is also not straightforward. In contrast to all these mentioned methods, we introduce a completely data-driven approach that learns to integrate the 2D spatial relationship, 3D geometry and temporal smoothness for the network training in a fully differential way.

We propose a novel Recurrent 3D Pose Sequence Machine (RPSM) for estimating 3D human poses from a sequence of images. Inspired by the pose machine [22] and convolutional pose machine [34] architectures for 2D pose estimation, our RPSM proposes a multi-stage training to capture long-range dependencies among multiple body-parts for 3D pose prediction, and further enforce the temporal consistency between the predictions of sequential frames. Specifically, the proposed RPSM recursively refines the predicted 3D pose sequences by sensing what already achieved in the previous stages, i.e., 2D pose representations and previously predicted 3D poses. At each stage, our RPSM is composed by a 2D pose module, a feature adaption module, and a 3D pose recurrent module. These three modules are constructed by the integration of the advanced convolutional and recurrent neural networks to fully exploit spatial and temporal constraints, which makes our RPSM with multi-stages a differentiable architecture that can be trained in an end-to-end way.

As illustrated in Fig. 1, our RPSM enables to gradually refine the 3D pose prediction for each frame with multiple sequential stages, contributing to seamlessly learning the image-dependent constraint between multiple body parts and sequence-dependent context from the previous frames. Specifically, at each stage, the 2D pose module takes each frame and 2D feature maps produced in previous stages as inputs and progressively updates the 2D pose representations. Then a feature adaption module is injected to transform learned pose representations from 2D to 3D domain. The 3D pose recurrent module, constructed by a Long-Short Term Memory (LSTM) layer, can thus regress the 3D pose estimation by combining the three lines of information, i.e. the transformed 2D pose representations, 3D joint prediction from the previous stage and the memorized states from past frames. Intuitively, the 2D pose representations are conditioned on the monocular image which captures the spatial appearance and context information. The 3D joint

prediction implicitly encodes the 3D geometry structural information by aggregating multi-stage computation. Then temporal contextual dependency is captured by the hidden states of LSTM units, which effectively improves robustness of the 3D pose estimations over time.

The main **contribution** of this work is three-fold. i) We propose a novel RPSM model that learns to recurrently integrate rich spatial and temporal long-range dependencies using a multi-stage sequential refinement, instead of relying on specifically manually defined body smoothness or kinematic constraints. ii) Casting the recurrent network models to sequentially incorporate 3D pose geometry structural information is innovative in literature, which may also inspire other 3D vision tasks. iii) Extensive evaluations on the public challenging Human3.6M dataset [16] and HumanEva-I dataset [25] show that our approach outperforms existing methods of 3D human pose estimation by large margins.

2. Related work

Considerable research has addressed the challenge of 3D human pose estimation. Early research on 3D monocular pose estimation from videos involves frame-to-frame pose tracking and dynamic models that rely on Markov dependencies among previous frames, e.g. [33, 26]. The main drawbacks of these approaches are the requirement of the initialization pose and the inability to recover from tracking failure. To overcome these drawbacks, more recently approaches [2, 5] focus on detecting candidate poses in each individual frames and a post-processing step attempts to establish temporal consistent poses. Yasin *et al.* [38] proposed a dual-source approach for 3D pose estimation from a single image. They combined the 3D pose data from motion capture system with image source annotated with 2D pose. They transformed the estimation to a 3D pose retrieval problem. One major limitation of this approach is the time efficiency. It takes more than 20 seconds to process an image. Sanzari *et al.* [24] proposed a hierarchical Bayesian non-parametric model, which relies on a representation of the idiosyncratic motion of human skeleton joints groups and the consistency of the connected group poses is taken into account when reconstructing the full-body pose. Their approach achieved state-of-the-art performance on the Human3.6M [16] dataset.

Recently, deep learning has proven its ability in many computer vision tasks, such as the 3D human pose estimation. Li and Chan [19] firstly used the CNNs to regress the 3D human pose from monocular images and proposed two training strategies to optimize the network. Li *et al.* [20] proposed to integrate the structure-learning into deep learning framework, which consists of a convolutional neural network to extract image feature, and two following subnetworks to transform the image features and pose into a joint embedding. Tekin *et al.* [30] proposed to exploit

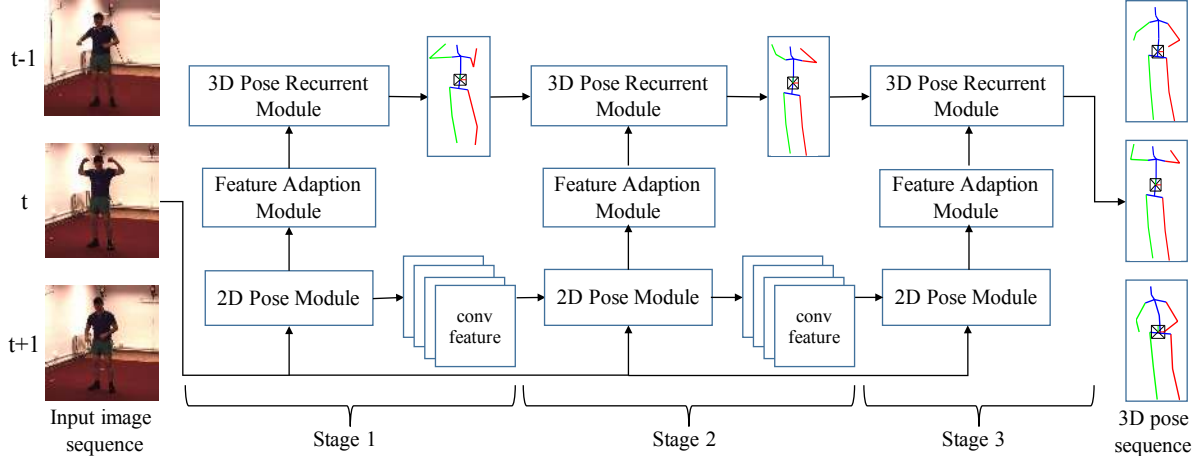


Figure 2: An overview of the proposed Recurrent 3D Pose Sequence Machine architecture. Our framework predicts the 3D human poses for all of the monocular image frames, and then sequentially refines them with multi-stage recurrent learning. At each stage, every frame of the input sequence is sequentially passed into three neural network modules: i) a 2D pose module extracting the image-dependent pose representations; 2) a feature adaption module for transforming the pose representations from 2D to 3D domain; 3) a 3D pose recurrent module predicting the human joints in 3D coordinates. Note that, the parameters of 3D pose recurrent module for all frames are shared to preserve the temporal motion coherence. Given the initial predicted 3D joints and 2D features from the first stage, we perform the multi-stage refinement to recurrently improve the pose accuracy. From the second stage, the previously predicted 17 joints (51 dimensions) and the 2D pose-aware features are also posed as the input of 2D pose module and 3D pose recurrent module, respectively. The final 3D pose sequence results are obtained after recurrently performing the multi-stage refinement.

motion information from consecutive frames and applied a deep learning network to regress the 3D pose. Zhou *et al.* [41] proposed a 3D pose estimation framework from videos that consists of a novel synthesis between a deep-learning-based 2D part detector, a sparsity-driven 3D reconstruction approach and a 3D temporal smoothness prior. Zhou *et al.* [40] proposed to directly embed a kinematic object model into the deep learning. Du *et al.* [9] introduced an additional built-in knowledge for reconstructing the 2D pose and formulated a new objective function to estimate 3D pose from detected 2D pose.

3. Recurrent 3D Pose Sequence Machines

As illustrated in Fig. 2, we propose a novel Recurrent 3D Pose Sequence Machine (RPSM) to resolve 3D pose sequence generation for monocular frames, which recurrently refines the predicted 3D poses at multiple stages. At each stage, RPSM consists of three consecutive modules: 1) 2D pose module to extract 2D pose-aware features; 2) feature adaption module to transform the representation from 2D to 3D domain; 3) 3D pose recurrent module to estimate 3D poses for each frame incorporating temporal dependency in the image sequence. These three modules are combined into a unified framework in each stage. The monocular image sequences are passed into multiple stages to gradually refine the predicted 3D poses. We train the network parameters recurrently at multiple stages in a fully end-to-end way.

3.1. Multi-stage Optimization

The 3D human pose is often represented as a set of P joints with 3D location relative to a root joint (*e.g.*, pelvis joint). Some exemplar poses are shown in Fig. 1. Our goal is to learn a mapping function that predicts the 3D pose sequence $\{S_1, \dots, S_T\}$ for the image sequence $\{I_1, \dots, I_T\}$, where I_t is the t -th frame containing a subject and $S_t \in \mathbb{R}^{3 \times P}$ is its corresponding 3D joint locations.

Aiming at obtaining the 3D pose S_t^k of the t -th frame at k -th stage, 2D pose module Ψ_p is first employed to extract the 2D pose-aware features $f_{2D}^{t,k}$ for each image by taking the image I_t and the previously 2D pose-aware features $f_{2D}^{t,k-1}$ as the input. Then the extracted 2D pose-aware features $f_{2D}^{t,k}$ are fed into the feature adaption module Ψ_a to generate adapted features $f_{3D}^{t,k}$. Finally, the 3D pose S_t^k is predicted according to the input of 3D pose recurrent module Ψ_r , which is composed of $f_{3D}^{t,k}$, the previously predicted 3D pose S_t^{k-1} and the hidden states H_{t-1}^k learned from the past frames. Formally, the $f_{2D}^{t,k}$, S_t^k , $f_{3D}^{t,k}$ of the t -th stage at k -th stage are formulated as,

$$\begin{aligned}
 f_{2D}^{t,k} &= \Psi_p(I_t, f_{2D}^{t,k-1}; W_p), \\
 f_{3D}^{t,k} &= \Psi_a(f_{2D}^{t,k}; W_a), \\
 S_t^k &= \Psi_r(f_{3D}^{t,k}, H_{t-1}^k, S_t^{k-1}; W_r),
 \end{aligned} \tag{1}$$

where W_p , W_a , W_r are network parameters of Ψ_p , Ψ_a , Ψ_r , respectively. At the first stage, the $f_{2D}^{t,0}$, S_t^0 are set as

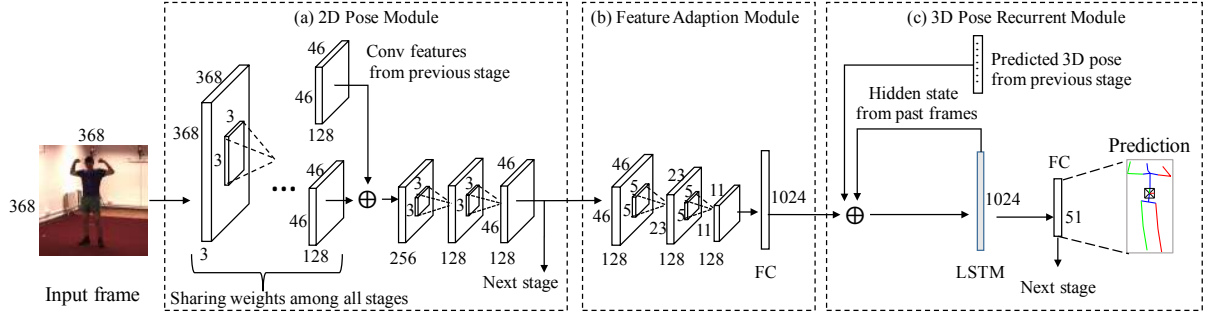


Figure 3: Detailed network architecture of our proposed RPSM at the k -th stage. An input frame with the 368×368 size is subsequently fed into 2D pose module, feature adaption module and 3D pose recurrent module to predict the locations of 17 joint points (51 dimensions output). The 2D pose module consists of 15 shared convolution layers across all stages and 2 specialized convolution layers for each stage. The specialized convolution layers take the shared features and the 2D pose-aware features at previous stage as the input, and output specialized features to the feature adaption module as well as the next stage. The feature adaption module consists of two convolution layers and one fully-connected layer with 1024 units. Finally, the adapted features, the hidden states of the LSTM layer and previously predicted 3D poses are concatenated together as the input of 3D pose recurrent module to produce the 3D pose of each frame. The symbol \oplus means the concatenation operation.

	1	2	3	4	5	6	7	8	9
Layer Name	conv1_1	conv1_2	max_1	conv2_1	conv2_2	max_2	conv3_1	conv3_2	conv3_3
Channel (kernel-stride)	64(3-1)	64(3-1)	64(2-2)	128(3-1)	128(3-1)	128(2-2)	256(3-1)	256(3-1)	256(3-1)
	10	11	12	13	14	15	16	17	18
Layer Name	conv3_4	max_3	conv4_1	conv4_2	conv4_3	conv4_4	conv4_5	conv4_6	conv4_7
Channel (kernel-stride)	256(3-1)	256(2-2)	512(3-1)	512(3-1)	256(3-1)	256(3-1)	256(3-1)	256(3-1)	128(3-1)

Table 1: Details of the shared convolutional layers in 2D pose module.

zero of the same size with those of other stages, and H_0^k is set to be a vector of zeros. The 3D pose sequence $\{S_1^K, S_2^K, \dots, S_T^K\}$ estimated by the last K -th stage stage is the final prediction. The sequential refinement procedure of our RPSM enables the gradually updating of the network status to better learn the mapping between the image sequence and 3D pose sequence.

3.2. 2D Pose Module

The goal of the 2D pose module is to encode each frame in the monocular sequence with a compact representation of the pose information, *e.g.* the body shape of the human. As a matter of fact, the lower convolution layers often extract the common low-level information, which is a very basic representation of the human image. Hence, we divide our proposed 2D pose module into two parts: the shared convolution layers across all stages and specialized pose-aware convolution layers in each stage. The architecture of 2D pose module is illustrated in Fig. 3(a).

The shared convolution layers, *i.e.*, those before the concatenation operation shown in Fig. 3(a), consist of 15 convolutional layers and four max-pooling layer. The kernel size of all shared convolutional layers are set to 3×3 , and the four max-pooling layers are set to have 2×2 kernel with a stride of 2. The numbers of channels for the shared convolution layers from left to right in Fig. 3(a) are 64, 64, 128, 128, 256, 256, 256, 256, 512, 512, 256, 256, 256, 256, 256

and 128, respectively (please see Table 1 for more details). Moreover, we append the Rectified Linear Unit(ReLU) layers on all convolution layers.

Afterwards, the shared convolution features and the extracted 2D pose-aware features at the previous stage are concatenated and then fed into the last two convolution layers to generate the updated 2D pose-aware features in 2D pose module. By combining the previously learned 2D pose-aware features at previous stage, the discriminative capability of the extracted 2D pose-aware features can be gradually enhanced, leading to a better 3D pose prediction. The higher convolution layers (*i.e.*, the last 2 convolution layers in Fig. 3(a)) of 2D pose module often capture more structure-sensitive information, which should be specialized for the refinement at each stage. Thus, we train the network parameters of the last 2 layers independently across all stages. Finally, the 2D pose module takes the 368×368 image as the input and outputs $128 \times 46 \times 46$ 2D pose-aware feature maps for each image.

3.3. Feature Adaption Module

Based on the features extracted by the 2D pose module, the feature adaption module is employed to adapt the 2D pose representations into a adapted feature space for the later 3D pose prediction. As depicted in Fig. 3(b), the proposed feature adaption module consists of two convolutional layers and one fully connected layers. Each convo-

lution layer contains 128 different kernels with the size of 5×5 , a stride of 2, and a max pooling layer with a 2×2 kernel size and a stride of 2 is appended on the convolutional layers. Finally, the convolution features are fed to a fully connected layer with 1024 units to produce the adapted feature vector. In this way, the feature adapter module transforms the 2D pose-aware features into the adapted feature vector of 1024 dimensions.

3.4. 3D Pose Recurrent Module

Given the adapted features for all frames, we propose a 3D pose sequence module to sequentially predict the 3D pose sequence. In this way, the rich temporal motion patterns between frames can be effectively incorporated into the 3D pose prediction. Note that Long Short-Term Memory (LSTM) [15] has proved better performance on exploiting temporal correlations than vanilla recurrent neural network in many tasks, *e.g.*, speech recognition [12] and video description [8]. In our RPSM, the 3D pose recurrent module resorts to the LSTM layers to capture the temporal dependency in monocular sequence for refining the 3D pose prediction for each frame.

As illustrated in Fig 3(c), the 3D pose recurrent module is constructed by one LSTM layer with 1024 hidden cells and an output layer that predicts the location of $P = 17$ joint points of the human. In particular, the hidden states learned by the LSTM layers are capable of implicitly encoding the temporal dependency across different frames of the input sequence. As formulated in Eq. (1), the adapted features, the previous hidden states and the previous 3D pose predictions are concatenated together as the current input of 3D poses recurrent module. Incorporating the previous 3D pose prediction at each stage endows our RPSM the ability of gradually refining the pose predictions.

4. Model Training and Testing

In the training phase, our RPSM enforces the 3D pose sequence prediction loss for all frames at all stages, which is defined as the Euclidean distances between the prediction for all P joints and ground truth:

$$\mathcal{L} = \sum_{k=1}^K \alpha_k \sum_{t=1}^T \|S_t^k - S_t^*\|_2^2, \quad (2)$$

where K is the number of stages, T is the length of an image sequence, S_t^* is the ground-truth 3D pose for t -th frame, and α_k is the loss weight for each stage.

The 2D Pose Module is first pretrained with MPII Human Pose dataset [1], since this dataset provides a larger variant of 2D pose data. Specifically, we temporally build up an extra convolution layer upon the public shared layers of 2D Pose Module to generate heat maps (joint confidence) as [31], which denote pixel-wise confidence maps of the

body joints. Then we exploit the MPII Human Pose dataset [1] to pretrain the tailored 2D Pose Module via the stochastic gradient decent algorithm. As for the whole framework, the ADAM [17] strategy is employed for parameter optimization.

In order to obtain sufficient samples to train the 3D pose recurrent module, we propose to decompose one long monocular image sequence into several small equal clips with C frames. According to the Eq. (2), we integrally fine-tune the parameters of 3D pose recurrent module, the feature adaption module and the specialized convolutional layers of the 2D pose module in a multi-stage optimization manner. In this way, the feature adaption module can learn the adapted feature representation according to the Eq. (2) for the further 3D pose estimation.

In the testing phase, every frame of the input image sequence is processed by our proposed RPSM in a stage-by-stage manner. In the end, after the final stage refinement, we output the 3D pose prediction.

5. Experiments

5.1. Experimental Settings

We perform the extensive evaluations on two publicly available datasets: Human3.6M [16] and HumanEva-I [25].

Human3.6M dataset. The Human3.6M dataset is a recently published dataset, which provides 3.6 million 3D human pose images and corresponding annotations in a controlled laboratory environment. It captures 11 professional actors performing in 15 scenarios under 4 difference viewpoints. In the following experiments, we strictly follow the same data partition protocol as in previous works [41, 20, 40, 30, 9, 24]. The data from five subjects (S1,S5,S6,S7,S8) is for training and two subjects (S9,S11) is for testing. Note that to increase the number of training samples, the sequences from different viewpoints of the same subject are treated as distinct sequences. Through downsampling the frame rate from 50FPS to 2FPS, 62,437 human pose images (104 images per sequence) are obtained for training while 21,911 images for testing (91 images per sequence). To be more general, our RPSM is trained on training samples from all 15 actions instead of exploiting individual action like [41, 20].

HumanEva-I dataset. The HumanEva-I dataset contains video sequences of four subjects performing six common actions(*e.g.*, walking, jogging, boxing *etc.*), and it also provides the 3D pose annotation for each frame in the video sequences. We train our RPSM on training sequences of the subject 1, 2 and 3 and test on the ‘validation’ sequence in the same protocol as [38, 30, 28, 27, 18, 3, 21, 33]. Similar as the Human3.6M dataset, the data from different camera viewpoints is also regarded as different training samples. Note that we have not downsampled the video sequences to

Method	Direction	Discuss	Eating	Greet	Phone	Pose	Purchase	Sitting	SitDown	Smoke	Photo	Wait	Walk	WalkDog	WalkPair	Avg.
LinKDE [16]	132.71	183.55	132.37	164.39	162.12	150.61	171.31	151.57	243.03	162.14	205.94	170.69	96.60	177.13	127.88	162.14
Li <i>et al.</i> [20]	-	136.88	96.94	124.74	-	-	-	-	-	-	168.68	-	69.97	132.17	-	-
Tekin <i>et al.</i> [30]	102.39	158.52	87.95	126.83	118.37	114.69	107.61	136.15	205.65	118.21	185.02	146.66	65.86	128.11	77.21	125.28
Zhou <i>et al.</i> [41]	87.36	109.31	87.05	103.16	116.18	106.88	99.78	124.52	199.23	107.42	143.32	118.09	79.39	114.23	97.70	113.01
Zhou <i>et al.</i> [40]	91.83	102.41	96.95	98.75	113.35	90.04	93.84	132.16	158.97	106.91	125.22	94.41	79.02	126.04	98.96	107.26
Du <i>et al.</i> [9]	85.07	112.68	104.90	122.05	139.08	105.93	166.16	117.49	226.94	120.02	135.91	117.65	99.26	137.36	106.54	126.47
Sanzari <i>et al.</i> [24]	48.82	56.31	95.98	84.78	96.47	66.30	107.41	116.89	129.63	97.84	105.58	65.94	92.58	130.46	102.21	93.15
Ours	58.02	68.16	63.25	65.77	75.26	61.16	65.71	98.65	127.68	70.37	93.05	68.17	50.63	72.94	57.74	73.10

Table 2: **Quantitative comparisons on Human3.6M dataset** using 3D pose errors (in millimeter) for different actions of subjects 9 and 11. The entries with the smallest 3D pose errors for each category are bold-faced. Our RPSM achieves the significant improvement over all compared state-of-the-art approaches, *i.e.* reduces mean error by **21.52%**.

Methods	Walking				Jogging				Boxing			
	S1	S2	S3	Avg.	S1	S2	S3	Avg.	S1	S2	S3	Avg.
Simo-Serra <i>et al.</i> [28]	99.6	108.3	127.4	111.8	109.2	93.1	115.8	108.9	-	-	-	-
Radwan <i>et al.</i> [21]	75.1	99.8	93.8	89.6	79.2	89.8	99.4	89.5	-	-	-	-
Wang <i>et al.</i> [33]	71.9	75.7	85.3	77.6	62.6	77.7	54.4	71.3	-	-	-	-
Du <i>et al.</i> [9]	62.2	61.9	69.2	64.4	56.3	59.3	59.3	58.3	-	-	-	-
Simo-Serra <i>et al.</i> [27]	65.1	48.6	73.5	62.4	74.2	46.6	32.2	56.7	-	-	-	-
Bo <i>et al.</i> [3]	45.4	28.3	62.3	45.3	55.1	43.2	37.4	45.2	42.5	64.0	69.3	58.6
Kostrikov <i>et al.</i> [18]	44.0	30.9	41.7	38.9	57.2	35.0	33.3	40.3	-	-	-	-
Tekin <i>et al.</i> [30]	37.5	25.1	49.2	37.3	-	-	-	-	50.5	61.7	57.5	56.6
Yasin <i>et al.</i> [38]	35.8	32.4	41.6	36.6	46.6	41.4	35.4	38.9	-	-	-	-
Ours	26.5	20.7	38.0	28.4	41.0	29.7	29.1	33.2	39.4	57.8	61.2	52.8

Table 3: **Quantitative comparisons on HumanEva-I dataset** using 3D pose errors (in millimeter) for the “Walking”, “Jogging” and “Boxing” sequences. ‘-’ indicates the corresponding method has not reported the accuracy on that action. The entries with the smallest 3D pose errors for each category are bold-faced. Our RPSM outperforms all the compared state-of-the-art methods by a clear margin.

obtain more samples for training.

Evaluation metric. Following [41, 9, 30], we employ the popular *3D pose error* metric [28], which calculates the Euclidean errors on all joints and all frames up to translation. In the following section, we will report the 3D pose error metric for all the experimental comparisons and analyses.

Implementation Details: Our RPSM is implemented by using Torch7 [7] deep learning toolbox. We follow [12] to build the LSTM memory cells, except that the peephole connections between cell and gates are omitted. The loss weights α_k for each stage are all set to 1. In total, three-stage refinements are performed for all our experiments since only unnoticeable performance difference is observed using more stages. Following [41, 20], the input image is cropped around the human. To keep the human ratio, we crop a square image of the subject from the image according to the bounding box provided by the dataset. Then we resize the image region inside the bounding box into 368×368 resolution before feeding it into the network. Moreover, we augment the training data only by random scaling with factors in $[0.9, 1.1]$. Note that to transform the absolute locations of joint points into the $[0, 1]$ range, max-min normalization strategy is applied. In the testing phase, the predicted 3D pose is transform to the origin scale

according to the maximum and minimal value of the pose from training frames. During the training, the Xavier initialization method [11] was used to initialize the weights of our RPSM. The decay is set as $1e^{-4}$ and base learning rate of $1e^{-3}$ is employed for training. It took about 2 days to train a 3-stage RPSM with 50 epochs on single NVIDIA GeForce GTX TITAN X with 12GB memory. In the testing phase, it takes about 50 ms to process an image.

5.2. Comparisons with state-of-the-art methods

Comparison on Human3.6M: We compare our RPSM with the state-of-the-art methods on Human3.6M [16] and HumanEva-I [25] dataset. These state-of-the-art methods are LinKDE [16], Tekin *et al.* [30], Li *et al.* [20], Zhou *et al.* [41] (CNN based), Zhou *et al.* [40], Du *et al.* [9] and Sanzari *et al.* [24].

The results are summarized in Table 2. As one can see from Table 2, our proposed RPSM model significantly outperforms all compared methods with mean error reduced by **31.85%** compared with [40] and **21.52%** compared with [24]. Note that some compared methods, *e.g.*, [20, 30, 9, 41, 40], also employ deep learning techniques. Especially, Zhou *et al.* [40]’s method has used the recently published Residual Network [13]. This superior performance achieved by RPSM demonstrates that utilizing multi-

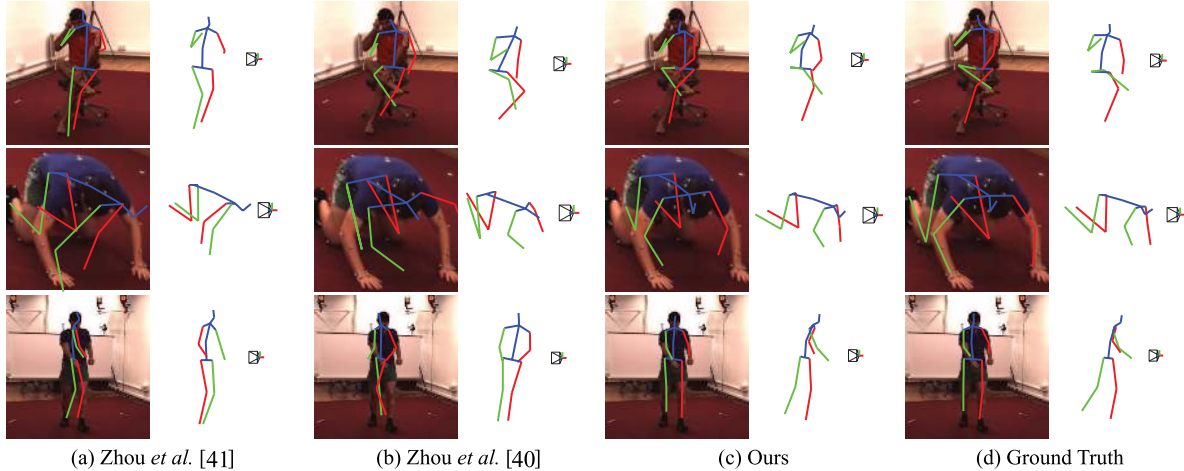


Figure 4: Empirical study on the qualitative comparisons on Human3.6M dataset. The 3D pose are visualized from the side view and the camera are also depicted. Zhou *et al.* [41], Zhou *et al.* [40], our RPSM and the ground truth are illustrated from left to right, respectively. Our RPSM achieves much more accurate estimations than the methods of Zhou *et al.* [41] and Zhou *et al.* [40]. Best view in color.

Method	Direction	Discuss	Eating	Greet	Phone	Pose	Purchase	Sitting	SitDown	Smoke	Photo	Wait	Walk	WalkDog	WalkPair	Avg.
RPSM-1-stage	62.89	74.74	67.86	73.33	79.76	67.48	76.19	100.21	148.03	75.95	100.26	75.82	58.03	78.74	62.93	80.15
RPSM-2-stage	58.96	68.50	65.64	68.18	78.41	62.82	67.04	100.63	136.72	73.35	96.87	67.96	51.64	77.27	59.31	75.55
RPSM-3-stage	58.02	68.16	63.25	65.77	75.26	61.16	65.71	98.65	127.68	70.37	93.05	68.17	50.63	72.94	57.74	73.10
RPSM_1stage_seq_1	70.46	83.36	76.46	80.96	88.14	76.00	92.39	116.62	163.14	85.87	111.46	83.60	65.38	95.10	73.54	90.83
RPSM_3stage_seq_1	61.94	75.84	65.25	71.28	79.39	67.73	77.88	105.47	153.58	76.01	101.84	74.12	56.07	85.63	64.78	81.12
RPSM_1stage_seq_5	62.89	74.74	67.86	73.33	79.76	67.48	76.19	100.21	148.03	75.95	100.26	75.82	58.03	78.74	62.93	80.15
RPSM_1stage_seq_10	66.73	76.82	73.57	76.56	84.80	70.57	75.44	110.70	143.10	80.35	103.61	75.66	58.52	80.55	66.19	82.88
RPSM-3-stage_no_MPII	91.58	109.35	93.28	98.52	102.16	93.87	118.15	134.94	190.6	109.39	121.49	101.82	88.69	110.14	105.56	111.3
RPSM-3-stage_sharing	58.36	66.52	63.37	64.5	72.22	59.39	63.9	90.73	129.99	68.26	93.86	65.22	48.47	70.53	56.26	71.44

Table 4: Top five rows: empirical study on different number of refinement stages. Middle two rows: empirical comparisons by different sequence lengths (i.e., 1, 5, 10) for each clip. Note that the results are evaluated by a single-stage RPSM. Bottom two rows: performance of RPSM variants. The entries with the smallest 3D pose errors on Human3.6m dataset for each category are bold-faced.

stage RPSM is simple yet powerful in capturing complex contextual features within images and learning temporal dependency within image sequences, which are critical for estimating 3D pose sequence.

Comparison on HumanEva-I: On this dataset, we compare our RPSM against methods which rely on several kinds of separate processing steps. These methods include discriminative regressions[3, 18], 2D pose detectors based [28, 27, 33, 38], CNN-based regressions [30]. For fair comparison, our RPSM also predicts the 3D pose consisting of 14 joints, i.e., left/right shoulder, elbow, wrist, left/right hip knee, ankle, head top and neck, as [38].

Table 3 illustrates the performance comparisons between our RPSM with compared methods. It is obvious that our RPSM model obtains substantially lower 3D pose errors than the compared methods, and achieves new state-of-the-art performance on all *Walking*, *Jogging* and *Boxing* sequences. In addition, in terms of the time efficiency, compared with [3] which takes around three minutes per image and [38] which takes more than 25 seconds per im-

age, our RPSM model only costs 50ms per image. This demonstrate the effectiveness and efficiency of our proposed RPSM model.

5.3. Component Analysis

Effectiveness of multi-stage refinement: To validate the superiority of the proposed multi-stage refinement of our RPSM, we conduct the following experiment: employing one, two, three stages for human pose estimation and denote them as “RPSM-1-stage”, “RPSM-2-stage” and “RPSM-3-stage”. The evaluations are performed on the Human3.6M dataset from the qualitative and quantitative aspects. The top five rows of Table 4 illustrates the comparisons of estimating 3D pose errors for using different number of stages. As one can see from Table 4, the performance increases monotonically within 3 stages. Moreover, the single/multi-stage performance without temporal dependency is also compared in Table. 4 (denoted as “RPSM-1stage_seq_1” and “RPSM-3stage_seq_1”, respectively). As illustrated in Table. 4, RPSM-3stage_seq_1 has achieved

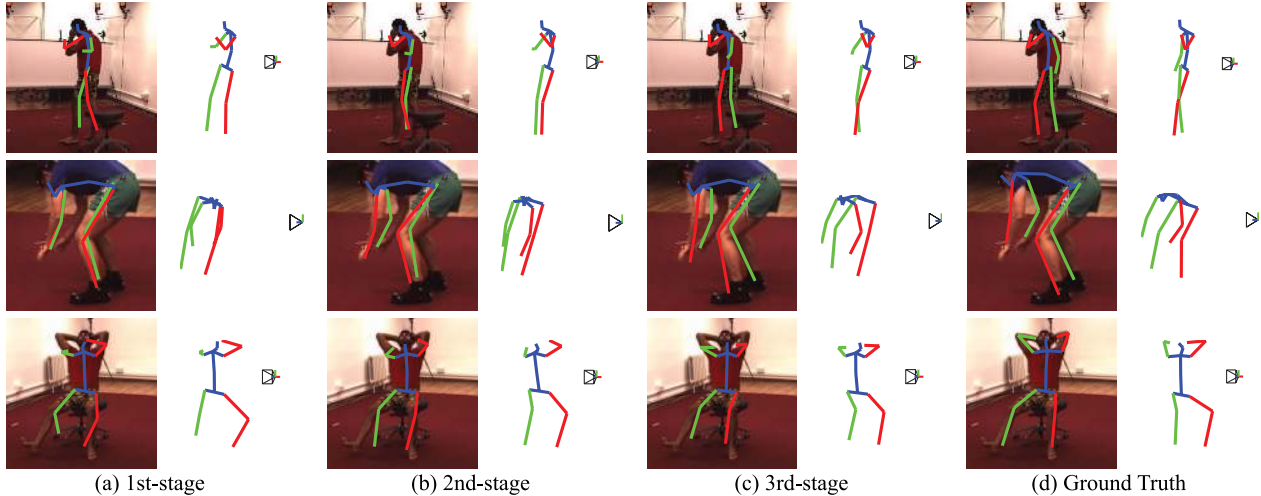


Figure 5: Qualitative comparisons of different stage refinement on Human3.6M dataset. The estimated 3D skeletons are reprojected into the images and shown by themselves from the side view (next to the images). The figures from left to right correspond to the estimated 3D poses generated by the 1st-stage, 2nd-stage, 3rd-stage of our RPSM and ground truth, respectively. We can observe that the predicted human 3D joints are progressively corrected along with the multi-stage sequential learning. Best viewed in color.

much lower 3D pose errors than RPSM-1stage_seq_1 (81.12 vs 90.83). This validates that the effectiveness of multi-stage refinement even when temporal information is ignored. Thanks to the exploited richer contextual information, our RPSM can learn more robust 2D pose-aware features and the representation of 3D pose sequences. Exemplar visual results on three different stages are shown in Fig. 5. It can be seen that the joint predictions are progressively corrected by performing multi-stage refinement.

Pre-training and Weight Sharing: To evaluate the performance without pre-training, we have only employed Human3.6m 2D pose data and annotations to train the 2D pose module. We denote this version of our RPSM as “RPSM-3-stage_no_MPII”. The result is reported in the bottom two rows of Table. 4. As one can see from Table. 4, RPSM-3-stage_no_MPII performs quite worse than RPSM-3-stage. This may be due to that Human3.6m 2D pose data, compared with MPII dataset, is less challenging for CNN to learn a rich 2D pose presentation. Note that according to the bottom row of Table. 4, the performance of sharing all layers in the 2D pose module (denoted as “RPSM-3stage-sharing”) is slightly better than the partially sharing one (denoted as “RPSM-3-stage”). However, the training time will significantly increased. Thus, we decide to choose partially sharing manner.

Importance of temporal dependency: To study the effectiveness of incorporating temporal dependency, we also evaluate the variants of our single-stage RPSM using different clip lengths, i.e., 1, 5 and 10, named as “RPSM-1stage_seq_ C ”, where C denotes the frame number of clips for training. Note that, when C is equal to 1, no temporal information is considered and thus the recurrent LSTM layer

in 3D pose errors is replaced with a fully connected layer with the same units as the LSTM. Results of using different clip length are reported in Table 4. From the comparison results, the importance of temporal dependency is well demonstrated. Considering temporal dependency methods (i.e., RPSM_1stage_seq_5 and RPSM_1stage_seq_10) all outperform the RPSM_1stage_seq_1 in a clear margin (about 10% reduction of the mean joint errors on the Human3.6M dataset). The minor performance difference between RPSM_1stage_seq_5 and RPSM_1stage_seq_10 may be due to effect of temporal inconsistency, which has higher probability to occur in long clips. Moreover, it also should be noted that “RPSM_1stage_seq_1” shows superiority over all state-of-the-art approaches owing to the contribution of the proposed 2D pose module and feature adaption module.

6. Conclusion

We have proposed a novel Recurrent 3D Pose Sequence Machines (RPSM) for estimating 3D human pose from a sequence of monocular images. Through the proposed unified architecture with 2D pose, feature adaption and 3D pose recurrent modules, our RPSM can learn to recurrently integrate rich spatio-temporal long-range dependencies in an implicit and comprehensive way. We also proposed to employ multiple sequential stages to refine the estimation results via the 3D pose geometry information. The extensive evaluations on two public 3D human pose dataset validate the effectiveness and superior performance of the our RPSM. In future work, we will extend the proposed framework for other sequence-based human centric analysis such as human action and activity recognition.

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. 5
- [2] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *CVPR*, 2010. 2
- [3] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *IJCV*, 87(1-2):28–52, 2010. 6, 7
- [4] 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 *ECCV*, 2016. 2
- [5] X. Burgos-Artizzu, D. Hall, P. Perona, and P. Dollár. Merging pose estimates across space and time. In *BMVC*, 2013. 2
- [6] L. Chen, Y. Zhou, and D. M. Chiu. Video browsing—a study of user behavior in online vod services. In *International Conference on Computer Communication and Networks (ICCCN)*, 2013. 1
- [7] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011. 6
- [8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 5
- [9] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. S. Kankanhalli, and W. Geng. Marker-less 3d human motion capture with monocular image sequence and height-maps. In *ECCV*, 2016. 3, 5, 6, 7
- [10] A. Errity. Human–computer interaction. *An Introduction to Cyberpsychology*, page 241, 2016. 1
- [11] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010. 6
- [12] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, 2014. 5, 6
- [13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7
- [14] C. Held, J. Krumm, P. Markel, and R. P. Schenke. Intelligent video surveillance. *Computer*, 3(45):83–84, 2012. 1
- [15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 5
- [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. *PAMI*, 36(7):1325–1339, 2014. 2, 5, 6
- [17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014. 5
- [18] I. Kostrikov and J. Gall. Depth sweep regression forests for estimating 3d human pose from images. In *BMVC*, 2014. 6, 7
- [19] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014. 3
- [20] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *ICCV*, 2015. 2, 3, 5, 6, 7
- [21] I. Radwan, A. Dhall, and R. Goecke. Monocular image 3d human pose estimation under self-occlusion. In *ICCV*, 2013. 6
- [22] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014. 2
- [23] H. Rheingold. *Virtual Reality: Exploring the Brave New Technologies*. Simon & Schuster Adult Publishing Group, 1991. 1
- [24] M. Sanzari, V. Ntouskos, and F. Pirri. Bayesian image based 3d pose estimation. In *ECCV*, 2016. 2, 5, 6, 7
- [25] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87(1-2):4–27, 2010. 2, 5, 6
- [26] L. Sigal, M. Isard, H. Haussecker, and M. J. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *IJCV*, 98(1):15–48, 2012. 2
- [27] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A Joint Model for 2D and 3D Pose Estimation from a Single Image. In *CVPR*, 2013. 6, 7
- [28] E. Simo-Serra, A. Ramisa, G. Alenyà, C. Torras, and F. Moreno-Noguer. Single image 3d human pose estimation from noisy observations. In *CVPR*, 2012. 6, 7
- [29] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013. 2
- [30] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*, 2016. 2, 3, 5, 6, 7
- [31] 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. 5
- [32] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 2
- [33] 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. 2, 6, 7
- [34] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. 2
- [35] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015. 2
- [36] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, 2016. 2
- [37] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 2
- [38] H. Yasin, U. Iqbal, B. Krüger, A. Weber, and J. Gall. A dual-source approach for 3d pose estimation from a single image. In *CVPR*, 2016. 2, 6, 7
- [39] F. Zhou and F. De la Torre. Spatio-temporal matching for human detection in video. In *ECCV*, 2014. 2

- [40] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. *arXiv preprint arXiv:1609.05317*, 2016. [2](#), [3](#), [5](#), [6](#), [7](#)
- [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. [2](#), [3](#), [5](#), [6](#), [7](#)