IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier XXX

An Adaptive Viewpoint Transformation Network for 3D Human Pose Estimation

GUOQIANG LIANG, XIANGPING ZHONG, LINGYAN RAN, YANNING ZHANG, (Senior Member, IEEE)

National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science and Engineering, Northwestern Polytechnical University, 710072, China

Corresponding author: Lingyan Ran (e-mail: lran@nwpu.edu.cn).

This work was supported in part by NSFC No. 61902321, the China Postdoctoral Science Foundation funded project under Grant 2019M653746, and the Fundamental Research Funds for Central Universities of China under Grant 31020182019gx007.

ABSTRACT Human pose estimation from a monocular image has attracted lots of interest due to its huge potential application in many areas. The performance of 2D human pose estimation has been improved a lot with the emergence of deep convolutional neural network. In contrast, the recovery of 3D human pose from an 2D pose is still a challenging problem. Currently, most of the methods try to learn a universal map, which can be applied for all human poses in any viewpoints. However, due to the large variety of human poses and camera viewpoints, it is very difficult to learn a such universal mapping from current datasets for 3D pose estimation. Instead of learning a universal map, we propose to learn an adaptive viewpoint transformation module, which transforms the 2D human pose to a more suitable viewpoint for recovering the 3D human pose. Specifically, our transformation module takes a 2D pose as input and predicts the transformation parameters. Rather than some hand-crafted criteria, this module is directly learned from the datasets and depends on the input 2D pose in testing phrase. Then the 3D pose is recovered from this transformed 2D pose. Since the difficulty of 3D pose recovery becomes smaller, we can obtain more accurate estimation results. Experiments on Human3.6M and MPII datasets show that the proposed adaptive viewpoint transformation can improve the performance of 3D human pose estimation.

INDEX TERMS 3D human pose estimation, adaptive viewpoint transformation, deep convolutional neural network

I. INTRODUCTION

HUMAN pose estimation is to estimate the 2D or 3D locations of human joints from images or videos. Specifically, this paper focuses on 3D human pose estimation from a monocular RGB image. Since there is no need of any specialized devices, like depth sensor, its possible application range is much wider than pose estimation from multi-views or RGB-D images. Due to its huge potential application in human motion prediction, action analysis and intelligent video surveillance [1]–[3], 3D human pose estimation from a monocular image has attracted more and more attention in recent years.

Due to the loss of depth information when projecting a person in real world to a 2D image space, it is an ill-posed problem to estimate the 3D pose from a 2D monocular image. Considering this nature, early research is restricted to some simplified settings, such as specified actions or fixed background. And the approaches either utilize example-based refinement or rely on very strong assumption including scaled orthographic cameras, or calibrated perspective cameras [4], [5]. Recently, inspired by the success of deep convolutional neural network (DCNN), lots of DCNN-based 3D human pose estimation methods are also proposed [6]–[10]. The 3D pose estimation methods can be roughly categorized into two kinds: one-step model and two-steps model. The former directly estimates the 3D pose from an image. In contrast, the latter first estimates 2D pose from an image then recovers 3D pose only using the predicted 2D pose. Compared with first kind, the two-steps methods have obtained better performance in wild environment. Since the first step can exploit the mature 2D estimation methods [11]–[13], current research priority including this paper focus on the second step.

In real applications, there is no constraint on the camera viewpoints. Therefore, for one 3D pose, the projected poses in 2D space will be highly different due to the large variation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013917, IEEE Access

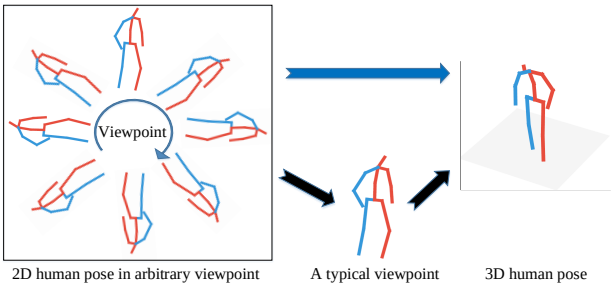

FIGURE 1. An illustrative example for our motivation. Instead of directly recovering the 3D pose from a 2D pose, we propose to first transform the 2D pose to a suitable viewpoint and then estimate the 3D pose.

of viewpoints. An illustrative example is shown in Fig. 1, where several possible 2D human poses in different viewpoints are corresponding to an identical 3D pose. Because of this many-to-one correspondence relationship, it is very difficult to accurately recover the 3D pose from different 2D poses. To address this problem, most of the current methods aim at learning a viewpoint-invariant model, which can recover the 3D pose from 2D pose in arbitrary viewpoint. In the early stage, this invariance is achieved through hand-crafted invariant feature, such as the SIFT descriptors [14]. Recently, many researchers resort to DCNN to learn a powerful model which can lift arbitrarily 2D pose to its corresponding 3D pose. For example, based on the spatial transformer networks [15], Haque et al. [16] proposed to embed a local patch into a viewpoint invariant feature space. However, it is very tough for a single model to learn such complex relationship from current datasets with limited samples. Therefore, these attempts do not obtain high performance.

Instead of learning a viewpoint-invariant model, we address this problem from another direction. We note that the different 2D human poses in Fig. 1 will become more similar after appropriate 2D rotation. Moreover, the relationship between the rotated 2D human pose and 3D pose is much simpler, which is easier to learn. Actually, in human vision system, people are habituated to rotate image virtually in the brain and then recover 3D human pose from the rotated 2D pose. Inspired by this, we propose to transform the 2D human pose prediction and then recover the 3D pose. Specifically, we design an adaptive viewpoint transformation module, which transforms the predicted 2D human pose to a more suitable viewpoint for recovering 3D human pose. This module just takes a 2D pose as input and outputs the transformation parameters. Rather than some hand-crafted criteria, our module is based on the DCNN and is directly learned from the datasets. In testing, it only depends on the input 2D pose. Then, a deep model is used for recovering the 3D pose from the transformed 2D pose. Since the relationship is simplified, the difficulty of 3D pose recovery process will be reduced, which will lead to more accurate estimation results. To evaluate the proposed method, we have conducted extensive experiments on the widely used datasets Human3.6M [17] and MPII [18]. The experimental results show the effectiveness of this method.

The rest of this paper is organized as follows: Section II reviews some very related works. In section III, we elaborate the proposed method. Experimental results and detailed analysis are in section IV. Finally, we draw the conclusion and describe the future work in section V.

II. RELATED WORKS

In this section, we will review some related works on 3D human pose estimation, viewpoint invariant model and dynamic adaptive network.

A. 3D HUMAN POSE ESTIMATION

Due to the loss of depth information and limited datasets, 3D human pose estimation from a monocular image is a challenging problem. Many methods belong to the one-step model, which estimates the 3D human pose directly from an image. Some methods are based on direct regression, especially based on the DCNN [6], [19], [20]. For example, Pavlakos et al. [21] extends the hourglass model in [11] to 3D case and design a coarse to fine approach to avoid the large dimension increase. Sun et al. [6] use integral regression to combine the heat-map and regression-based representation. Due to the absence of large well-annotated dataset and complexity of the mapping from a single image to 3D joints location, the performance of these one-step models is far from satisfactory.

To address the above issues, the two-steps methods are also widely used, which first estimates 2D human pose from an input image, and then lifts the predicted 2D pose to 3D human pose. Since the first step has obtained huge advancement, current researches focus on the second step. Chen et al. formulate the 3D recovery as a matching problem [8]. A simple yet effective network architecture is proposed in [9], which just contains several linear layers and residual blocks. Based on that the localization difficulty of joints are different, a coarse-to-fine model and a set of constraints is developed to gradually localize the joints [22]. In [23], the joints are divided into two groups. And attention model and random enhancement module are used to improve the performance.

Our method belongs to the second kind. Compared with the above methods, we propose to transform the predicted 2D pose to a suitable viewpoint then recover the 3D pose from the transformed 2D pose. We deem that it is easier to recover the 3D pose from transformed 2D pose than the original pose with larger viewpoint range. Note that it is easier to combine our framework with the newest methods for estimating the 3D pose from 2D pose to further improve the performance.

B. VIEWPOINT INVARIANT MODEL

Due to the camera viewpoint variation, a single object or human pose will show different appearance in 2D images. Therefore, the computer society is always trying to construct viewpoint invariant description or models. The early attempts focus on designing invariant features, like the SIFT, one of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013917, IEEE Access

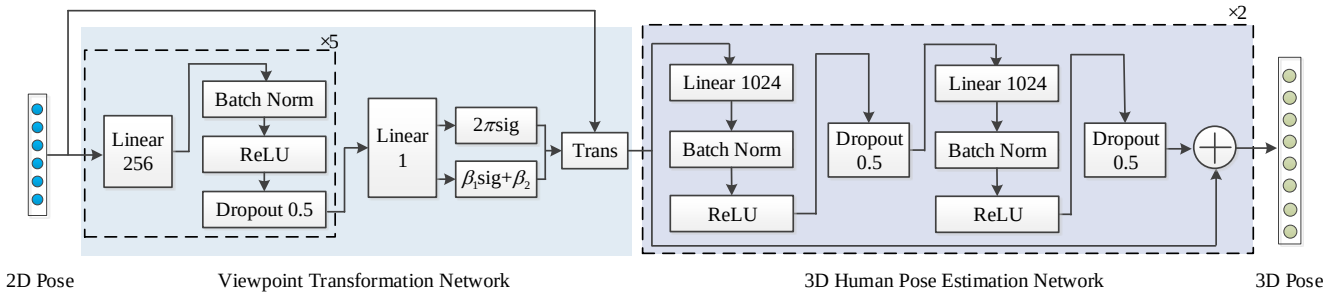

FIGURE 2. The overall proposed network architecture for 3D human pose estimation, which consists of a viewpoint transformation network and a 3D human pose estimation network. The former is to transform a predicted 2D pose to a suitable viewpoint. Then, the transformed 2D pose is delivered to the 3D pose estimation network, which outputs the final 3D pose estimation.

the most famous features. Besides the features, many viewpoint invariant models are also developed. To solve the cross-camera problem, especially the cameras from different angles, [24] employs a Siamese architecture to learn a rotation equivariant representation. Based on the adversarial learning, a viewpoints transformation module is used to reduce the difference in observation coordinate between 2D datasets and 3D datasets [25]. In [26], view point transformation is used to construct a self-supervised learning. [27] shows that combination of 2D joint location and camera view point can improve the performance. Yun et al. [12] analyzes the robustness of 2D human pose estimation with respect to rotational changes. However, their method focuses on the view rotation around the vertical axis.

C. DYNAMIC ADAPTIVE NETWORK

As stated in the section I, dynamic adaptive networks are proposed to deal with the large variation of input. Jia et al. [28] propose the dynamic filter network, which can change the parameters based on the input. In [29], a hypernetwork is used to generate the weight of main network, which is a relaxed form of parameters sharing. Shen et al. [30] employ a meta-network to dynamically produce input-related parameters, which improves the adaptability of the model to different style. All these methods aim to use a network to generate dynamic parameters, which are highly related to the input. Therefore, the overall model can better adapt to the variation of the input, which will lead to higher performance.

Different from the above methods, our dynamic network is used to generate a transformation, which can transform the predicted 2D human pose to a more appropriate viewpoint for 3D pose recovery. The idea has some similarities with [2], which adaptively transformed the 3D human skeleton to the most suitable virtual observation viewpoints for skeleton based action recognition. However, we focus on transforming the 2D pose atomically for 3D human pose estimation, which is highly different from the recognition task.

III. METHOD

As shown in Fig. 2, our method consists of two modules: viewpoint transformation network and 3D human pose estimation network. The former module transforms an original 2D pose to a suitable viewpoint, which depends on the input pose and is learned from datasets. The latter module aims to recover 3D pose from the transformed 2D pose.

A. VIEWPOINT TRANSFORMATION

The viewpoint transformation network is to transform the input 2D pose to a suitable viewpoint for 3D human pose estimation. As we all known, there exist six parameters for a general 2D affine transformation, which can be divided into translation, rotation, scale, skew and aspect ratio. However, through some simple data normalization process, the pelvis location of input 2D poses will be translated into the same position, i.e. the origin of coordinate system. Besides, the skew and aspect ratio transformation will change the relative position of different human joints, which will not conform to the original human poses structure. As a result, we just employ the rotation and scale transformation. In other words, our transformation matrix T_r can be represented as following:

$$T_r = \begin{bmatrix} s \times cos(\alpha) & -s \times sin(\alpha) \\ s \times sin(\alpha) & s \times cos(\alpha) \end{bmatrix} \tag{1}$$

where $s \in R$ is a factor for scale transformation and α is the rotation angle.

As indicted in Eqn. (1), there are two parameters for describing our transformation, i.e the scale factor and rotation angle. Instead of designing some hand-crafted criteria, a deep neural network is employed to generate these two parameters. In theory, the scale factor should be larger than 0. However, our actual experiments show that using too large range for the scale factor will hurt the performance. On the other hand, since the trigonometric function is periodic in rotation angle, we can set the rotation angle in $[0, 2\pi]$ for simplicity. Based on these settings, we employ the sigmoid function to produce the scale factor and rotation angle. In detail, our process for producing the rotation angle and scale factor is defined as following:

$$F = TrNet(P; \theta_t) \tag{2}$$

$$s = \beta_1 \times sigmoid(F) + \beta_2 \tag{3}$$

$$\alpha = 2\pi sigmoid(F) \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013917, IEEE Access

where $P \in R^{2\times J}$ is the 2D human pose predicted from an image, J denotes the number of human joints. $F \in R$ represents the feature after the last linear layer of $TrNet$, which is the network for generating the transformation parameters. And θ_t represents its corresponding parameters to be learned. As shown in Eqn. (3), we use the sigmoid function to generate the scale factor. To increase its range, we magnify the range of scale factor to $[\beta_2,\ \beta_1 + \beta_2]$. The value of β_1 and β_2 are chosen by experiments, which will be detailed in experiment section. We also multiply the sigmoid result by 2π to make the rotation angle in $[0, 2\pi]$.

The detailed architecture of this network is shown in left part of Fig. 2. We can see the network is very simple. First, the combination of linear layer, batch normalization, ReLU and dropout is repeated several times to extract powerful feature from the input pose. Then, a linear layer and two sigmoid functions are use to produce the scale factor and rotation angle respectively.

After obtained the transformation parameters, we can construct the transformation matrix T_r according to Eqn. (1). Then, the transformed human pose can be calculated as following:

$$P_t = T_r \star P \tag{5}$$

where $P_t \in R^{2\times J}$ is the transformed 2D pose and $\star$ denotes the matrix multiplication.

B. 3D HUMAN POSE ESTIMATION

After obtained the transformed 2D pose, the next step is to recover 3D human pose. In this paper, we employ the method proposed in [9] to predict 3D pose due to its high efficiency. Nevertheless, our method is not constricted to this specific model. It is very easy to apply newly proposed 3D human pose estimation method to further improve the performance.

The specific architecture of the 3D pose estimation network [9] is shown in the right part of Fig. 2. Firstly, the transformed 2D pose is projected to a 1024 dimension feature through a linear layer. Then two residual blocks are used to extract powerful features, each of which is the combination of linear layer, batch normalization, ReLU and dropout. Finally, a linear layer is used to produce the 3D human pose. As a result, there are 6 linear layers. Like [9], this figure does not contain the first layer applied to the input pose and the last layer to generate the final 3D pose. Refer to [9] for more details. Although it is not very deep, it has obtained excellent performance in 3D pose estimation.

We train the overall network end-to-end without any pretrain. All the parameter are initialized randomly using the Kaiming normal method. To learn the parameters, we employ the Euclidean distance between the predicted pose and the ground-truth, which can be expressed as following

$$L = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J}\|S_j^n - \hat{S}_j^n\| \tag{6}$$

where N is the number of training samples, S_j^n and $\hat{S}_j^n$ are the predicted 3D location and corresponding ground-truth for j-th joint of n-th training sample respectively. $\|\cdot\|$ denotes the Euclidean loss.

C. DISCUSSION

As stated in the section III-A, we just consider the transformation consisting of scale and rotation. Is this the best transformation? To answer this question, we will conduct experiments by using different kinds of transformation. To facilitate subsequent discussion, we give the representation of a general 2D affine transformation in advance. There exist four parameters for a general transformation without translation. In detail, its transformation matrix can be denoted as following

$$T_r = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{7}$$

where a, b, c and d are the transformation parameters. Compared with Eqn. (1), this equation contains extra skew and aspect ratio transformation. This transformation is more free, but the relative location may be changed through this transformation, which may affect the performance of human pose estimation. We will compare the effect of different transformation in the experimental section.

As shown in [9], it obtains better estimation performance by using camera coordinate frame for both 2D pose and 3D pose. After the transformation, the 2D pose will be in a new coordinate, which maybe lead the 3D pose in this coordinate rather than camera coordinate. Thus, it may hurt the performance if we directly optimize the distance between these two poses in different coordinates. To address this problem, we can transform the predicted 3D pose back to camera coordinate. Actually, this is the inverse transformation of Eqn. (7). After some mathematical operation, the inverse transformation can be represented as following

$$S = T_{inv} * S_t \tag{8}$$

$$T_{inv} = \frac{1}{ad - bc}\begin{bmatrix} d & -b & 0 \\ -c & a & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where $S_t \in R^{3\times J}$ is the output of 3D pose estimation network, S is the final prediction, $T_{inv} \in R^{3\times 3}$ is the matrix for inverse transformation. Since the original transformation is in 2D space, the inverse transformation also acts on 2D space instead of 3D space. In other words, the depth component of joints' location is kept the same. Based on the general formula (9), it is very easy to get the inverse matrix for all kinds of transformation. So we will not give the specific inverse matrix for other kinds of transformation for simplicity.

IV. EXPERIMENTAL RESULTS

This section first introduces the datasets, evaluation metric and some implementation details. Then we analyze the influence of the network architecture, different transformation and some other parameters. Finally, we compare the proposed method with several similar methods and show some sample results.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013917, IEEE Access

A. DATASETS AND EVALUATION METRIC

We mainly focus the numerical evaluation on Human3.6M dataset [17], which is widely used for testing 3D human pose estimation methods. To show the performance of our method on unconstrained environments, we also use the MPII dataset [18].

Human3.6M is currently the largest and widely used dataset for 3D human pose estimation. It is collected in a lab environment, where a person is performing 15 daily activities such as eating, walking, sitting and so on. In total, there are 11 professional subjects, which results in 3.6 million images. For each image, the person bounding box, 2D image coordinate of joints are annotated by human while the 3D positions are gathered by sensors. Refer to [17] for more details. MPII dataset [18] is one of the largest datasets for evaluating 2D human pose estimation methods, which covers a larger range of pose and appearance variation. Since there is no ground-truth 3D pose in the MPII dataset, we just give some qualitative results to demonstrate the performance of our method on the wild images.

For fair comparison with previous works, we follow the standard protocol, which uses subjects 1, 5, 6, 7 and 8 for training, and subjects 9 and 11 for testing. As in [9], we train a single model for all actions. We report the Mean Per Joint Position Error (MJPE) between our prediction and the ground truth across all joints and cameras. There are two different protocols for calculating the MJPE. In protocol $\#1$, the root joint (central hip) is just aligned through 3D translation. In contrast, a rigid transformation is used to align the predicted 3D pose with the ground truth in protocol $\#2$.

B. IMPLEMENTATION DETAILS

We use the Pytorch V1.3.1 to implement the proposed network. The batchsize for training is set to 128. The initial learning rate is set to 1×10^{-4} and decayed every 100,000 iterations according to the exp decay rule of Pytorch. The network parameters are initialized randomly using the Kaiming normal method, which are then optimized by the Adam method. The maximum epoch number is 400. On our machine with an Nvidia 2080Ti GPU, the training of our method takes about 48 hours. Like Martinez et al. [9], we take the 2D pose estimated by a fine-tuned stacked hourglass 2D pose detector as input unless otherwise stated.

C. INFLUENCE OF THE VIEWPOINT TRANSFORMATION NETWORK ARCHITECTURE

We propose to transform a 2D pose to a more suitable viewpoint and then estimate its corresponding 3D human pose from this transformed 2D pose. Hence, the specific configuration of network architecture, which is to generate the viewpoint transformation matrix, plays an important role in improving the performance of 3D human pose estimation. To investigate its influence, we conducted experiments using different network architectures while other things are kept the same. For simplicity, we only use protocol $\#1$ to evaluate the results.

TABLE 1. Influence of the network architecture for generating view transformation on Human3.6M dataset using protocol $\#1$. Conv $M \times N$ refers to the network containing N fully connected layers, each of which contains M neurons. Res means that the residual connection is used. See text for more details

Network	MJPE Error(mm)	# Parameters(M)
Martinez et al. [9]	62.9	4.30
Conv 256×2	60.54	4.37
Conv 256×3	60.59	4.43
Conv 256×4	60.63	4.50
Conv 256×5	**60.38**	4.57
Conv 256×6	60.48	4.63
Conv 256×7	60.82	4.70
Conv 32×5	60.69	4.30
Conv 64×5	60.73	4.31
Conv 128×5	60.70	4.36
Conv 512×5	60.69	5.36
Res 32×2	60.70	4.30
Res 64×2	60.49	4.31
Res 128×2	60.43	4.36
Res 256×2	60.64	4.57
Res 512×2	60.78	5.36
Res 128×1	60.65	4.33
Res 128×3	60.49	4.40

The experimental results using different architectures are listed in Table 1. The first line denoted as "Martinez et al. [9]" is the baseline model without any viewpoint transformation module. The remaining lines give the results of methods using different viewpoint transformation network architecture. Conv $M \times N$ refers to the network composed of N units, each of which contains a combination of a fully connected layer with M neurons, batch normalization, ReLU and a dropout layer. For simplicity, we set the number of neurons in each fully connected layer to be the same. Note that the last linear is not included in the Conv $M \times N$ and is kept the same in all the architectures. We can see that the performance is increasing when the number of fully connected layers is less than 5. When it is bigger than 5, the performance declines. We also compare the performance using different number of neurons in each layers. Specifically, we set it to be 32, 64, 128, 256 and 512. As shown in Table 1, the MJPE error achieves the lowest when it equals 256. Due to the overfitting, the performance degrades using more neuron or more layers.

Recently, the residual connection achieves huge success in many fields. Therefore, we validate whether using residual connection can improve the performance. For simplicity, we employ the same residual units in [9], whose specific architecture is shown in the left part of Fig. 2. The bottom lines of Table 1 gives the influence of the number of residual connection layers and its neurons. Res $M \times N$ refers to the network composed of N residual units, whose fully connected layer contains M neurons. From this table, we can find the MJPE error reaches the minimum 60.43 mm when

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013917, IEEE Access

TABLE 2. Comparison results of our method with other methods on Human3.6M dataset using protocol #1. Numbers are the MJPE error (mm). FT refers that the papers use H3.6M to fine-tune the 2D detector model. GT denotes that the ground-truth 2D locations are used. For all methods, a single model is trained for all actions.

Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD	Walk	WalkT	Avg
Zhou et al. [20]	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
Zhou et al. [31]	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
Pavlakos et al. [21]	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. [9](FT)	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
Véges et al. [24] (FT)	50.1	54.7	56.0	**56.5**	67.7	76.4	53.1	**54.7**	73.3	93.2	60.4	58.5	62.8	51.5	48.2	61.1
Guo et al. [22] (FT)	**49.4**	**54.3**	**55.7**	56.9	66.4	**74.5**	53.2	55.4	71.7	**89.0**	60.0	**57.0**	**62.7**	48.0	**50.7**	60.6
Ours(FT)	49.6	54.6	57.0	57.2	**65.2**	74.7	**53.0**	55.4	**71.3**	**89.0**	**59.7**	57.5	62.8	**47.7**	50.9	**60.4**
Martinez et al. [9](GT)	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
Ours(GT)	**35.3**	**41.9**	**38.4**	**39.5**	**42.9**	**50.6**	**42.9**	**37.5**	**50.3**	**52.8**	**41.6**	**42.1**	**42.2**	**32.4**	**35.8**	**41.8**

TABLE 3. Effects of different kinds of transformation on Human3.6M dataset using protocol #1.

Transformation	Rot	Sca	Rot+Sca	Aff
MJPE Error(mm)	61.47	61.85	**60.38**	61.34
Transformation	Rot+Invs	Sca+Invs	Rot+Sca+Invs	Aff+Invs
MJPE Error(mm)	61.40	61.42	61.25	64.07

TABLE 4. Influence of the scale parameters β_1 and β_2 on Human3.6M dataset using protocol #1.

β_1	1	1.5	1.5	10
β_2	0	0	0.25	0.25
Range	[0, 1]	[0, 1.5]	[0.25, 1.75]	[0.25, 10.25]
MJPE Error(mm)	60.78	60.77	**60.38**	60.63

the number of residual connection and neurons equals 2 and 128 respectively. Other settings will hurt the pose estimation performance. For example, the MJPE error increases to 60.64 mm if 256 neurons are used rather than 128.

Besides, we also give the number of parameters for all models. Compared with the baseline, the total number of parameters dose not increases almost. In terms of normal connection or residual connection, the normal connection obtains a little better performance. In the following experiments, we will utilize the network composed of 5 fully connected layers within 256 neurons.

D. EFFECTS OF DIFFERENT KINDS OF VIEWPOINT TRANSFORMATION

As stated before, there exist several different kinds of transformation, like rotation, scale and affine. So which one is the most suitable for our method? To address this doubt, we have done different experiments using different kinds of transformation. For simplicity, we use the same network architecture, which consists of 5 fully connected layers with 256 neurons. The difference lies in the output of this network. The output are rotation angle, scale factor, rotation angle and scale factor, and four values for rotation, scale, the combination of rotation and scale, and affine respectively. Then, these values are used to construct the transformation matrix, which is used to produce the transformed 2D pose as shown in Eqn. (5)

Table 3 gives the results of different experiments. In this table, the 'Rot', 'Sca', 'Rot+Sca' and 'Aff' denote the rotation, scale, the combination of rotation and scale, and the affine transformation respectively. From the first line of Table 3, we find the combination of rotation and scale gets the best performance. Although the affine transformation is more general, it can not preserve the angle between different limbs, which leads to the performance reduction.

In the second line of Table 3, we give the results of transformation adding an inverse 2D transformation as shown in Eqn. (8). When using rotation or scale individually, the performance improves a little. For the affine or the combination of rotation and scale, the performance declines. In our opinion, the inverse 2D transformation may not be very accurate especially for complex transformation. Due to this result, we will employ the 'Rot+Sca' for comparison in the following experiments.

As stated in Eqn. (3), there are two parameters β_1 and β_2, which is to map the scale factor to a new range instead of $[0, 1]$. Table 4 gives the influence of these two parameters on the 3D pose estimation performance. We can find the performance achieves the best using $1.5, 0.25$ for β_1 and β_2 respectively. Compared the fourth with the fifth column, too larger range for scale factor will result in performance reduction. Therefore, we set the β_1 and β_2 to $1.5, 0.25$, which maps the range of scale factor to $[0.25, 1.75]$.

E. COMPARISONS WITH SIMILAR METHODS

In this section, we compare our method with several other 3D human pose estimation methods under two protocols. Our method is to recover the 3D pose only using an 2D prediction. The appearance or temporal information from images or videos is not used. Therefore, we only use the methods belonging to the same category for fair comparison. In other words, all the methods compared in this section is to recover the 3D pose only from a 2D pose instead of an image. Besides, our method can be further enhanced by other tools, like the generative adversarial network, graph neural network, but this paper focuses on the viewpoint transfor-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013917, IEEE Access

TABLE 5. Comparison results of our method with other methods on Human3.6M dataset using protocol #2. Numbers are the MJPE error (mm). We takes the predictions by a fine-tuned stacked hourglass [11] as input and only train a single model for all actions.

Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD	Walk	WalkT	Avg
Bogo et al. [32]	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 [33]	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3	103.5	74.6	92.6	69.6	71.5	78.0	73.2	74.0
Pavlakos et al. [21]	47.5	50.5	48.3	49.3	50.7	55.2	46.1	48.0	61.1	78.1	51.1	48.3	52.9	41.5	46.4	51.9
Martinez et al. [9]	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
Véges et al. [24]	42.2	44.8	47.5	47.6	54.8	57.8	42.2	40.8	60.5	69.8	50.8	47.4	51.1	44.3	**40.0**	49.4
Guo et al. [22]	**38.8**	**42.1**	**44.4**	**46.0**	**49.8**	**53.4**	**40.5**	**39.3**	**54.5**	**63.5**	**47.6**	**43.2**	**48.6**	**36.6**	41.5	**46.5**
Ours	39.3	43.2	45.6	46.6	50.1	54.0	41.3	40.0	54.8	65.2	48.4	44.5	49.0	37.7	43.1	46.9

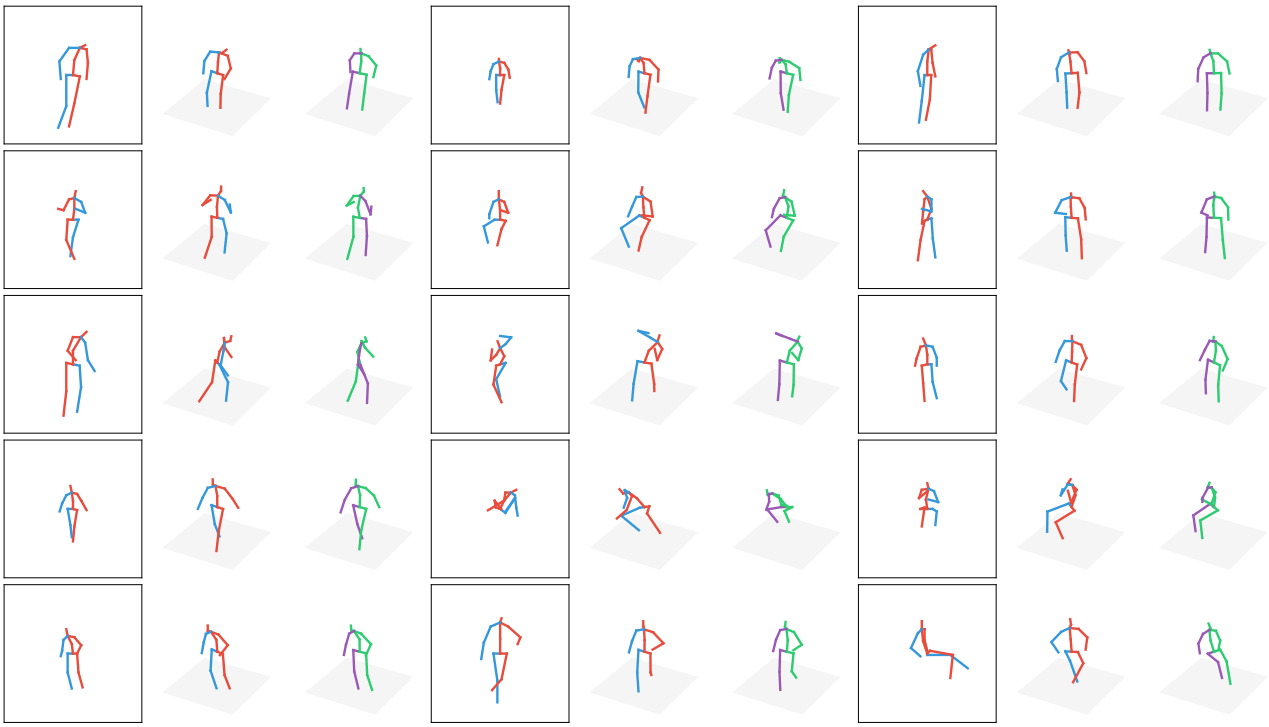

FIGURE 3. Some sample human pose estimation results on the images from the test part of Human3.6M dataset. The 2D pose estimated from a image is in the square. The 3D GT pose in the one in red and blue. Our 3D prediction is the one in green and purple.

mation rather than other tools. So we will not compare our method with these method as well. Therefore, the compared methods include Zhou et al. [20], Zhou et al. [31], Pavlakos et al. [21], Martinez et al. [9], Véges et al. [24], Guo et al. [22], Bogo et al. [32] and Moreno-Noguer [33]. For simplicity, we directly cite the scores reported in the original papers.

The MJPE scores of all methods under protocol #1 and #2 are shown in the Table 2 and 5 respectively. Note that the recent methods [9], [22], [24] use the 2D pose detections by the fine-tuned Stacked Hourglass model [11] as input. To show the limit of our method, we also give the performance using the ground-truth 2D pose in Table 2, which is denoted as GT. We can find our method obtains the best performance under protocol #1. Under protocol #2, the MJPE error is very close to the best method. Compared with our baseline Martinez et al. [9], the average MJPE error reduces 2.5 mm and 0.8 mm under protocol #1 and #2 respectively. The performance gain under protocol #1 is larger than that of protocol #2. Comparing the performance gain between our method and Martinez et al. when using the predicted 2D pose or the ground-truth pose, we can see that the gain is larger when ground-truth 2D poses are used. This indicates our method may obtain better performance if more accurate pose prediction is input.

F. VISUALIZATION RESULTS

Finally, we give some qualitative results on Human3.6M dataset and MPII dataset. First, we show some sample results from Human3.6M dataset in Fig 3. In this figure, the 2D poses estimated from images are in the right. The ground-truth and our predicted 3D human pose are show in the middle and left respectively. Note that these samples are chosen randomly from the test set of Human3.6M dataset. From this figure, we can see our pose estimation results are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013917, IEEE Access

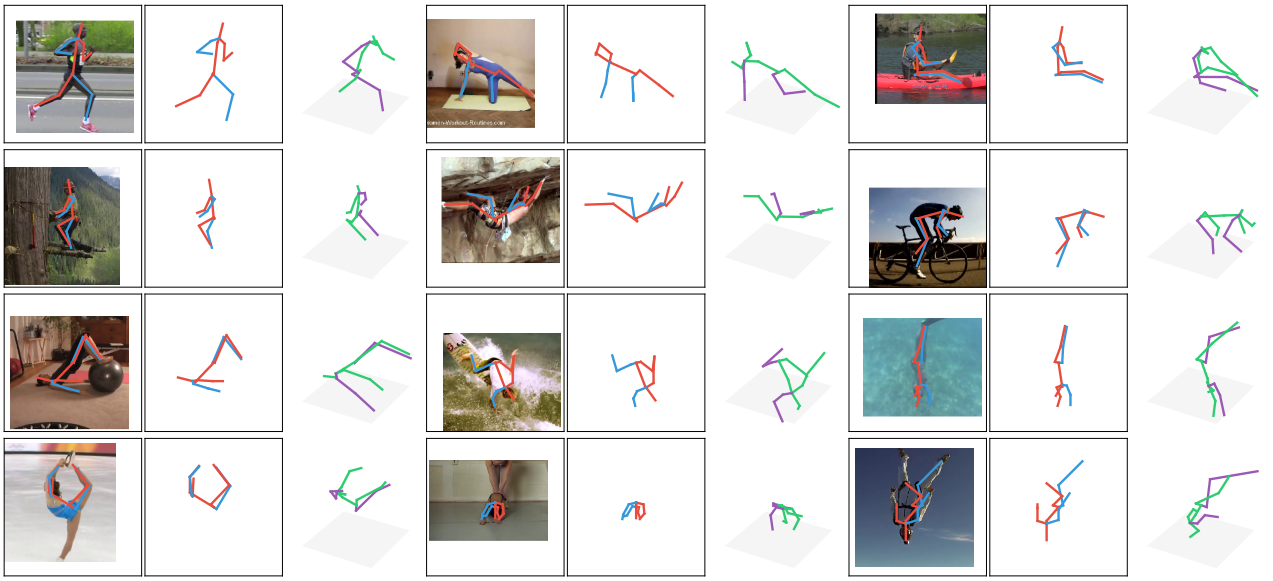

FIGURE 4. Some sample 3D human pose estimation results on the MPII dataset. Left: the input images superposed with 2D pose prediction by Stacked Hourglass model. Middle: 2D pose prediction. Right: our 3D predictions. The top three rows are the right results while the last row shows the failure examples.

highly accorded with the ground-truth 3D pose for all actions. In some samples, our prediction maybe not completely match the ground-truth, especially the angle between upper leg and lower leg, like the third example in the fifth row. This is caused by the inaccurate 2D pose estimation or the viewpoint. For example, in this sample, it is difficult for human to perceive the angle from the 2D pose in this viewpoint. Nevertheless, our prediction is right in most cases. These visualization results verify the effectiveness of our proposed method in lab environment.

To illustrate the performance of our method on images captured in the wild, we have given some sample estimation results on the MPII dataset in Fig. 4. In this figure, the cropped images overlapped with predicted 2D poses by the stacked hourglass model [11] is shown in the right. The 2D pose prediction and our 3D pose estimation results are shown in the middle and left part respectively. Note that the cropped images are just the input of stacked hourglass model. For clear, we use white color to visualize the padding added in the cropping process. Note that we directly use the model trained on Human3.6M instead of training a new model. From Fig. 4, we can find that our method can produce reasonable 3D pose prediction for most cases, although the model dose not see the 2D poses before. For example, our method obtains right prediction for upside-down people which is not similar to any images in Human3.6M dataset. Some failure examples are shown in the last row. This is mainly caused by the false 2D pose prediction. Since our method just takes the 2D pose as input, it can only make prediction based on this input. Therefore, some false results are generated if the 2D pose is not right. Besides, since our method is trained on full body, it will also fail if some joints are missing in the 2D pose prediction. Nevertheless, the predicted 3D poses conform to the input 2D poses.

V. CONCLUSION

In this paper, an adaptive viewpoint transformation network is proposed for 3D human pose estimation. The overall process can be divided into two parts. Given the 2D pose predicted from an image, the first part produces a viewpoint transformation, which is used to transform the 2D pose to a more suitable viewpoint. Next, the 3D human pose is recovered from the transformed 2D pose. In contrast to handcrafted criteria, our viewpoint transformation module is directly learned from the dataset. In inference, it only depends on the input 2D pose. Compared with the original 2D pose, the difficulty of 3D human pose recovery from transformed 2D pose is much smaller. As a result, our method can get better results. Experiments on Human3.6M and MPII datasets show the proposed method can improve the performance of 3D human pose estimation.

This paper mainly focuses on learning a suitable viewpoint transformation from 2D pose prediction, whose information may not be very sufficient. In the future, we will study how to combine the original image and its corresponding 2D pose prediction to learn a more accurate transformation to further improve the performance. Moreover, we will also investigate the application of viewpoint transformation in more fields, such as 2D human pose estimation, action recognition.

REFERENCES

[1] J. Zhong, H. Sun, W. Cao, and Z. He, "Pedestrian motion trajectory prediction with stereo-based 3d deep pose estimation and trajectory learning," IEEE access, vol. 8, pp. 23 480–23 486, 2020.

[2] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human

action recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1963–1978, 2019.
[3] G. Liang, X. Lan, X. Chen, K. Zheng, S. Wang, and N. Zheng, "Cross-view person identification based on confidence-weighted human pose matching," IEEE Trans. Image Process., vol. 28, no. 8, pp. 3821–3835, 2019.
[4] H.-J. Lee and Z. Chen, "Determination of 3d human body postures from a single view," Computer Vision, Graphics, and Image Processing, vol. 30, no. 2, pp. 148–168, 1985.
[5] C. J. Taylor, "Reconstruction of articulated objects from point correspondences in a single uncalibrated image," Computer Vision and Image Understanding, vol. 80, no. 3, pp. 349–363, 2000.
[6] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in Proc. Eur. Conf. Comput. Vis., 2018, pp. 529–545.
[7] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, "Structured prediction of 3d human pose with deep neural networks," in British Machine Vision Conference (BMVC), no. CONF, 2016.
[8] C.-H. Chen and D. Ramanan, "3d human pose estimation = 2d pose estimation + matching," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 7035–7043.
[9] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2640–2649.
[10] H. Ci, C. Wang, X. Ma, and Y. Wang, "Optimizing network structure for 3d human pose estimation," in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 2262–2271.
[11] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in Proc. Eur. Conf. Comput. Vis., 2016, pp. 483–499.
[12] K. Yun, J. Park, and J. Cho, "Robust human pose estimation for rotation via self-supervised learning," IEEE Access, vol. 8, pp. 32 502–32 517, 2020.
[13] R. Wang, C. Huang, and X. Wang, "Relation reasoning graph convolutional networks for human pose estimation," IEEE Access, vol. 8, pp. 38 472–38 480, 2020.
[14] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Fast algorithms for large scale conditional 3d prediction," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. IEEE, 2008, pp. 1–8.
[15] M. Jaderberg, K. Simonyan, A. Zisserman et al., "Spatial transformer networks," in Advances in neural information processing systems, 2015, pp. 2017–2025.
[16] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei, "Towards viewpoint invariant 3d human pose estimation," in Proc. Eur. Conf. Comput. Vis., 2016, pp. 160–177.
[17] 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 Trans. Pattern Anal. Mach. Intell., vol. 36, no. 7, pp. 1325–1339, 2014.
[18] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3686–3693.
[19] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2602–2611.
[20] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in Proc. Eur. Conf. Comput. Vis., 2016, pp. 186–201.
[21] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 7025–7034.
[22] Y. Guo, L. Zhao, S. Zhang, and J. Yang, "Coarse-to-fine 3d human pose estimation," in Image and Graphics, ICIG, 2019, pp. 579–592.
[23] Z. Tang, X. Zhang, and J. Hou, "An articulated structure-aware network for 3d human pose estimation," in Asian Conference on Machine Learning, 2019, pp. 48–63.
[24] M. Véges, V. Varga, and A. Lőrincz, "3d human pose estimation with siamese equivariant embedding," Neurocomputing, vol. 339, pp. 194–201, 2019.
[25] Y. Li, J. Xiao, D. Xie, J. Shao, and J. Wang, "Adversarial learning for viewpoints invariant 3d human pose estimation," Journal of Visual Communication and Image Representation, vol. 58, pp. 374–379, 2019.
[26] Y. Kudo, K. Ogaki, Y. Matsui, and Y. Odagiri, "Unsupervised adversarial learning of 3d human pose from 2d joint locations," arXiv preprint arXiv:1803.08244, 2018.
[27] M. F. Ghezelghieh, R. Kasturi, and S. Sarkar, "Learning camera viewpoint using cnn to improve 3d body pose estimation," in Proc. 2016 fourth international conference on 3D vision (3DV), 2016, pp. 685–693.
[28] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in Advances in Neural Information Processing Systems, 2016, pp. 667–675.
[29] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," arXiv preprint arXiv:1609.09106, 2016.
[30] F. Shen, S. Yan, and G. Zeng, "Neural style transfer via meta networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8061–8069.
[31] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3d human pose estimation from monocular video," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4966–4975.
[32] 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 Proc. Eur. Conf. Comput. Vis., 2016, pp. 561–578.
[33] F. Moreno-Noguer, "3d human pose estimation from a single image via distance matrix regression," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2823–2832.

GUOQIANG LIANG received the B.S. in automation and the Ph.D. degrees in pattern recognition and intelligent systems from Xi′an Jiaotong University (XJTU), Xi′an, China in 2012 and 2018 respectively. From Mar. to Sep. 2017, he was a visiting Ph.D. Student with the University of South Carolina, Columbia, SC, USA. Currently, he is doing the Post-Doctoral Research at the School of Computer Science and Engineering, Northwestern Polytechnical University, Xi′an, China. His research interests include human pose estimation and human action classification.

XIANGPING ZHONG Xiangping Zhong has received the bachelor's degree in engineering from Northwestern Polytechnical University in Jun. 2020. From Sep. 2020 to Jun. 2023, she will pursue a master's degree at the School of Computer Science and Engineering, Northwestern Polytechnical University. Her research interests include human pose estimation and application of artificial intelligence.

LINGYAN RAN received his B.S. degree and Ph.D. degree from Northwestern Polytechnical University(NWPU), Xi′an China, in 2011 and 2018. Earlier, he was a visiting scholar in Stevens Institute of Technology from 2013 to 2015. His research interests include image classification and semantic segmentation. He is currently a member of CSIG.

YANNING ZHANG (SM'10) received the B.S. degree from the Dalian University of Science and Engineering in 1988 and the M.S. and Ph.D. degrees from Northwestern Polytechnical University in 1993 and 1996, respectively. She is currently a Professor with the School of Computer Science, Northwestern Polytechnical University. She has published over 200 papers in international journals, conferences, and Chinese key journals. Her research work focuses on signal and image processing, computer vision, and pattern recognition. She was the Organization Chair of the Ninth Asian Conference on Computer Vision in 2009.