SPGNet: Spatial Projection Guided 3D Human Pose Estimation in Low Dimensional Space

Zihan Wang$^{(\boxtimes)}$, Ruimin Chen , Mengxuan Liu , Guanfang Dong ,
and Anup Basu

University of Alberta, Edmonton, Canada
`{zihan6,ruimin6,mliu2}@ualberta.ca`

Abstract. We propose a method SPGNet for 3D human pose estimation that mixes multi-dimensional re-projection into supervised learning. In this method, the 2D-to-3D-lifting network predicts the global position and coordinates of the 3D human pose. Then, we re-project the estimated 3D pose back to the 2D key points along with spatial adjustments. The loss functions compare the estimated 3D pose with the 3D pose ground truth, and re-projected 2D pose with the input 2D pose. In addition, we propose a kinematic constraint to restrict the predicted target with constant human bone length. Based on the estimation results for the dataset Human3.6M, our approach outperforms many state-of-the-art methods both qualitatively and quantitatively.

Keywords: Machine learning for multimedia · Pattern processing

1 Introduction

3D human shape and posture estimation from a single image or video is a fundamental topic in computer vision. Unfortunately, it is not easy to estimate 3D body shape and posture directly from monocular images without any 3D information. The problem of 3D human shape and posture estimation can be defined as giving images as the input, and generating a 3D skeleton as the output. A typical 3D skeleton consists of 3D points for 17 joints. Mathematically it can be written as a mapping function $f(M) = x$, where M is a fixed-sized matrix representing the image input and x is a matrix representing the 3D joints (size is $(17, 3)$ in our case). Generally, f is applied to each frame of the video.

Before deep learning was widely used, massive 3D labeled data and 3D parameters with prior knowledge were necessary to deal with this problem [1]. After introducing the deep learning strategy, some approaches extract the 3D human pose based on the input image directly without any intermediate stage. Some of the existing strategies rely on convolutional neural networks to learn visual representations successfully from a very large dataset [19,23]. However, recent research has shifted to two-stage approaches. The 2D key points detection algorithm is first used to acquire the 2D poses from images. Then, 2D-to-3D pose

lifting is applied as the second stage [3,15,18]. Existing methods have focused on optimizing the loss functions or neural network structures [3,15,18,20].

Inspired by the cycle consistency in unsupervised learning, some approaches re-map the predicted 3D poses to 2D poses in a semi-supervised learning framework [20]. Like following the CycleGAN Loss, the network is designed incorporating two components. One is the mapping from 2D to 3D. Another one is re-mapping 3D to 2D and comparing the re-projected 2D poses with the 2D input. However, the results for unlabeled videos are relatively unremarkable and the training process is relatively slow. In our work, we also extract the intermediate 2D pose. However, we do not intuitively deploy the CycleGAN semi-supervised learning framework, which leads us to avoid the abovementioned drawbacks.
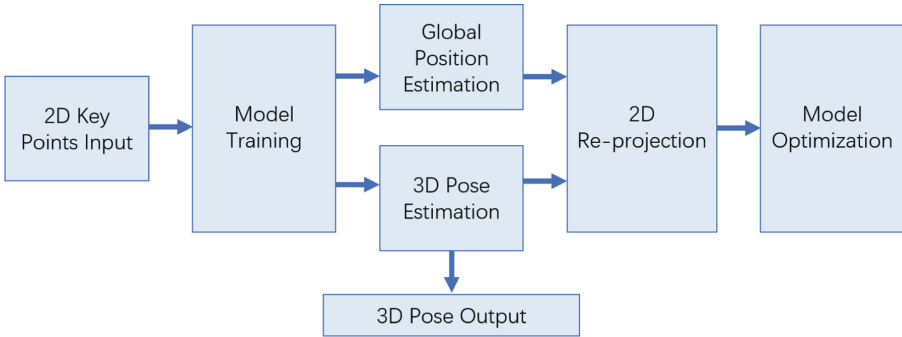

Fig. 1. General workflow of our SPGNet, take 2D key points as input, and estimate the 3D pose with its global position. The 2D re-projection combines both estimations and loss to optimize the model.

We design SPGNet, a novel neural network architecture that combines traditional supervised learning with a spatial projection that re-projects the predicted 3D poses to 2D poses. Instead of traditional loss functions based on the ground truth and 3D pose output, we introduce the 2D MPJPE loss (defined as Eq. 3). As shown in Fig. 1, SPGNet computes the 2D projection of 3D poses output with the estimated global position. The 2D MPJPE loss minimizes the re-projected 2D poses with the 2D poses input, which reuses the 2D poses input during the learning process. This approach increases the accuracy of 3D pose prediction. Overall, our main contributions are:

1. Introducing an adaptive supervised training framework for 3D human pose estimation under the category of 2D-to-3D lifting approach.
2. Exploiting the 2D pose input efficiently and improving robustness by presenting a re-projection loss, which is based on global pose estimation from 2D poses and the 2D poses themself.

Our model achieves 45.3 mm accuracy in Protocol 1 and 35.7 mm in Protocol 2 in the Human3.6M dataset, which are 0.6% and 1.4% relative improvements over previous approaches [28].

2   Related Work

Previous research on 3D Human Pose Estimation can be classified into two main categories. The first category extracts the 3D human pose based on the input image directly without any intermediate stage. Under this category, some recent approaches rely on convolutional neural networks to learn visual representations successfully from very large datasets. The predicted accuracy has increased considerably during the past few years [19,23]. The second category consists of two stages, namely 2D key points detection from video and 2D-to-3D pose lifting [3,15,18].

2.1   2D Key Joint Detection

Before estimating 3D human pose, many methods require labeled key joints. Inaccurate key joint labels may cause pose prediction to fail. Initially, 2D human pose was estimated based on Deep Neural Networks (DNNs), later Convolutional Neural Networks (CNNs) have shown more advantages in 2D key joint detection [16,24,25,27,29]. CNN-based models have outstanding performances in extreme test conditions (body occlusion or low-resolution images). A heatmap is generated to show the possibilities of a specific joint shown in an image and the estimate is refined to improve localization. Reusing the hidden layer in the CNN-based model, Tompson et al. proposed a method that uses heatmap, works as a regularizer to find their output, and increases the accuracy [24]. A similar method proposed by Yang et al. uses an end-to-end mixture of parts model [29]. In their method, the probability is calculated by the softmax function and a Max-sum algorithm is applied between pose parts. The pose machine uses multi-stage differentiable iterations to the joint on the heat-map to finally converge to one solution [16,27]. The pose machine combines the previous output and the updated prediction for the same input image in each subsequent stage.

2.2   Image to 3D Pose

Without estimating the 2D human pose, the 3D pose can be constructed directly through an image. This method minimizes the effects of error prediction during the 2D pose estimation, and in general, it can be more robust [9,21,26]. Additionally, it eliminates the limitation of the unlabeled image used for training [30]. One of the methods uses Pose Orientation Net (PONet) and generates heat maps: limb confidence maps and 3D orientation maps. With these heat maps, the model uses a fixed-length skeleton to match with the 3D orientation maps and complete the missing limbs in a 3D pose using sub-networks [26]. For a more specific pose, Ruiz et al. used three loss functions for each angle's rotation, with classification and regression [21]. Another approach proposed by Yang et al. uses the Generative Adversarial Networks (GANs) to directly predict the 3D pose using unlabeled images in outdoor scenarios [30]. Therefore, skipping 2D pose detection provides a solution for unpaired 2D-to-3D data training [9,12].

2.3   2D-to-3D Pose Lifting

3D pose estimation can be lifted from 2D human pose joints [3,15,18]. During the training, some approaches directly using 2D human pose ground truth as input [5,11]. This approach ensures the accuracy of the inputs and distraction is minimized. Moreover, 2D label on the image is easier to obtain. Some datasets, like Human3.6M and HumanEval, contain multiple 2D views, and EpipolarPose with another method estimate 3D pose from different directions at the same frame. The mixture of the multi-views provides consistency loss and recovers the pose via triangulation [11]. The model can be optimized by blending multiple 2D views into the same 3D pose. Depth prediction is important for some 2D-to-3D approaches. The simpler method uses binary ordinal depth relation prediction, while other have explicit depth prediction on each joint [3,17,18].
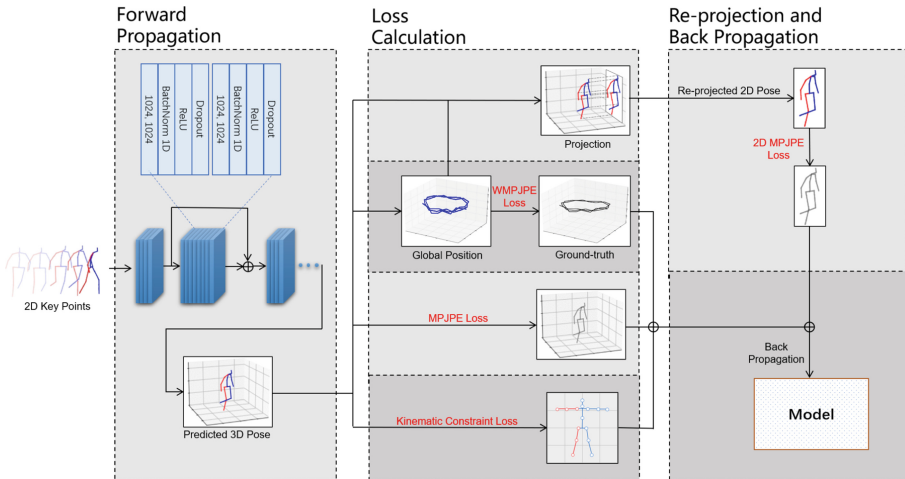

Fig. 2. Supervised training framework with a 3D pose model with predicted 2D pose sequence input. In addition to the kinematic constraint and MPJPE loss, we concatenate the global position and key points of the 3D poses to perform the projection. Finally, we compare the re-projected 2D pose and original input 2D poses to compute a 2D MPJPE loss and perform backpropagation.

3   Proposed Method

In this section, we briefly explain the details of the proposed SPGNet architecture, as shown in Fig. 2.

3.1   Problem Formulation

To better illustrate our method, we first formulate the 3D pose estimation problem as a 2D-to-3D lifting pipeline. Then, we assume the dataset defined as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ consisting of N data, where each data x_i is associated with a corresponding label y_i. Here, $x_i \in \mathbb{R}^{M \times 2}$ represents the one frame input of 2D pose key points. Similarly, $y_i \in \mathbb{R}^{M \times 3}$ represent estimated 3D pose key points labels, where M denoted as number of joints for each human pose. In order to take advantage of temporal information between frames, we define the sequence of input for one frame as vector $\{\mathbf{x}^k \in \mathbb{R}^{1 \times M \times 2} | k = 1, 2, 3...J\}$, where J represents the number of frames in the input sequence. The goal is to optimize the prediction function (estimated 3D pose key points in three dimensions) f from the training data,

$$f(\mathbf{x}; W) = \arg\max_{y \in \mathcal{D}} F(\mathbf{x}, y; W) \tag{1}$$

where W is the weight of the neural network and F is the optimizing function.

3.2   Spatial Projection Guided Approach

In the training process, SPGNet contains three components: an encoder, a projector and loss functions. First, the encoder processes the input 2D key points, aiming to transform the data into precise 3D coordinates that represent the estimated human pose. Then, the projector transforms the estimated 3D human pose into a re-projected 2D pose for computing the similarity with the original input. Finally, multiple loss functions guide the backpropagation of the neural network in order to learn helpful representations in the latent space.

The Encoder. First, we encode our input data through an encoder and obtain the predicted 3D pose key points in three dimensions. Then, we decompose our output $\hat{t}_i$ for a certain frame into $\hat{t}_i^k$ and $\hat{t}_i^p$, where $\hat{t}_i^k$ is the predicted 3D pose key points coordinates in a certain 2D plane and $\hat{t}_i^p$ is the spatial information to confirm the global position of pose. In this case, $\hat{t}_i^k \in \mathbb{R}^{M \times 2}$, which is the (x, y) coordinates of each joint of a 3D pose in three dimensions. Similarly, $\hat{t}_i^p \in \mathbb{R}^{M}$ is just the (z) coordinate as predicted by the position. We do same decomposition for label y_i and gain y_i^k, y_i^p for further loss computation.

The Projector. A projector is utilized to map the estimated 3D human pose into the re-projected 2D pose. In order to minimize the impact of lens distortion on projection, we chose the nonlinear projector. The schematic diagram is shown in Fig. 3. This schematic diagram shows the forward projection onto the image plane that maps (x, y, z) coordinate into the (x, y) plane. The center is the camera center, and the focal length is the distance between the camera center and the image plane along the principal axis perpendicular to the image plane. These two parameters represent the spatial geometry relationship of the 3D pose in low dimensions. Another essential aspect to consider is the camera
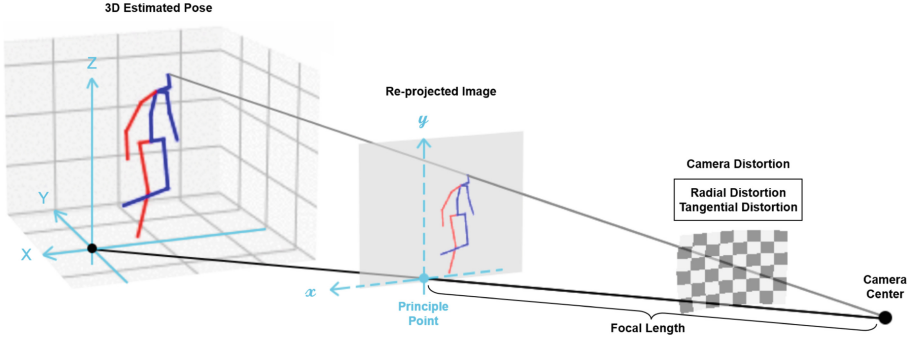
Fig. 3. Schematic diagram of the spatial dimension projection for SPGNet. The projector simulates the real camera to maintain consistency. The 3D estimated pose is projected to a 2D image with the same principle point, focal length, radial distortion and tangential distortion.

distortion. It is a kind of optical aberration that causes a straight line in the scene to not remain straight in an image. There are two common camera distortions. First, the radial distortion causes the magnification of the image to decrease or increase with distance from the principal axis. Second, tangential distortion occurs because the lens assembly is not centered and parallel to the image plane. Fixing lens distortion during projection can increase projection accuracy. Consequently, the projector has several intrinsic camera parameters as input to ensure the projection is at a right angle, specified by the particular dataset. Here, we define focal length, center, radial distortion, and tangential distortion as f_c, c_e, d_r, and d_t, respectively. The pseudocode of nonlinear projection is summarized in Algorithm 1.

Algorithm 1: Nonlinear projector mapping estimated 3D pose to re-projected 2D pose.

---

1 <u>Projector</u> $(\hat{t}_i^k, \hat{t}_i^p, f_c, c_e, d_r, d_t)$;

    Input : Estimated 3D pose $\hat{t}_i^k$, estimated global position $\hat{t}_i^p$, intrinsic camera parameters (f_c, c_e, d_r, d_t)

    Output: Re-projected 2D pose $pose_{2d}$

2 $pose_{depth} = \hat{t}_i^k / \hat{t}_i^p$;

3 $pose_{cons} = $ Clamp $(pose_{depth})$ with $(min = -1, max = 1)$;

4 $r = $ Sum $(pose_{cons})$ value between (x, y) coordinates;

5 $concat = $ Concatenates the given sequence (r, r^2, r^3) ;

6 $R = $ Sum $(1 + (d_r \cdot concat))$ value between (x, y, z) coordinates;

7 $T = $ Sum $(d_t \cdot pose_{cons})$ value between (x, y) coordinates;

8 $pose_{trans} = pose_{cons} \cdot (R + T) + d_t \cdot r$;

9 $pose_{2d} = f_c \cdot pose_{trans} + c_e$;

10 **return** $pose_{2d}$

Loss Functions. Usually, deep learning models predict the target without restraint. Thus, we propose the kinematic constraint loss as a penalty to maintain consistency. This is under the assumption that the length of human bones is constant from beginning to end. We construct the output 3D pose pairs $(\hat{\alpha}, \hat{\beta})$, where $\hat{\alpha}$ is the previous predicted frame $\hat{t}$, and $\hat{\beta}$ is current predicted frame. Then, we define kinematic constraint loss as follows:

$$\mathcal{L}_{kc}(\hat{\alpha}; \hat{\beta}) = \frac{1}{2M} \sum_{i=1}^{M} \sum_{j=1}^{P} abs(\|\hat{\alpha}_i - \hat{\alpha}_j\|_2 - \|\hat{\beta}_i - \hat{\beta}_j\|_2) \tag{2}$$

where, $\hat{\alpha}_j$ is the parent joint of the current joint $\hat{\alpha}_i$. The 3D human pose can be regarded as a tree-like structure. Each joint has at least one connection with the other joint (parent), and some of them have multiple connections. Thus, for a particular joint, the number of connections is determined by the P denoted as parents. The dummy variable P depends on a specific dataset that consists of an indeterminate number of joints. The constant 0.5 is a multiplier for model stability because the repeated bone length would be calculated twice. Note that the first predicted frame does not have $\hat{\alpha}$. Thus, we omit the computation of the loss function for the first predicted frame. This loss can be regarded as a penalty between consecutive frames and effectively maintaining the length of the bone, as shown in Sect. 4.5.

The loss function for estimated 3D human pose is simply the mean per joint position error (MPJPE), which is the Protocol 1 used in many existing work:

$$\mathcal{L}_{mpjpe}(\hat{t}_i^k; y^k) = \frac{1}{M} \sum_{i=1}^{M} \|\hat{t}_i^k - y_i^k\|_2 \tag{3}$$

The MPJPE loss will calculate the euclidean distance between all the joints of the predicted 3D pose and the ground truth. During the backpropagation, the loss gradient provides information for optimizing the degree of the key point's accuracy. Notably, the ground truth is in the camera space, transformed by using the intrinsic and extrinsic camera parameters. Therefore, for the global position y^p, MPJPE cannot hold the depth information of the 3D pose. We use the weighted MPJPE loss function for estimated global position to retain the maximum spatial feature:

$$\mathcal{L}_{wmpjpe}(\hat{t}_i^p; y^p) = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{y_i^k} \|\hat{t}_i^p - y_i^k\|_2 \tag{4}$$

The inverse term $1/y_i^k$ is the regularization term compared with MPJPE loss to force the predicted 3D pose centered around the trim area. This is assuming that the pose object cannot move very far away from the camera position. The model learns the 3D pose characteristics of centralization in terms of results. In addition, the weighted MPJPE loss significantly increases the accuracy of the projected 2D pose and reduces the error caused by the abnormal predicted global position.

3.3   Design of Encoder

The architecture of the 2D-to-3D lifting neural network we designed is a temporal dilated convolutional model inspired by previous lifting approaches [15,20]. The neural network is fully implemented with the residual connections [6] in order to transform the sequence of input $\mathbf{x}$ defined in Sect. 3.1 through temporal convolutional layers. In detail, each residual blocks can be defined as

$$\mathbf{z} = D(\sigma(N(C(\mathbf{x})))), \tag{5}$$

where $\mathbf{z}$ is the extracted feature, C is the convolutional layer with 1024 input sizes except the input layer and 1024 output sizes, B is the batch normalization layer, σ is the ReLU activation layer, and finally, D is the dropout layer. Two residual blocks form a residual connection shown in Fig. 2. The number of residual blocks depends on the number of input frames. For example, input $\mathbf{x}$ with $J = 243$ needs 8 residual blocks for residual connections. Reminder, the input convolutional layer has $2 \cdot M$ input size to adapt the 2D key points, where M is defined in Sect. 3.1. Finally, the output layer is a convolutional layer with output size $3 \cdot M$, fitting with size of estimated 3D pose, defined as:

$$\hat{t}_i = C_{out}(\mathbf{z}), \tag{6}$$

where $\hat{t}_i$ is the output of encoder for one frame defined in Sect. 3.2.

Table 1. Detailed results under Protocol 1 (MPJPE)

Methods	Martinez [15]	Sun [22]	Yang [30]	Lee [13]	Pavllo [20]	Cai [2]	Xu [28]	Our
Direct.	51.8	52.8	51.5	40.2	45.1	44.6	**37.4**	37.5
Discuss.	56.2	54.8	58.9	49.2	47.4	47.4	**43.5**	44.7
Eat	58.1	54.2	50.4	47.8	42.0	45.6	42.7	**41.8**
Greet	59.0	54.3	57.0	52.6	46.0	48.8	42.7	**42.1**
Phone	69.5	61.8	62.1	50.1	49.1	50.8	46.6	**45.5**
Photo	78.4	67.2	65.4	75.0	56.7	59.0	59.7	**58.9**
Pose	55.2	53.1	49.8	50.2	44.5	47.2	**41.3**	42.0
Purch	58.1	53.6	52.7	43.0	44.4	**43.9**	45.1	46.7
Sit	74.0	71.7	69.2	55.8	57.2	57.9	**52.7**	52.8
SitD	94.6	86.7	85.2	73.9	66.1	61.9	60.2	**59.4**
Smoke	62.3	61.5	57.4	54.1	47.5	49.7	**45.8**	46.7
Wait	59.1	53.4	58.4	55.6	44.8	46.6	43.1	**42.8**
WalkD	65.1	61.6	**43.6**	58.2	49.2	51.3	47.7	46.6
Walk	49.5	47.1	60.1	43.3	32.6	37.1	**33.7**	34.7
WalkT	52.4	53.4	47.7	43.3	34.0	39.4	37.1	**36.8**
Avg	62.9	59.1	58.6	52.8	47.1	48.8	45.6	**45.3**

*The table reports the result with CPN 2D detection pose key points as input. The last line is the average of all 15 action results in millimeter. Best results in bold, second best underlined.

4 Experiments

4.1 Dataset and Metrics

We evaluate our method on the public dataset **Human3.6M**, widely used in other work [8]. The Human3.6M dataset is collected by a motion capture system. As one of the largest 3D human pose estimation datasets, 11 professional actors performed 15 scenarios consisting of 3.6 million video frames. There are four digital video cameras, one time-of-flight sensor, and ten motion cameras to capture the human pose. Our experiments follow the previous work [20] to adopt a standard 17-joints skeleton and split the dataset into a training set (S1, S5, S6, S7, S8) and a testing set (S9, S11).

We take two widely used protocols to evaluate our model: **Protocol 1** is the mean per joint position error (MPJPE) defined in Eq. 3. MPJPE calculates the Euclidean distance between the estimated positions of the joints and ground truth in millimeters. **Protocol 2** is the Procrustes mean per joint position error (P-MPJPE), which calculates the error after aligning the estimated 3D pose to the ground truth in a rigid transformation, such as translation, rotation, and scale.

4.2 Implementation Details

We train our model with input in camera space for consistency of other work through quaternion transformation. Here, we set our temporal convolutional model with 243 frames to benefit from the consecutive video stream. We choose Adam [10] as an optimizer and train our model with 100 epochs. The learning rate starts from 0.001 and decays exponentially every epoch. We adopt fine-tuned 2D pose detection key points through the Cascaded Pyramid Network [4] and ground truth 2D as our input. We apply the data augmentation method, pose flipping horizontally in the training set, with settings similar to [20].

4.3 Comparison with State-of-the-art Methods

This section reports our model's performance on 15 actions belonging to S9 and S11. First, we use the CPN network as the 2D pose detector to obtain the 2D key points as our input data. We compare them using Protocol 1 and Protocol 2, shown in Tables 1 and 2. Our model has a lower average error than all other approaches under both protocols and does not rely on additional data as many other approaches. Under Protocol 1 (Table 1), our model slightly outperforms the previous best result [28] by 0.3 mm on the average, corresponding to a 0.6% error reduction. To be more specific, we got the best 7 out of 15 actions and 7 s best actions. This indicates that our model's architecture has a better learning ability to extract features in the latent space in order to keep the spatial and temporal information. For Protocol 2, our model achieves the best results in terms of average P-MJPJE, with a 1.4% error reduction. However, for individual actions, most actions predicted by our model only achieve the second-best results.

Table 2. Detailed results under Protocol 2 (P-MPJPE)

Methods	Martinez [15]	Sun [22]	Yang [30]	Lee [13]	Pavllo [20]	Liu [14]	Xu [28]	Our
Direct.	39.5	42.1	**26.9**	34.9	34.2	32.5	31.0	<u>30.8</u>
Discuss.	43.2	44.3	**30.9**	35.2	36.8	35.3	36.8	<u>34.6</u>
Eat	46.4	45.0	36.3	43.2	**33.9**	34.3	34.7	<u>34.1</u>
Greet	47.0	45.4	39.9	42.6	37.5	36.2	**34.4**	<u>35.8</u>
Phone	51.0	51.5	43.9	46.2	37.1	37.8	<u>36.2</u>	**35.3**
Photo	56.0	53.0	47.4	55.0	43.2	**43.0**	43.9	<u>43.2</u>
Pose	41.4	43.2	**28.8**	37.6	34.4	33.0	<u>31.6</u>	31.9
Purch	40.6	41.3	**29.4**	38.8	33.5	<u>32.2</u>	33.5	32.5
Sit	56.5	59.3	**36.9**	50.9	45.3	45.7	42.3	<u>42.1</u>
SitD	69.4	73.3	58.4	67.3	52.7	51.8	**49.0**	<u>49.9</u>
Smoke	49.2	51.0	41.5	48.9	<u>37.7</u>	38.4	**37.1**	39.0
Wait	45.0	44.0	**30.5**	35.2	34.1	32.8	33.0	<u>32.6</u>
WalkD	49.5	48.0	**29.5**	50.7	38.0	37.5	39.1	<u>37.2</u>
Walk	38.0	38.3	42.5	31.0	**25.8**	**25.8**	26.9	<u>26.9</u>
WalkT	43.1	44.8	32.2	34.6	**27.7**	<u>28.9</u>	31.9	29.7
Avg	47.7	48.3	37.7	43.4	36.8	36.8	<u>36.2</u>	**35.7**

*The table reports the result with CPN 2D detection pose key points as input. The last line is the average of all 15 action results in millimeter. Best in bold, second best underlined.

Yang's [30] work reports a significant improvement in the actions with complex spatial relationships, such as sitting or direction, achieving the seven best results. Compared to results in Table 1, their method uses the feature of GANs but makes it hard to detect the global position of the 3D human pose, leading the better performance in Protocol 1.

To further study our method, we utilize the ground truth of 2D key points as our input to evaluate our model, with results shown in Table 3 under Protocol 1 and Protocol 2. By using 2D ground truth, the models generally get better performance than Table 2. Compared to the previous best result [14], our model outperforms by 1.3 mm on the average. Our method highlights the reuse of the 2D key points for the 2D MPJPE loss, so the model depends more on the 2D key points than other models that only use those data at the input stage. This is evident from the higher error deduction comparing our model to previously implemented methods between Tables 2 and 3. More discussion about the improvement in performance in this case is presented in a later section. We also report our model's performance in Protocol 2, resulting in better performance of 25.3 mm on the average, outperforming the state-of-the-art.

4.4   Ablation Study

Analysis of Re-Projected 2D Pose Loss. We further analyze the MPJPE loss between the re-projected 2D pose and the original input. We choose the encoder only consisting of MPJPE loss for the estimated 3D human pose as our

Table 3. Detailed results based on ground truth of 2D human pose.

Methods	Protocol 1				Protocol 2
	Hossain [7]	Lee [13]	Liu [14]	Our	Our
Direct.	35.2	32.1	34.5	**29.5**	21.1
Discuss.	40.8	36.6	37.1	**33.6**	23.9
Eat	37.2	34.3	33.6	**32.8**	24.5
Greet	37.4	37.8	34.2	**32.5**	24.6
Phone	43.2	44.5	32.9	**32.5**	25.3
Photo	44.0	49.9	37.1	**33.6**	29.9
Pose	38.9	40.9	39.6	**37.5**	21.1
Purch.	35.6	36.2	35.8	**29.8**	23.7
Sit	42.3	44.1	40.7	**30.7**	30.4
SitD	44.6	45.6	41.4	**38.1**	38.9
Smoke	39.7	35.3	**33.0**	46.4	26.9
Wait	39.7	35.9	**33.8**	35.9	23.1
WalkD	40.2	30.3	33.0	**31.8**	27.7
Walk	32.8	37.6	26.6	**25.5**	18.8
WalkT	35.5	35.5	**26.9**	28.5	19.6
Avg	39.2	38.4	34.7	**33.4**	25.3

*Utilize the ground truth 2D human pose to predict the target. The table reports results of both Protocal 1 (left side) and Protocol 2 (right side). We label the best in bold.

encoder and add kinematic constraint as Baseline* to compare with SPGNet, as shown in Fig. 4a. The histogram indicates that our model dramatically benefits from the MPJPE loss of the re-projected 2D pose, leading to the re-projected 2D pose being closer to the ground truth 2D human pose. Furthermore, the kinematic constraint slightly improves the model's accuracy, by adding a bone length constraint.

Analysis of Size of Input Frames. The size of the input frames has a significant impact on the pose estimation performance. For example, the chart in Fig. 4b shows that the error in Protocol 1 decreases by 0.9 mm and 1.8 mm while the size of the input frames increases from 27 to 81 and 243, respectively. A similar error decrease trend is also reflected in Protocol 2. Consequently, we conclude that the larger size of sequential input provides more temporal information to allow our model to capture the movement of the 3D human pose between frames, similar to [20].
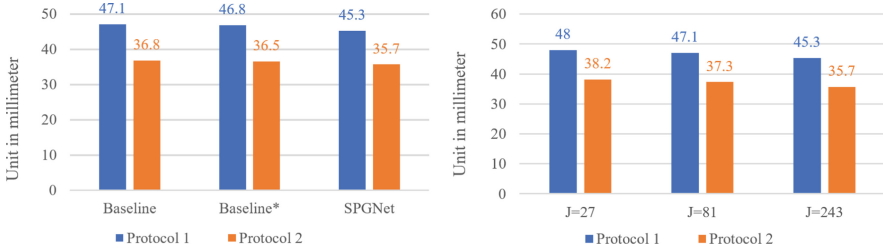
(a) Comparisons of different settings   (b) Comparisons of size of input frames

Fig. 4. Clustered column charts (a) and (b) illustrate the comparisons of different sittings and input frames, respectively. The error shows a decreasing trend as the model is applied from the Baseline to our SPGNet or the size of input frames increases. Notably, in Fig. 4a, Baseline is the model only learned from MPJPE loss, and Baseline* represents the Baseline adding kinematic constraints. In the chart 4b, J represents the number of frames in the input sequence. (a) Comparisons of different settings. (b) Comparisons of size of input frames.

4.5   Effect of Kinematic Constraints

In this section, we compare the sixteen bone lengths measured in different frames in one particular walking scenario, shown in Fig. 5. We can see that the difference in most small bones is almost negligible. Some large bones, such as leg bone or spine, have some relatively large variance compared to the small bone. However, the errors are all within $0.065\,\mathrm{m}$, which is acceptable and may be caused by movement of the frame. As a consequence, this line chart indicates that our kinematic constraint is effective.
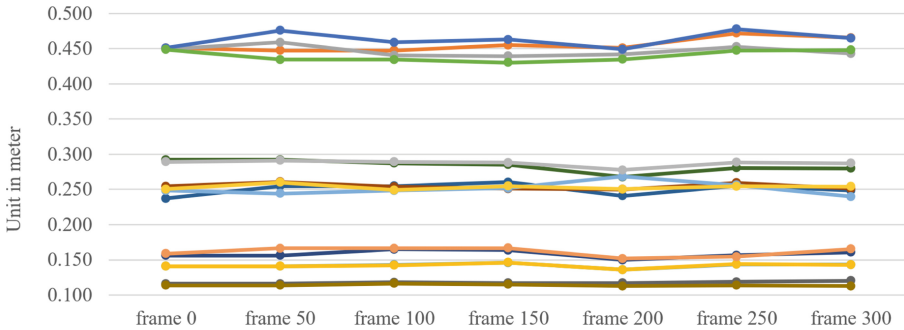

Fig. 5. Line chart of the measured bone lengths in seven different frames extracted from S11 walking scenarios. The different colors of lines represent the sixteen bones of a 3D human pose (tree-like structure).

4.6   Shielding Problem

We further analyze the limitation of our method from visualization results, shown in Fig. 6. We found that in the first row, which represents the action of sitting, the human limbs barely overlap in the camera's view. Thus, our results cannot be distinguished by the human eye if the consideration of global position is ignored. However, in the third row, the red arm of the human overlaps with the human body, which forms a shield masking the inner limbs. Consequently, the spatial relationship is not perfectly represented for both human pose in the third row, the variance of body tilt, and the difference of the arm's spatial positions, respectively.
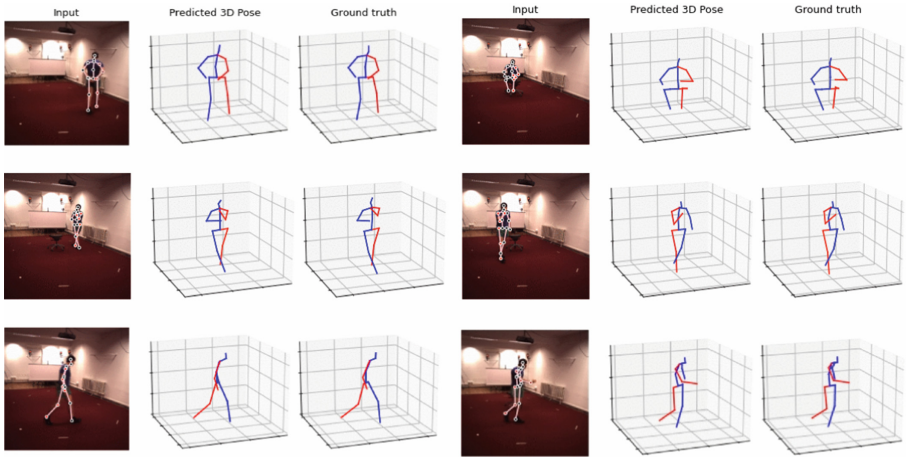

Fig. 6. Visualization of qualitative results on video clips.

5   Conclusion

We proposed SPGNet, a fully convolutional network based on supervised learning for human 3D pose estimation. To utilize our 2D-to-3d-lifting network, we used 2D key points in both input and re-projection stages and introduced kinematic constraints of human bone length and the corresponding loss function. Our model achieved more reliable estimates than state-of-the-art methods. SPGNet utilizes 2D labels in a more effective way, so the performance is expected to increase using image-to-2D methods with higher accuracies. Furthermore, besides the popular Human3.6M dataset, more datasets need to be tested for better analysis of our method.

References

1. Andriluka, M., Roth, S., Schiele, B.: Monocular 3D pose estimation and tracking by detection. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 623–630 (2010). https://doi.org/10.1109/CVPR.2010.5540156

2. Cai, Y., et al.: Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2272–2281 (2019). https://doi.org/10.1109/ICCV.2019.00236

3. Chen, C., Ramanan, D.: 3D human pose estimation = 2D pose estimation + matching. CoRR abs/1612.06524 (2016). https://arxiv.org/abs/1612.06524

4. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. CoRR abs/1711.07319 (2017). https://arxiv.org/abs/1711.07319

5. Drover, D., Rohith, M.V., Chen, C., Agrawal, A., Tyagi, A., Huynh, C.P.: Can 3D pose be learned from 2D projections alone? CoRR abs/1808.07182 (2018). https://arxiv.org/abs/1808.07182

6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). https://arxiv.org/abs/1512.03385

7. Hossain, M.R.I., Little, J.J.: Exploiting temporal information for 3D pose estimation. CoRR abs/1711.08585 (2017). https://arxiv.org/abs/1711.08585

8. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)

9. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. CoRR abs/1712.06584 (2017). https://arxiv.org/abs/1712.06584

10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014). https://dblp.uni-trier.de/db/journals/corr/corr1412.html

11. Kocabas, M., Karagoz, S., Akbas, E.: Self-supervised learning of 3D human pose using multi-view geometry. CoRR abs/1903.02330 (2019). https://arxiv.org/abs/1903.02330

12. Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M.J., Gehler, P.V.: Unite the people: closing the loop between 3D and 2D human representations. CoRR abs/1701.02468 (2017). https://arxiv.org/abs/1701.02468

13. Lee, K., Lee, I., Lee, S.: Propagating LSTM: 3D pose estimation based on joint interdependency. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 123–141. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_8

14. Liu, R., Shen, J., Wang, H., Chen, C., Cheung, S.C., Asari, V.: Attention mechanism exploits temporal contexts: real-time 3D human pose reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5064–5073 (2020)

15. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3D human pose estimation. CoRR abs/1705.03098 (2017). https://arxiv.org/abs/1705.03098

16. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. CoRR abs/1603.06937 (2016). https://arxiv.org/abs/1603.06937

17. Nie, X., Wei, P., Zhu, S.C.: Monocular 3D human pose estimation by predicting depth on joints. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 3467–3475 (2017)

18. Pavlakos, G., Zhou, X., Daniilidis, K.: Ordinal depth supervision for 3D human pose estimation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7307–7316 (2018). https://doi.org/10.1109/CVPR.2018.00763
19. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3D human pose. CoRR abs/1611.07828 (2016). https://arxiv.org/abs/1611.07828
20. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3D human pose estimation in video with temporal convolutions and semi-supervised training. CoRR abs/1811.11742 (2018). https://arxiv.org/abs/1811.11742
21. Ruiz, N., Chong, E., Rehg, J.M.: Fine-grained head pose estimation without keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 2074–2083 (2018)
22. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. CoRR abs/1704.00159 (2017). https://arxiv.org/abs/1704.00159
23. Sun, X., Xiao, B., Liang, S., Wei, Y.: Integral human pose regression. CoRR abs/1711.08229 (2017). https://arxiv.org/abs/1711.08229
24. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 648–656 (2015)
25. Toshev, A., Szegedy, C.: DeepPose: human pose estimation via deep neural networks. CoRR abs/1312.4659 (2013). https://arxiv.org/abs/1312.4659
26. Wang, J., Huang, S., Wang, X., Tao, D.: PONet: robust 3D human pose estimation via learning orientations only. CoRR abs/2112.11153 (2021). https://arxiv.org/abs/2112.11153
27. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4732 (2016)
28. Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., Zhang, W.: Deep kinematics analysis for monocular 3D human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 899–908 (2020)
29. Yang, W., Ouyang, W., Li, H., Wang, X.: End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3073–3082 (2016). https://doi.org/10.1109/CVPR.2016.335
30. Yang, W., Ouyang, W., Wang, X., Ren, J.S.J., Li, H., Wang, X.: 3D human pose estimation in the wild by adversarial learning. CoRR abs/1803.09722 (2018). https://arxiv.org/abs/1803.09722