symmetry

MDPI

Article

Multi-View Pose Generator Based on Deep Learning for Monocular 3D Human Pose Estimation

Jun Sun [1,2], Mantao Wang [1,2,*], Xin Zhao [3] and Dejun Zhang [3]

[1] College of Information and Engineering, Sichuan Agricultural University, Yaan 625014, China; 2019319014@stu.sicau.edu.cn
[2] The Lab of Agricultural Information Engineering, Sichuan Key Laboratory, Yaan 625014, China
[3] School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China; zxin@cug.edu.cn (X.Z.); zhangdejun@cug.edu.cn (D.Z.)
* Correspondence: wangmantao@sicau.edu.cn

Received: 9 June 2020; Accepted: 3 July 2020; Published: 4 July 2020

Abstract: In this paper, we study the problem of monocular 3D human pose estimation based on deep learning. Due to single view limitations, the monocular human pose estimation cannot avoid the inherent occlusion problem. The common methods use the multi-view based 3D pose estimation method to solve this problem. However, single-view images cannot be used directly in multi-view methods, which greatly limits practical applications. To address the above-mentioned issues, we propose a novel end-to-end 3D pose estimation network for monocular 3D human pose estimation. First, we propose a multi-view pose generator to predict multi-view 2D poses from the 2D poses in a single view. Secondly, we propose a simple but effective data augmentation method for generating multi-view 2D pose annotations, on account of the existing datasets (e.g., Human3.6M, etc.) not containing a large number of 2D pose annotations in different views. Thirdly, we employ graph convolutional network to infer a 3D pose from multi-view 2D poses. From experiments conducted on public datasets, the results have verified the effectiveness of our method. Furthermore, the ablation studies show that our method improved the performance of existing 3D pose estimation networks.

Keywords: multi-view; single view; pose estimation; pose generator

1. Introduction

3D human pose estimation is a popular research field in computer vision. Its development has played a promoting role in many applications, such as action recognition [1], motion capture [2], virtual reality, human-computer interaction, clinical research [3–5], and video surveillance. Due to the development of powerful Convolutional Neural Networks (CNNs), 2D pose estimation has made significant progress. Therefore, increasingly researchers have been investing their energies to 3D pose estimation, and various advanced technologies have been widely used, e.g., deep conditional variational autoencoder [6], Generative Adversarial Nets (GANs) [7,8], Graph Convolutional Networks (GCNs) [9], the Fully Connected Network (FCN) [10], and the self-supervised approach [11,12].

In recent years, research that studied 3D pose estimation has mainly focused on three different directions, namely 2D-to-3D pose estimation [10,13], monocular image-based 3D pose estimation [8,10,14,15], and multi-view images based 3D pose estimation [16–19]. These methods were mainly evaluated on the Human3.6M dataset [20], which was collected in a highly constrained environment with limited subjects and background variations. The current methods still have problems such as insufficient fitting, self-occlusion, limited representation ability, and difficulty in training.

Multi-view 3D pose estimation methods have proven to be effective to improve 3D pose estimation [17,19,21]. The advantages of these methods are avoiding partial occlusion, have easier

access to more available information, and better performance, compared to using a single image. However, these methods need multi-view datasets during training, but such datasets are more difficult to obtain.

Human pose-like graphic data are composed of joint points and skeletons. Zhao et al. [15] improved GCNs and proposed a novel graph neural network architecture for regression that takes full advantage of local and global relationships of nodes, called Semantic Graph Convolutional Networks (SemGCN). Ci et al. [22] overcame the limitation of GCNs representation power by introducing a Locally Connected Network (LCN). To sum up, GCNs have been demonstrated to be an effective approach with fewer parameters, higher precision, and easier training in the application of 3D pose estimation.

In this work, we propose a method that achieves multi-view 3D pose estimation on single-view data input. As shown in Figure 1, our framework includes two stages: (i) Multi-view pose generator (MvPG) and (ii) GCNs for multi-view 2D to 3D pose. Our experiments show that MvPG can significantly improve the overall effect of the 3D pose estimation model. In a word, our method is general, and effectively improves the effect of 3D pose estimation. Our contributions can be summarized as follows:

- We introduce an end-to-end network to implement a multi-view 3D pose estimation framework with single-view 2D pose as input;
- We establish a strong MvPG to predict the 2D poses of multiple views from the 2D poses in a single view;
- We present a simple and effective multi-view 2D pose datasets generation method.
- We propose a novel loss function for constraining both joint points and bone length.

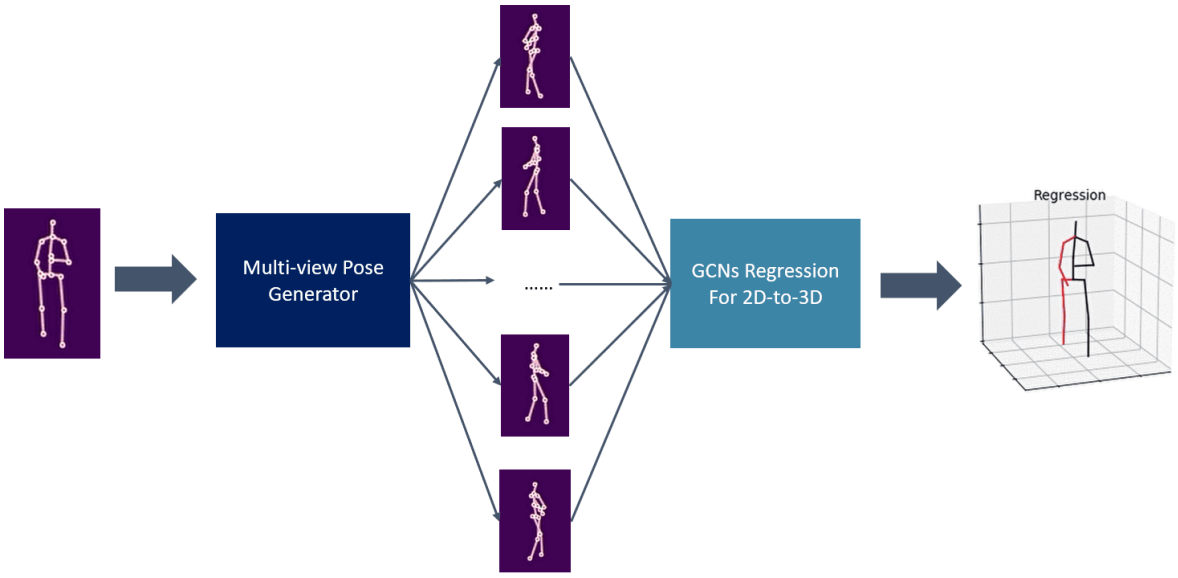

Figure 1. A schematic illustration of our framework. (**Left**): A Multi-view pose generator (MvPG) to predict several 2D poses. (**Right**): 2D to 3D regression model takes a merged multi-view pose data as input.

2. Related Work

There are two distinct categories of human pose estimation: Single-view methods and multi-view methods. Due to our method containing both the above elements and GCNs, we briefly summarize the past approaches for single-view, multi-view, and GCNs. Most of these approaches train model from large-scale datasets Human3.6m [20] to regress 3D human joint transformations.

2.1. Multi-View 3D Pose Estimation

These methods usually consist of two steps: (1) Estimating the 2D poses in multi-view images and (2) recovering the 3D pose from multi-view poses. It is easy to envision that the increase in the number of views could solve the self-occlusion problem, which is inherent in pose estimation. But the lack of datasets is a major problem in multi-view methods. To alleviate this problem, the majority of conducted research have focused on using weakly or self-supervised training methods to harvest annotations from different perspectives [12,23], or fusing features to achieve better results with as few perspectives as possible [17,19,21,24,25], such as, fusing the Inertial Measurement Unit (IMU) data and vision data to achieve better results [21,24], using multi-camera setup as an additional training source and fusing it with 3D models generated by individual cameras [25], and cross-view fusion [17].

This paper proposes an effective and efficient approach, which directly uses 2D poses to predict 3D poses. Specifically, we design a module named MvPG that generates 2D poses with multi-view from a monocular image. Then, using the generated 2D poses to estimate the 3D poses. Our entire model can effectively avoid dependence on a multi-view dataset, while alleviating the self-occlusion problem.

2.2. Single-View 3D Pose Estimation

Inspired by Martinez, most of the current solutions for the monocular 3D pose estimation mainly focused on two-stage methods. They established a simple baseline for 2D-to-3D human pose estimation by using neural networks to learn effectively 2D-to-3D mapping. The inevitable depth ambiguity in 3D pose estimation from single view images limit the estimation accuracy. Extensive research have exploited extra information to constrain the training process [15,26–29]. A more common piece of extra information is temporal information. For example, Yan et al. [26] introduced the Spatial-Temporal Graph Convolutional Networks (ST-GCN) to automatically learn both spatial and temporal patterns from data. Cheng et al. [29] exploit estimated 2D confidence heatmaps of keypoints and an optical-flow consistency constraint to filter out unreliable estimations of occluded keypoints. Lin et al. [28] utilize matrix factorization (such as singular value decomposition or discrete cosine transform) to process all input frames simultaneously to avoid sensitivity and drift issues. In addition, Sharma et al. [27] employ Deep Conditional Variational Autoencoder (CVAE) [30] to learn anatomical constraints and sample 3D pose candidates.

Additionally, we add extra information during estimation, that is, other views of the pose in a monocular image. Finally, we optimize this method computationally inexpensive but still be able to improve the performance. To the best of our knowledge, there is no previous work that generates multi-view 2D keypoints from a monocular image to estimate 3D pose.

2.3. GCNs for 3D Pose Estimation

GCNs generalize convolutions to graph-structured data and have great performance for irregular data structures. In recent years, a number of researchers have introduced the idea of GCNs to the study of action recognition [26,31] and 3D human pose estimation [15,32–34]. Constructing the GCNs can learn both spatial and temporal features for action recognition, such as Spatial Temporal Graph Convolutional Networks (ST-GCN) [26] and Actional-Structural Graph Convolution Network (AS-GCN) [31]. They harness the locality of graph convolution together with temporal dynamics. For pose estimation, it has also made full use of spatio-temporal information in GCNs [32]. Above that, Liu et al. [34] encode the strength of the relationship among joints by graph attention block and Zhang et al. [33] invented a 4D association graph for real-time multi-person motion capture. In this paper, we use a SemGCN [15] as the 2D to 3D regression network. It has the advantage of capturing local and global semantic relations, and is able to easily expand small parameters.Therefore, it is very suitable for our proposed multi-view pose generator.

3. Framework

The framework is illustrated in Figure 1, the whole model is formulated as an end-to-end network, which consists of two modules: (1) MvPG and (2) 2D to 3D regression networks. The MvPG predicts 2D poses of multiple views from a single view. The 2D to 3D regression network predicts an accurate 3D pose from multi-view 2D poses. During the training process, we first pre-train the MvPG on the human pose dataset. Then, we train the entire network in an end-to-end manner. Finally, accurate 3D pose data can be obtained.

3.1. Multi-View Pose Generator

Previous research has shown that multi-view methods [17,21] can effectively improve the performance of 3D pose estimation, however, multi-view poses are not easily available in real scenes. Accordingly, we tried to obtain the multi-view 2D pose from a single view so that the multi-view method can be utilized.

2D human pose is defined as a skeleton with $N = 16$ joints that can fully describe various postures of the human body, parameterized by a $2N$ vector $\left[q_1^\top, \ldots, q_N^\top\right]^\top$ (see Figure 2a). The *i-th* 2D joints denoted as $q_i = (x_i, y_i)$. Inspired by [35], to predict the right view from the left view, we propose a MvPG that aims to predict multi-view 2D poses from a single-view 2D pose. As shown in Figure 2b, given a single-view 2D pose keypoints q_i, the goal of MvPG is to learn a mapping function $f : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{M \times 2N}$ for predicting a set of multi-view 2d pose $f(q_i)$ from q_i, where M is the number of multi-views. Each group of networks in the multi-view pose generator learns the corresponding parameters to predict the multi-view 2D pose $\{f_1(q_i), f_2(q_i), \ldots, f_M(q_i)\}$ (see Figure 2c).

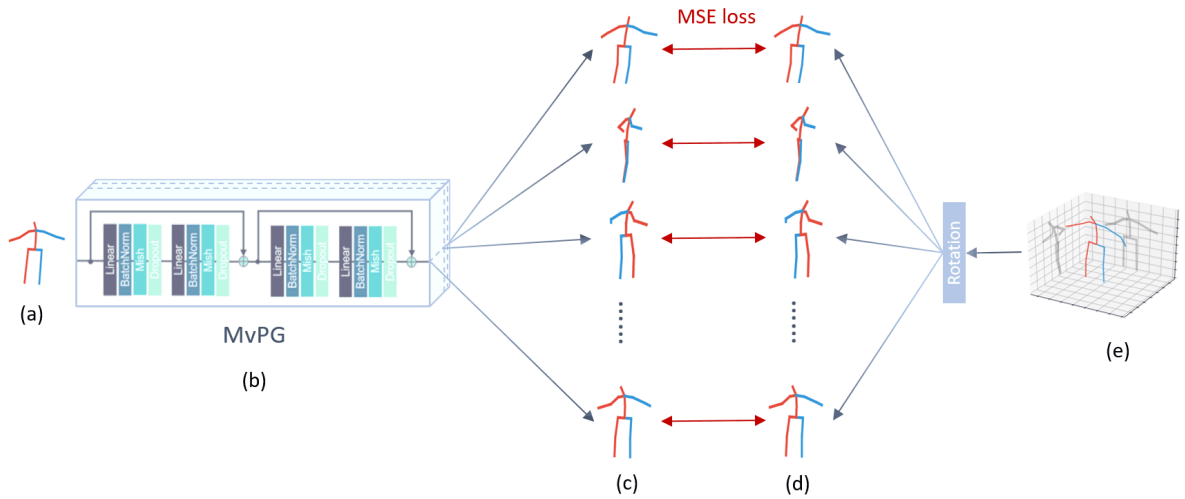

Figure 2. Multi-view pose generator. We use a Fully Connected Network (FCN) to build MvPG and predict the poses of other views from a 2D pose.

In order to train M pose generators, we need 2D pose data of M views $\{q_i^1, q_i^2, \ldots, q_i^M\}$ to supervise, such as Figure 2d. These data can be obtained by using camera parameter projection on the 3D pose (Figure 2e). The model aims to learn a regression function F_g which minimizes the error over $f_m(q_i)$ and q_i^m:

$$\mathcal{F}_g = \underset{f}{\operatorname{argmin}} \frac{1}{N} \frac{1}{M} \sum_{i=1}^{N} \sum_{m=1}^{M} \left|\left| f_m\left(q_i\right), q_i^m \right|\right|^2 \tag{1}$$

where, M is the number of 2D poses generated by the network, $f_m(q_i)$ is the prediction of the *m-th* view, q_i^m is the 2D pose annotation of the *m-th* view.

In order to train our MvPG, the datasets provide us with a series of 3D pose data, which contain the 3D coordinates of the joint points, skeleton information, and the camera coordinates in space and

other data. We use the 3D coordinates of the human body and the coordinates of the camera in space to generate a 2D pose corresponding to the camera view. We use the 3D pose coordinates and the corresponding 2D pose coordinates of each view to train MvPG based on Equation (1).

Before using the MvPG for a 3D pose estimation task, we need to pre-train it. Multi-view 2D pose annotations are annotation for body pose estimation, and are labels used for supervised learning. Our MvPG model is being trained on the Human3.6M dataset [20], while the dataset only provides limited camera angle parameters. Therefore we need to augment the training data.

3.2. 2D Pose Data Augmentation

The 3D pose is defined as a skeleton with $N = 16$ joints and parameterized by a $3N$ vector $\left[P_1^\top, \ldots, P_N^\top\right]^\top$. All keypoints are denoted as $P_i = (x_i, y_i, z_i)$. Existing 3D datasets such as Human3.6M [20] provides 3D coordinates of human joints and camera parameters that generate 2D poses in four perspectives. However, the limited number of cameras cannot meet the data requirements for training MvPG. Therefore, we introduce a rotation operation [29] to generate multi-view 2D pose annotations.

The Figure 3 describes our rotation operation. First, we obtain the 3D pose of ground-truth from the dataset and extract coordinate parameters $P_i = (x_i, y_i, z_i)$ of the key point. Second, we fix the Y-axis parameter y_i in the three-dimensional coordinate system and consider only the rotation operations of x_i and z_i. The x_i and z_i rotate in $[-\pi, \pi]$ with a sampling step of $2\pi/M$. The coordinates after rotation are (x_i^m, y_i^m), which can be described as:

$$\begin{bmatrix} x_i^m \\ z_i^m \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{2\pi m}{M}\right) & -\sin\left(\frac{2\pi m}{M}\right) \\ \sin\left(\frac{2\pi m}{M}\right) & \cos\left(\frac{2\pi m}{M}\right) \end{bmatrix} \begin{bmatrix} x_i \\ z_i. \end{bmatrix} \tag{2}$$

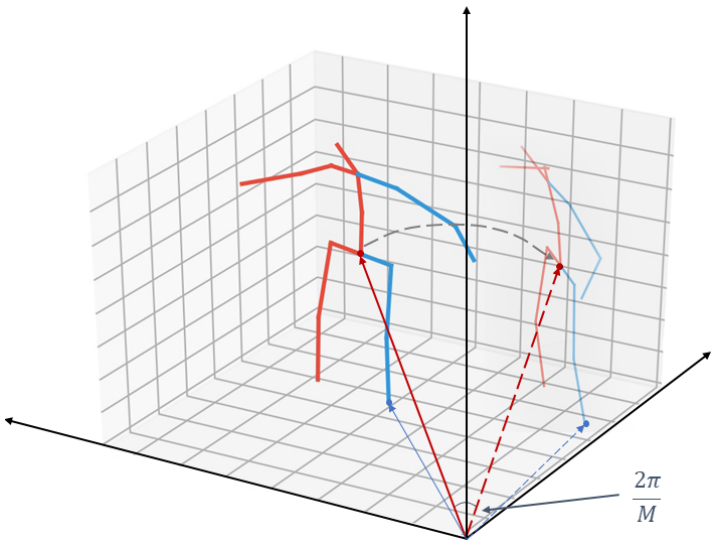

Figure 3. The illustration of $2\pi/M$ rotation.

Using the above formula, we get a set of 3D poses from different views: $\left\{(x_i^m, y_i^m, z_i^m)\right\}_{m=1}^{M}$. Third, map the multi-view 3D keypoints to the Y-axis and Z-axis planes. Finally, we obtain M virtual views with different angles in each 3D ground truth. For example, as shown in Figure 4, $M = 16$ virtual view angle 2D poses are generated from a 3D ground truth $\{P_i = (x_i, y_i, z_i)\}$.

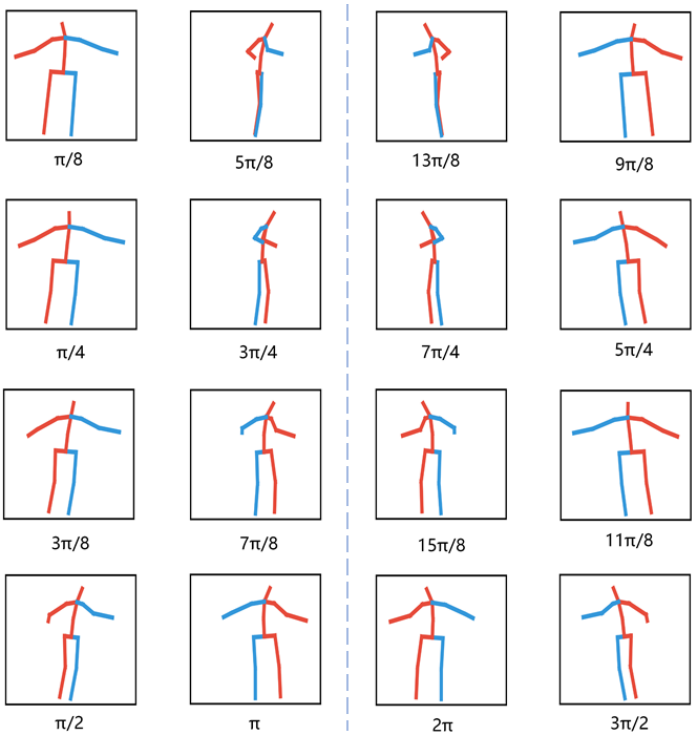

Figure 4. Multi-view 2D poses generated by rotation. Each pose is symmetrical with it after rotating π.

3.3. 2D to 3D Pose Regression Network

The goal of our method is to estimate body joint locations in 3D space. Formally, given a series of 2D keypoints of the monocular view $q_i = (x_i, y_i)$ and their corresponding 3D keypoints $P_i = (x_i, y_i, z_i)$. The 2D to 3D pose regression network $\mathcal{F}$ takes q_i as input and predicts the corresponding coordinates $\tilde{P}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ in 3D space. Our model can be described as a function $\mathcal{F}^*$:

$$\mathcal{F}^* = \underset{\mathcal{F}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\mathcal{F}\left(q_i\right), P_i\right). \tag{3}$$

The model aims to learn a regression function $\mathcal{F}^*$ which minimizes the error over $\tilde{P}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ and $P_i = (x_i, y_i, z_i)$.

3.3.1. Network Design

Firstly, we use the method described in Section 3.1 to build a MvPG. As shown in the upper part of Figure 5, in order to generate 2D poses from M perspectives, we need to combine MvPG in a symmetrical manner. The generated multi-views are symmetrical to each other at intervals of π, thereby alleviating the occlusion problem and blurring problem of the front and back of the limbs in the single view. Then multi-view pose data are concatenated, which enables them to contain more hidden information than single-view data. Each pose is represented by a 16×2 matrix. We simply combine the M pose data into a $16 \times 2M$ matrix, where every two columns represent a 2D pose in a single view angle and the row vectors represent the coordinates of each key point at 16 view angles. Then we take the $16 \times 2M$ matrix as the input of the 2D to 3D network.

Secondly, we use SemGCN [15] as a 2D to 3D Pose Regression Network. In order to obtain more high-level features and better performance, we deepen the SemGCN [15] network. In our experiments, we double the depth of the original SemGCN to get better performance, as shown in the lower part of Figure 5.

Finally, the network ends with a 1024-way fully-connected layer. This step is added to alleviate redundancy and prevent the network from overfitting.

In previous studies [8,15,22], the models have a large difference in the Mean Per Joint Position Error (MPJPE) for different poses. This indicates instability in model training. To alleviate this problem, we used the Mish [36] activation function instead of ReLU [37], defined as: $f(x) = x \cdot \tanh(\sigma(x))$. Where, $\sigma(x) = \ln(1 + e^x)$ is the softplus activation function [38].

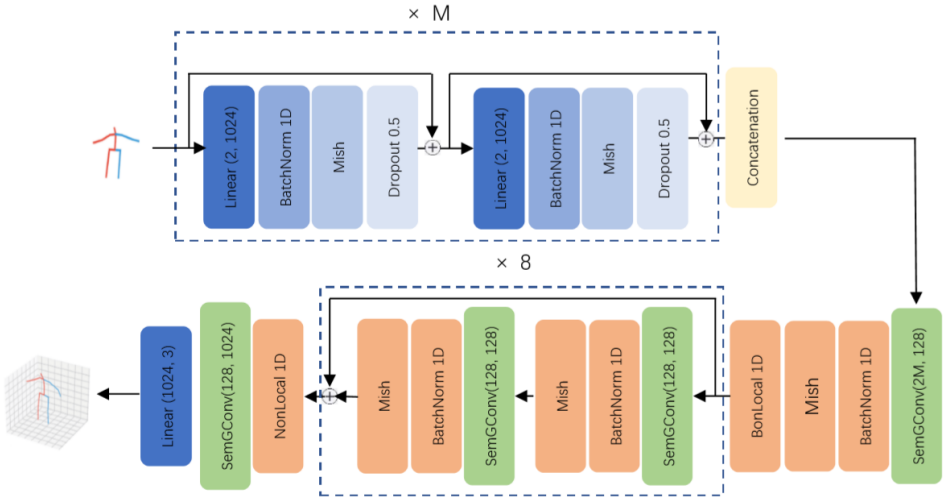

Figure 5. 3D pose estimation network with multi-view pose generator.

3.3.2. Loss Function

Most previous studies have used minimizing Mean Square Error (MSE), which has proved to be a simple and efficient method that performs well on this task. On the basis of MSE we add bone-length consistency loss to constrain bone length. After the MvPG obtain the multi-view 2D keypoints, we feed them into our improved GCNs, which outputs the estimated 3D joint coordinates for all keypoints. The 2D to 3D networks employs the MSE loss based on 3D joints expressed as:

$$\mathcal{L} = \sum_{i=1}^{N} \left\| \tilde{P}_i - P_i \right\|^2 + \sum_{j=1}^{B} \left\| \tilde{b}_j - b_j \right\|^2 \tag{4}$$

where P_i is the ground-truth 3D joint, $\tilde{P}_i$ is the corresponding predicted 3D joint by our model, and B is the number of bones of one skeleton. The bone length $\tilde{b}$ and b are calculated from the predicted 3D joint and ground-truth 3D joint, respectively.

In this way, we construct an end-to-end deep neural network for posture estimation from 2D to 3D. We use Adam optimizer to pre-train MvPG based on Equation (1) with the augmented data. Afterward, we train the whole network on the dataset to achieve the best results.

4. Experiments

In this section, we first introduce the dataset Human3.6M [20] used to evaluate network performance and the evaluation protocol. Second, according to Section 3.3.1, we design the network and conduct ablation studies on the components in our method. Finally, we report the results of our evaluation of the public datasets and compare them with state-of-the-art methods.

4.1. Setting

Datasets: The Human3.6M [20] dataset is one of the largest and widely used datasets for 3D human pose estimation. This dataset provides 3.6 million 3D human pose images and poses labels. It contains various poses captured from four cameras such as discussion, eating, sitting, smoking, etc. The ground-truth 3D poses are captured by the Mocap system, while the 2D poses can be obtained by projection with the known intrinsic and extrinsic camera parameters.

Evaluation Protocols: We follow the standard protocol on Human3.6M to use the subjects 1, 5, 6, 7, 8 for training and the subjects 9 and 11 for evaluation. The evaluation metric is the Mean Per Joint Position Error (MPJPE) in millimeter between the ground-truth and the prediction across all cameras and joints after aligning the depth of the root joints. We refer to this as Protocol #1. According to Protocol #1, only the frontal view is considered for testing, i.e., testing is performed on every 5th frame of the sequences from the frontal camera (cam-3) from trial 1 of each activity with ground-truth cropping. The training data includes all actions and perspectives. This protocol is named Protocol #2.

Experimental Settings: The model is trained by using Pytorch. To benefit from the efficiency of the parallel computation of the tensors, all simulation studies are conducted with RTX2080S GPU on an Ubuntu OS. Furthermore, in order to verify the effectiveness and efficiency of our method, we designed two sets of experiments: (1) Effects of different scales on MvPG results and (2) the influence of MvPG on the performance of 3D pose estimation on different networks.

Implementation Details: We use the ground truth 2D and 3D joint locations provided in the dataset as input of the MvPG for pre-training, and use the method of Section 3.2 for data augmentation during the training process. Before training the entire network, we import the pre-training parameters into the MvPG part. In this stage, the loss function is defined by Equation (4).

We train our model for 15 epochs using the Adam optimizer, set the learning rate of 0.008 with exponential decay, and set the mini-batches size to 256. During testing, it processes an epoch per 15 min using batch processing mode (256 samples per batch) on a single RTX 2080S GPU. It is worth noting that during the training process, initializing different random number seeds in the network will have different effects on the training results. After a lot of experiments, we finally trained the best parameters on our device.

4.2. Ablation Study

In this section, we designed two sets of experiments. Firstly, in Section 4.2.1, in order to verify the effect of the number of views generated by MvPG on the algorithm, we design MvPG to generate different numbers of views for ablation experiments. Secondly, we apply MvPG to different 2D to 3D networks to verify the commonality of our method.

4.2.1. Performance Analysis of the Number of Views Generated by MvPG

Figure 5 shows the network architecture of SemGCN [15] with MvPG, which is composed of two main modules: (1) A basic version that SemGCN [15] without MvPG and (2) the MvPG on different scales. To evaluate the efficacy of MvPG, we conduct an ablation study on Human3.6M [20] under Protocol #1. The Table 1 lists the average error of all joints. The notations are as follows:

Basic version: Refers to the pose estimator without the MvPG. The mean error of our basic version model is 40.81 mm, which is very close to the 40.78 mm error reported on SemGCN [15] with non-local [39].

MvPG: Refers to the model with the MvPG. MvPG-4, MvPG-8, and MvPG-16 respectively represent the number of poses generated by the MvPG, and there are 4, 8, and 16 poses, respectively.

Table 1. Ablation studies on the Human3.6M [20] dataset under Protocol #1. The first rows refer to the baseline pose estimator without MvPG. The rest of the rows refer to variants with MvPG. MPJPE (mm) denotes protocol #1. The top 1 ranked values are highlighted in bold number.

Basic Version (SemGCN [15])	MvPG-4	MvPG-8	MvPG-16	MPJPE (mm)
√				40.8
√	√			36.9
√		√		39.0
√			√	**35.8**

We compare different scales for the MvPG, and the results are shown in Table 1. The first line shows the results of a basic version with only the SemGCN [15] and non-local [39]. Unsurprisingly, the performance without the MvPG modules is poor. The second line shows the results of integrating the MvPG-4 modules, and the third line shows the results of integrating the MvPG-8 modules. As the results show, the introduction of the MvPG-4 and MvPG-8 modules performance improved by 9.38% and 4.2%, respectively. When MvPG-16 was introduced, as shown in the last line, our model achieved an estimation mean error of 35.8. Based on these experimental results, we set multi-views generated by MvPG to 16 at last.

We analyze the results of this experiment as follows: (1) Although different module scales make different contributions to the mean errors, the final mean errors could be further improved by selecting the appropriate module scales. (2) In the 3D pose estimation task, the more multi-views were more important than the fewer views, which fully demonstrates that the MvPG-16 module effectively extracted the multi-views feature. This result confirms the effectiveness of MvPG-16. (3) While increasing the number of views, it is necessary to further increase the depth of the subsequent 2D to 3D network to match the MvPG and learn more features, see Table 1.

4.2.2. Impact of MvPG on 3D Pose Estimation Network

To analyze the impact of using different 2D to 3D networks on MvPG in the entire pose estimation task we use SemGCN [15] and FCN [10], respectively, as 2D to 3D networks and then conduct ablation analysis on Human3.6M [20] under Protocol #1. As shown in Table 2, after adding a MvPG, FCN [10] get a 5.67% improvement, and SemGCN [15] get a 5.22% improvement. This experiment shows that our MvPG is generally applicable to various 2D to 3D pose estimation networks.

Table 2. After using MvPG, the performance of the previous methods improved. The top 2 ranked values are highlighted in bold and the first and second are shown in red and blue, respectively.

Method	MPJPE (mm)
FCN [10]	44.4
SemGCN [15]	42.1
MvPG-16 + FCN [10]	**41.8**
MvPG-16 + SemGCN [15]	**39.9**

4.3. Comparison with the State of the Art

We performed quantitative comparisons on all state-of-the-art methods based on single-view 3D pose estimation. These models were trained and tested on ground truth 2D pose. The results are shown in Table 3. We found that our method, using only 2D joints as inputs SemGCN [15] with the non-local [39] layer as the 2D to 3D network, was able to match the state-of-the-art performance. In particular, we reviewed the previous method, for the action of Directions, Greeting, Posing, Waiting, Walking, Walking Dog, and Walking together. There was serious self-occlusion in these actions, and our MvPG could compensate for this problem by predicting the pose of the multi-view. For Protocol #1, our method (GT) obtained the state-of-the-art results with a 35.8 mm of error, which had 12% improvements compared to the SemGCN architecture [15]. Compared to the recent best result [22], our method still had a 1.3% improvement.

Table 3. Quantitative evaluations on the Human3.6M [20] under Protocol#1. GT indicates that the network was trained and tested on ground truth 2D pose. Non-local indicates that on our 2D to 3D network architecture in SemGCN [15] with non-local [39]. The top 1 ranked values are highlighted in bold number.

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez et al. [10]	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
Sun et al. [40]	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7	86.7	61.5	67.2	53.4	47.1	61.6	53.4	59.1
Fang et al. [41]	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Yang et al. [8]	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Zhao et al. [15]	47.3	60.7	51.4	60.5	61.1	49.9	47.3	68.1	86.2	55.0	67.8	61.0	42.1	60.6	45.3	57.6
Ci et al. [22]	46.8	52.3	44.7	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Martinez et al. [10] (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
Zhao et al. [15] (GT)	37.8	49.4	37.6	40.9	45.1	41.4	40.1	48.3	50.1	42.2	53.5	44.3	40.5	47.3	39.0	43.8
Wang et al. [35] (GT)	32.1	39.2	33.4	36.4	38.9	45.9	38.4	31.7	42.5	48.1	37.8	37.9	38.7	30.6	32.6	37.6
Ci et al. [22] (GT)	36.3	**38.8**	**29.7**	37.8	**34.6**	**42.5**	39.8	32.5	**36.2**	**39.5**	**34.4**	38.4	38.2	31.3	34.2	36.3
Ours (GT)	37.1	42.9	35.7	37.3	39.4	47.6	40.3	37.1	48.1	51.7	38.8	39.0	40.5	29.7	33.6	39.9
Ours (GT/non-local)	**30.9**	39.2	31.1	**33.7**	36.2	45.1	**35.7**	**30.1**	40.4	48.1	35.1	**35.2**	**37.4**	**27.9**	**30.2**	**35.8**

Compared with the latest single-view models, our model combined the advantages of the multi-view model. The experiments showed that our model could effectively improve the accuracy of single-view 3D pose estimation. Additionally, our model could be directly used in real scenes because it only needed one view to achieve high-precision 3D pose estimation. It is clear that our approach also has certain drawbacks, as our approach raised network size resulting in longer network training time. In the next step, we will improve this problem.

4.4. Qualitative Results

Figure 6 shows the visualization results of our approach and compares them with 3D ground-truth on Human3.6M. Using single-view 2D pose as input, our approach is able to generate multi-view 2D pose data and mine hidden occlusion information for reconstructing 3D pose. As we can see, our method could accurately estimate the 3D pose, which shows that MvPG could handle self-occlusion more effectively.

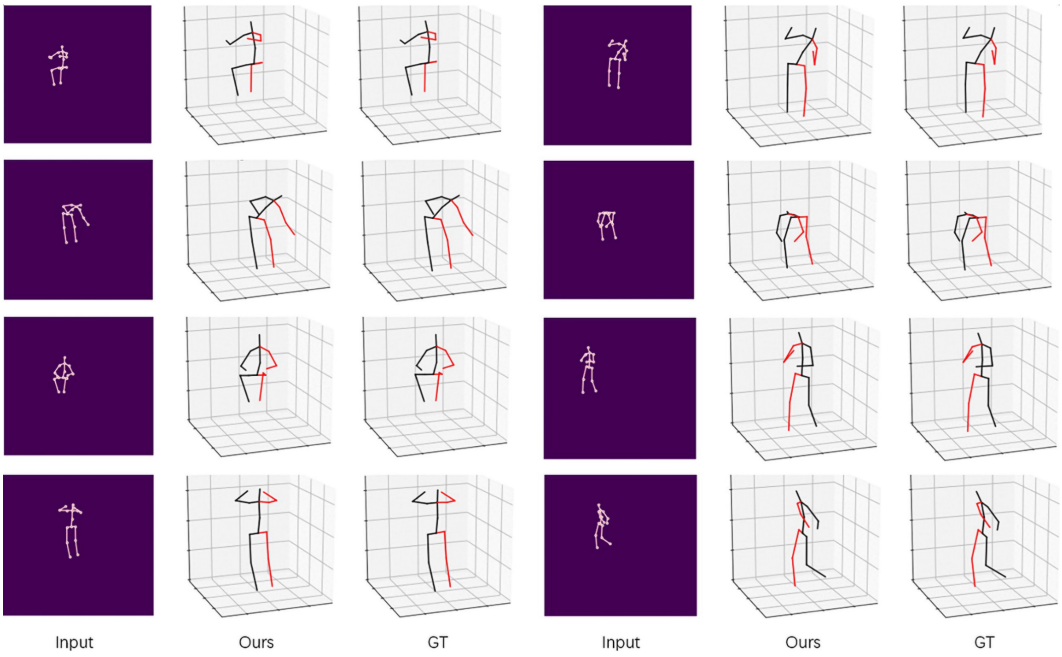

Figure 6. Visual results of our method on Human3.6M [20]. Red and black indicate left and right, respectively.

5. Conclusions

In this paper, we proposed a Multi-view Pose Generator (MvPG) for 3D pose estimation from a novel perspective. Our method was able to predict a set of symmetric multi-view poses using a single-view 2D pose, which is used in 2D to 3D regression networks to solve the problem of self-occlusion in pose estimation. Combined with the advanced SemGCN model, the performance of 3D human pose estimation is further improved. The results of training and testing with ground truth 2D poses as input show that our method improved by 1.3% compared with the state of the art. Compared with multi-view 3D pose estimation, our method still has deficiencies. Our method can be applied to a 3D human pose estimation task that provides the only single view. For example in surveillance video equipment, clinical research, interactive games, etc. In future work, we plan to study the use of more advanced network design multi-view pose generators to achieve higher performance with a smaller network scale.

Author Contributions: Conceptualization, J.S.; Methodology, J.S. and D.Z.; Formal analysis, J.S.; Data curation, J.S. and X.Z.; Visualization, X.Z.; Writing—original draft, J.S. and X.Z.; Validation, D.Z. and M.W.; Writing—review & editing, J.S. and D.Z.; Supervision, D.Z. and M.W.; Funding acquisition, M.W. and D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 61702350.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, D.; He, L.; Tu, Z.; Han, F.; Zhang, S.; Yang, B. Learning motion representation for real-time spatio-temporal action localization. *Pattern Recognit.* **2020**, *103*, 107312. [CrossRef]
2. Buys, K.; Cagniart, C.; Baksheev, A.; Laet, T.D.; Schutter, J.D.; Pantofaru, C. An adaptable system for RGB-D based human body detection and pose estimation. *J. Vis. Commun. Image Represent.* **2014**, *25*, 39–52. [CrossRef]
3. Ancillao, A. *Stereophotogrammetry in Functional Evaluation: History and Modern Protocols*; Springer: Cham, Switzerland, 2018.
4. Procházka, A.; Vyšata, O.; Vališ, M.; Ťupa, O.; Schätz, M.; Mařík, V. Bayesian classification and analysis of gait disorders using image and depth sensors of Microsoft Kinect. *Digit. Signal Process.* **2015**, *47*, 169–177. [CrossRef]
5. Clark, R.A.; Bower, K.J.; Mentiplay, B.F.; Paterson, K.; Pua, Y.H. Concurrent validity of the Microsoft Kinect for assessment of spatiotemporal gait variables. *J. Biomech.* **2013**, *46*, 2722–2725. [CrossRef] [PubMed]
6. Sohn, K.; Lee, H.; Yan, X. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Dutchess County, NY, USA, 2015; pp. 3483–3491.
7. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Dutchess County, NY, USA, 2014; pp. 2672–2680.
8. Yang, W.; Ouyang, W.; Wang, X.; Ren, J.; Li, H.; Wang, X. 3D Human Pose Estimation in the Wild by Adversarial Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5255–5264.
9. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
10. Martinez, J.; Hossain, R.; Romero, J.; Little, J.J. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2640–2649.
11. Wang, K.; Lin, L.; Jiang, C.; Qian, C.; Wei, P. 3D Human Pose Machines with Self-supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 1069–1082.
12. Kocabas, M.; Karagoz, S.; Akbas, E. Self-Supervised Learning of 3D Human Pose Using Multi-View Geometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1077–1086.
13. Fang, H.; Xu, Y.; Wang, W.; Liu, X.; Zhu, S.C. Learning Knowledge-guided Pose Grammar Machine for 3D Human Pose Estimation. *arXiv* **2017**, arXiv:1710.06513.
14. Zhou, X.; Huang, Q.; Sun, X.; Xue, X.; Wei, Y. Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 398–407.
15. Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; Metaxas, D.N. Semantic Graph Convolutional Networks for 3D Human Pose Regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3425–3435.
16. Rhodin, H.; Spörri, J.; Katircioglu, I.; Constantin, V.; Meyer, F.; Müller, E.; Salzmann, M.; Fua, P. Learning Monocular 3D Human Pose Estimation From Multi-View Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8437–8446.

17. Qiu, H.; Wang, C.; Wang, J.; Wang, N.; Zeng, W. Cross View Fusion for 3D Human Pose Estimation. In Proceedings of the International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 915–922.
18. Dong, J.; Jiang, W.; Huang, Q.; Bao, H.; Zhou, X. Fast and Robust Multi-Person 3D Pose Estimation from Multiple Views. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7792–7801.
19. Iskakov, K.; Burkov, E.; Lempitsky, V.; Malkov, Y. Learnable Triangulation of Human Pose. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 7718–7727.
20. 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.* **2014**, *36*, 1325–1339. [CrossRef] [PubMed]
21. Huang, F.; Zeng, A.; Liu, M.; Lai, Q.; Xu, Q. DeepFuse: An IMU-Aware Network for Real-Time 3D Human Pose Estimation from Multi-View Image. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 429–438.
22. Ci, H.; Wang, C.; Ma, X.; Wang, Y. Optimizing Network Structure for 3D Human Pose Estimation. In Proceedings of the IEEE International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 2262–2271.
23. Pavlakos, G.; Zhou, X.; Derpanis, K.G.; Daniilidis, K. Harvesting multiple views for marker-less 3d human pose annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6988–6997.
24. Trumble, M.; Gilbert, A.; Malleson, C.; Hilton, A.; Collomosse, J. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. *BMVC* **2017**, *2*, 3.
25. Tome, D.; Toso, M.; Agapito, L.; Russell, C. Rethinking pose in 3d: Multi-stage refinement and recovery for markerless motion capture. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 474–483.
26. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
27. Pham, H.H.; Salmane, H.; Khoudour, L.; Crouzil, A.; Velastin, S.A.; Zegers, P. A unified deep framework for joint 3d pose estimation and action recognition from a single rgb camera. *Sensors* **2020**, *20*, 1825. [CrossRef] [PubMed]
28. Lin, J.; Lee, G.H. Trajectory Space Factorization for Deep Video-Based 3D Human Pose Estimation. *arXiv* **2019**, arXiv:1908.08289.
29. Cheng, Y.; Yang, B.; Wang, B.; Yan, W.; Tan, R.T. Occlusion-Aware Networks for 3D Human Pose Estimation in Video. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 723–732.
30. Zhang, D.; Zou, L.; Zhou, X.; He, F. Integrating feature selection and feature extraction methods with deep learning to predict clinical outcome of breast cancer. *IEEE Access* **2018**, *6*, 28936–28944. [CrossRef]
31. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Actional-structural graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, 16–20 June 2019; pp. 3595–3603.
32. Cai, Y.; Ge, L.; Liu, J.; Cai, J.; Cham, T.J.; Yuan, J.; Thalmann, N.M. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 2272–2281.
33. Zhang, Y.; An, L.; Yu, T.; Li, X.; Li, K.; Liu, Y. 4D Association Graph for Realtime Multi-person Motion Capture Using Multiple Video Cameras. *arXiv* **2020**, arXiv:2002.12625.
34. Liu, J.; Guang, Y.; Rojas, J. GAST-Net: Graph Attention Spatio-temporal Convolutional Networks for 3D Human Pose Estimation in Video. *arXiv* **2020**, arXiv:2003.14179.
35. Wang, L.; Chen, Y.; Guo, Z.; Qian, K.; Lin, M.; Li, H.; Ren, J.S. Generalizing Monocular 3D Human Pose Estimation in the Wild. *arXiv* **2019**, arXiv:1904.05512.
36. Misra, D. Mish: A Self Regularized Non-Monotonic Neural Activation Function. *arXiv* **2019**, arXiv:1908.08681.

37. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
38. Liu, Q.; Furber, S. Noisy Softplus: A Biology Inspired Activation Function. In *International Conference on Neural Information Processing*; Springer: Cham, Switzerland, 2016; Volume 9950, pp. 405–412.
39. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794-7803.
40. Sun, X.; Shang, J.; Liang, S.; Wei, Y. Compositional Human Pose Regression. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2602–2611.
41. Fang, H.; Xu, Y.; Wang, W.; Liu, X.; Zhu, S.C. Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.