electronics

MDPI

Article

Pointless Pose: Part Affinity Field-Based 3D Pose Estimation without Detecting Keypoints

Jue Wang and Zhigang Luo *

College of Computer, National University of Defense Technology, Changsha 410000, China; jue.wang.0911@gmail.com
* Correspondence: zgluo@nudt.edu.cn

Abstract: Human pose estimation finds its application in an extremely wide domain and is therefore never pointless. We propose in this paper a new approach that, unlike any prior one that we are aware of, bypasses the 2D keypoint detection step based on which the 3D pose is estimated, and is thus *pointless*. Our motivation is rather straightforward: 2D keypoint detection is vulnerable to occlusions and out-of-image absences, in which case the 2D errors propagate to 3D recovery and deteriorate the results. To this end, we resort to explicitly estimating the human body regions of interest (ROI) and their 3D orientations. Even if a portion of the human body, like the lower arm, is partially absent, the predicted orientation vector pointing from the upper arm will take advantage of the local image evidence and recover the 3D pose. This is achieved, specifically, by deforming a skeleton-shaped puppet template to fit the estimated orientation vectors. Despite its simple nature, the proposed approach yields truly robust and state-of-the-art results on several benchmarks and in-the-wild data.

Keywords: 3D human pose estimation; part affinity field; robust; pointless

check for updates

Citation: Wang, J.; Luo, Z. *Pointless* Pose: Part Affinity Field-Based 3D Pose Estimation without Detecting Keypoints. *Electronics* **2021**, *10*, 929. https://doi.org/10.3390/electronics10080929

Academic Editor: D. J. Lee

Received: 23 March 2021
Accepted: 9 April 2021
Published: 13 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Human pose estimation aims at recovering the coordinates of a human body captured from one or multiple images, and therefore plays a vital role in an exceptionally broad spectrum of applications. Thanks to the recent development of deep learning, 2D human pose estimation has witnessed unprecedented advances [1–3]. Despite the encouraging progress, estimating 3D poses from a single image, being an ill-posed problem by nature, remains a challenging task.

Traditional 3D pose estimation methods depend on first detecting 2D body keypoints from the input image, followed by mapping the 2D detections back to the 3D world. The advantage of building 3D pose estimation basing on 2D keypoint detection is that the latter is a mature technique with high generalization capacity, which means that if a 3D pose estimation method only requires 2D keypoint locations as input, it would automatically inherit the generalization capacity. However, depth information, which is crucial to 3D pose estimation, is completely lost in the 2D keypoint estimation process, making the subsequent 2D-to-3D regression ill-posed and thus ambiguous. Estimating the depth along with 2D keypoint can solve this problem in theory, but this is a task of almost the same level of difficulty with 3D pose estimation. Some methods [4–10] tried to estimates the relative depth simultaneously with 2D keypoints. However these methods require camera parameters in post-processing, which greatly limits the range of application. In addition, the 2D keypoints may be in most cases robustly detected, only if the keypoints are present in the image. Such prerequisite is unfortunately too strong for real-world application scenarios, where heavy occlusions and out-of-image absences of body joints frequently occur and thus collapse the 3D estimation results.

We propose in this paper an end-to-end *pointless* 3D pose estimation approach, bypassing the 2D keypoint-detection step to avoid those problems mentioned above. We substitute the detection of the only intermittently visible 2D points, with the estimation of regions of body parts, and the corresponding 2D, 3D orientations, which are represented by *part affinity fields* (PAFs) [3], as shown in Figure 1c. The PAF encodes both region and orientation information at the same time. This PAF-based body regions and orientations learning makes our method robust to the cases with visually absent keypoints. The rationale behind is, the vectorized ROIs and orientations enable the 3D pose recovery by utilizing the estimated directional vector oriented from a neighboring body part, even if the part of interest is visually incomplete and hence the keypoints are absent.

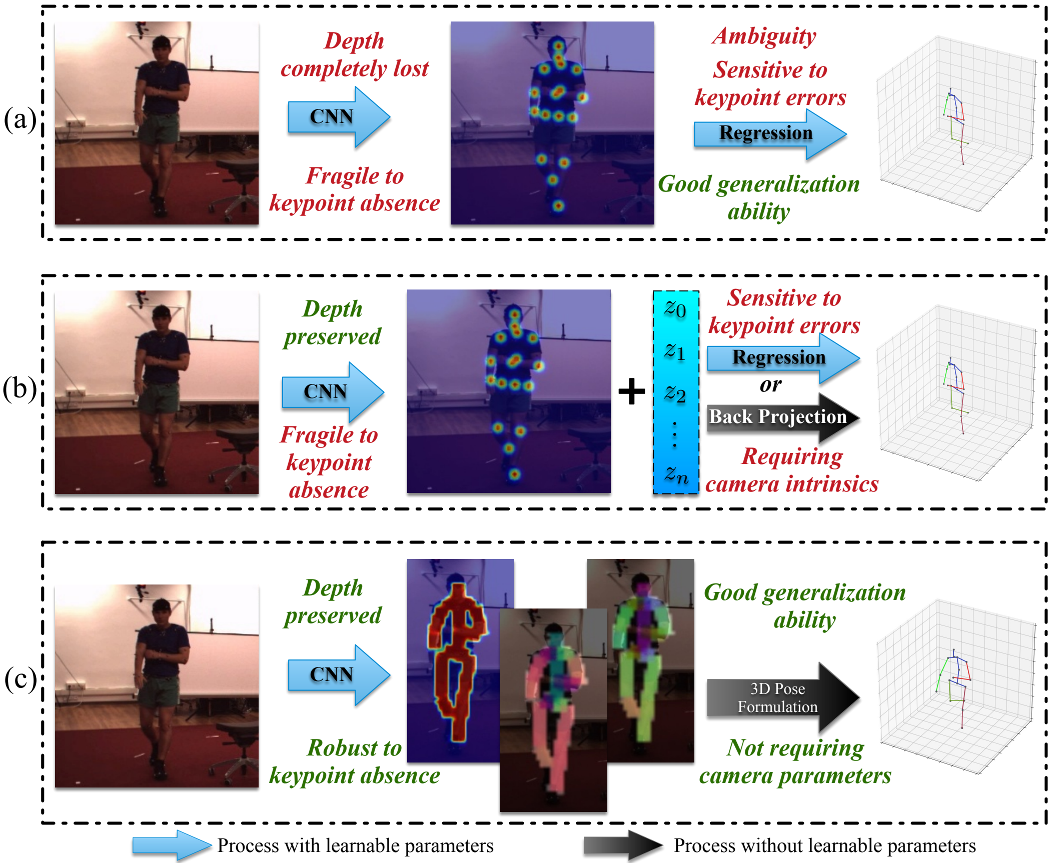
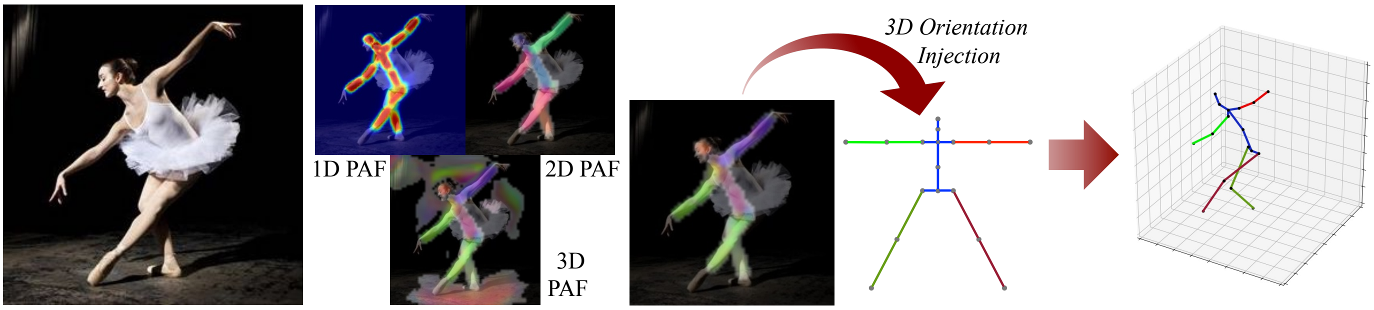

(a)
Depth completely lost
CNN
Fragile to keypoint absence
Ambiguity
Sensitive to keypoint errors
Regression
Good generalization ability
(b)
Depth preserved
CNN
Fragile to keypoint absence
z_0
z_1
z_2
z_n
Sensitive to keypoint errors
Regression
or
Back Projection
Requiring camera intrinsics
(c)
Depth preserved
CNN
Robust to keypoint absence
Good generalization ability
3D Pose Formulation
Not requiring camera parameters
Process with learnable parameters
Process without learnable parameters

Figure 1. The pros and cons of 2D keypoint based pipelines (**a**,**b**) and the proposed *pointless* one (**c**). Our method bypasses 2D pose estimation process, by directly estimating from images 1D, 2D and 3D part affinity fields, which represent the regions, 2D and 3D orientations (encoded in color) of different body regions, respectively. This fully part affinity fields (PAFs)-based estimation proves to be robust to keypoint absence. In addition, the proposed new pipeline achieves outstanding generalization ability to in-the-wild images with a simple semi-supervised approach. Our method does not require any camera parameters, which means it can be easily applied to other testing images.

An overview of the proposed method is illustrated in Figure 2. Specifically, we use a fully convolutional neural network (FCNN) to simultaneously predict 1D, 2D and 3D PAFs, which represent the regions, 2D and 3D orientations (encoded in color) of different limbs, respectively. To improve the generalization ability of the network, we train the FCNN on the mixture of a 2D pose dataset MPII [11] and a 3D pose dataset Human3.6M [12] in a semi-supervised approach. Once the PAFs are predicted by the FCNN, we first refine the 3D PAFs to remove the noises in it. The estimated PAFs are then aggregated to form the 3D orientation vectors over the human body, which are further aligned with a skeleton-shaped *puppet* template to produce the final 3D pose estimation result. The puppet adopted here features limbs of fixed sizes and adjustable body joints and is, in this process, deformed in a way that exactly fits the estimated 3D orientation vectors. The reason we choose to freeze the limb lengths of the puppet lies in that, the absolute length estimation from a single color image is in many cases unreliable, given that persons of different heights could have identical 2D projections. It is worth noting that all the post-processing steps are differentiable, which makes end-to-end training possible.

Despite its very simple nature, the proposed approach yields truly encouraging performances on lab-monitored benchmarks as well as in-the-wild images where large portions of human bodies are absent in the scene. To gain deeper insight into the behavior of the approach, we also introduce a new orientation-based evaluation metric for 3D pose estimation, which explicitly accounts for the angle between the estimated and the ground truth 3D limb vectors. Under both the keypoint- and orientation-based metrics, the proposed approach accomplishes state-of-the-art 3D pose estimation results.



(a) Input Image (b) Predicted PAFs (c) Refined 3D PAFs (d) The Human Model (e) 3D Pose Output

Figure 2. The pipeline of our method. The system takes a color image (**a**) as input and simultaneously predicts 1D/2D/3D PAFs (**b**). Then it refines the 3D PAFs by denoising and 2D/3D PAFs ensemble (**c**) (see Section 3.3.1). Finally, the 3D directional vectors are extracted from the refined 3D PAFs and injected into a skeleton-shaped puppet (**d**) (see Section 3.3.2) to produce the 3D pose prediction (**e**).

In summary, our contribution is an end-to-end pointless approach towards the never-pointless 3D human pose estimation. Our method bypasses the error-prone 2D keypoint detection step by substituting it with an orientation-based estimation to recover the 3D orientations of a subject, and then deforms a fixed-size puppet template to fit the predicted directional vectors so as to produce final 3D pose estimation. Such orientation-based estimations allows us to in many cases remedy the partially absent body parts that occur frequently in real-world scenarios. Experiments on several benchmarks and in-the-wild data show that the proposed approach, despite simple, achieves state-of-the-art results in terms of both the conventional keypoint-based and the newly proposed orientation-based evaluation metrics.

2. Related Work

In this section, we briefly review here approaches related to ours. We categorize them into two overlapping groups—methods relying on 2D keypoint detection and those explicitly using part affinity fields, where all methods in the latter group, in fact, utilize 2D keypoint detections as well.

2.1. 2D Keypoint Estimation Based Methods

There are two most widely used pipelines as shown in Figure 1a,b. Methods that follow pipeline (a) divide the 3D pose estimation task into two steps, 2D pose estimation and 3D pose inference. These methods comprise a 2D keypoint detector and a subsequent optimization [13–15] or regression [16–26] step to estimate 3D pose. Early efforts on 3D pose estimation used dictionary learning, with the assumption that a 3D pose can be represented by a linear combination of a set of base poses [13–15]. Recently, many researchers have begun to use neural networks for 3D pose regression. For instance, Moreno-Noguer [21] used a Convolutional Neural Network (CNN) to regress the 3D joints distance matrices instead of 3D poses. Sun et al. [19] proposed to regress the bones instead of joints by re-parameterizing the pose presentation. Lee et al. [24] proposed a long short-term memory

(LSTM) architecture to reconstruct 3D depth from the centroid to edge joints through learning the joint inter-dependencies. Chen et al. [27] proposed improve 3D human pose estimation by synthesizing human images [28,29]. Hossain et al. [30] designed an LSTM-based sequence-to-sequence network to estimates a sequence of 3D poses from a sequence of 2D poses. Given fact that 2D-to-3D inference is an ill-posed problem, methods along this line are prone to ambiguities in the 2D-to-3D regression at the second stage of this pipeline, if no addition image evidences are utilized.

The major difference between pipeline (a) and (b) is that the latter learns addition image evidences, like depths on joints, to help the 3D inference. Pons-Moll et al. [31] proposed an extensive set of posebits representing the boolean geometric relationships between body parts, and designed an algorithm to select useful posebits for 3D pose inference. Nie et al. [22] used the 2D keypoints and the correpsonding local image patches to predict the depth of human joints. Zhou et al. [32] proposed to learn the 2D keypoint locations and the corresponding depth using a weakly supervised approach. Pavlakos et al. [33] predicted the depth of human joints using manually annotated ordinal depth supervision by a ranking loss. Wang et al. [34] defined the pose attributes as intermediate image cues to reduce the ambiguity in lifting 2D pose into 3D space. These methods requires that all the human keypoints are present in the image, which is unfortunately too strong for real-world application scenarios, where out-of-image absences of body joints frequently occur and thus collapse the 3D estimation results.

2.2. Part Affinity Fields Based Methods

The part affinity field is originally proposed by Cao et al. [3]. In their work, 2D PAFs were used to help linking the kepoints on a person in the multi-person 2D pose detection problem. After that, several early attempts have been made to use 3D PAFs for 3D pose estimation. Luo et al. [35] and Xiang et al. [36] followed Cao et al. 's idea to predict 2D keypoint heatmaps and 3D PAFs. In their method, the 3D orientations were extracted according to the predicted 2D keypoint locations. Unfortunately, this step is non-differentiable making end-to-end training infeasible. In Liu et al. 's work [37], 3D PAFs is used as additional image evidence to improve the 2D-to-3D regression. All these methods actually still rely heavily on 2D keypoint detections. As a result, these methods actually are still fragile to keypoints absences.

2.3. Our Approach

Our *pointless* 3D pose estimation method substitutes the detection of the only intermittently visible 2D points, with the estimation of 1D PAFs, i.e., the regions of limbs. This replacement not only makes our method robust to the cases with visually absent keypoints, but also provides us a simple and differentiable way to extract 3D vectors from PAFs, making end-to-end training possible. Lastly, we introduce an auxiliary task, the 2D PAFs estimation, which enables us to train the network on 2D pose dataset for better generalization ability. Compared to prior PAF-based methods, our method is end-to-end, robust to partial absence of body parts from the image, and achieves excellent generalization capacity to in-the-wild images.

3. Method

Our method takes as input a 256×256 color image and predicts three groups of part affinity fields simultaneously. Then the predicted 3D PAFs, which could be noisy, are refined by the 1D and 2D PAFs through a parameter-free process. The 3D limb orientations are obtained by averaging and unitizing each PAF. Then we injected the 3D orientations into a puppet with fixed limb lengths to obtain the final 3D pose output. The first step, predicting the PAFs, is the only step with learnable parameters in our method. All the remaining steps are differentiable, enabling us to end-to-end train the whole model. In the following sections, we explain these steps in detail.

3.1. Consistent 1D/2D/3D PAFs Representations

Part affinity field is designed to represent vectorised information such as directional vectors, in a specific region of interest. In this paper, the 1D PAF is taken to be a binary map that indicates

whether each pixel is in the region of interest or not. In human pose estimation, the exact human limb region is unavailable, so we use two adjacent keypoints and a fixed width d to define a rectangular region of a limb. When the Euclidean distance of two adjacent keypoint is smaller than $2d$, we define the region as a circle centerd at the mid-point of the two keypoints with a radius of $r = \sqrt{2}d$, to avoid a too small region. Figure 3 shows two example ROIs. Here a limb is defined as the body part that connects two adjacent keypoints.

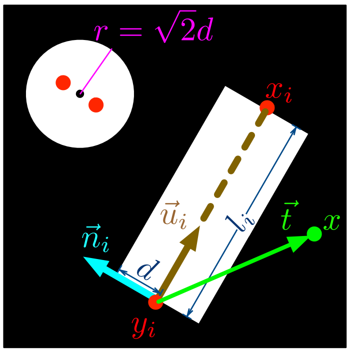


Figure 3. Regions of Interest (ROI) examples.

For 2D and 3D PAF, the region is defined the same as 1D PAF. The only difference is that in an N-D PAF, each pixel represents an N-D vector. In this paper, we use PAFs to represent 2D and 3D limb directions, so the vectors in PAFs are unitized.

In prior works that adopt PAFs [3,35–37], a two-branch architecture is used to learn two inconsistent sub-tasks, PAFs and 2D keypoint heatmaps estimation. This inconsistency leads to a a non-differentiable post-processing, so that end-to-end training is infeasible. However, the end-to-end training in 3D pose estimation is especially important in orientation based methods [19], because we need a long term objective to achieve an overall better pose estimation, otherwise the errors in each part will accumulate and result in large error to the far end body joints.

In this paper, we propose a three-branch architecture to learn three different but consistent sub-tasks, 1D, 2D and 3D PAFs estimation, respectively. This design not only simplifies the post-processing, but also make it differentiable, so that our method is end-to-end. In addition, in previous works, the networks have to learn two totally different representations, i.e., Gaussian kernel based keypoint heatmaps and limb region based PAFs. In our method, the three sub-tasks are more consistent than the previous keypoint based design, because they are all limb region based and orientation based. The consistency among the three branches makes it easier for the network to learn domain-independent features, which is crucial to improving the generalization capacity of the network.

3.2. Simultaneous 1D/2D/3D PAFs Learning

Figure 4 illustrates the core architecture of our FCNN. In stage $T = t$, the feature map F_t is first fed into an hourglass [2] block, then it flows through three branches simultaneously, to predict the 1D/2D/3D PAFs . After that, all the PAFs are transformed into features maps with the same number of channels to F_t by 1×1 convolutional layers. Then the feature maps, including F_t, are added up to generate a new one F_{t+1} as the input for the next stage. At training phase, each stage produces a group of PAFs and a 3D pose. At inference phase, only the output of the last stage are used as the final predictions.

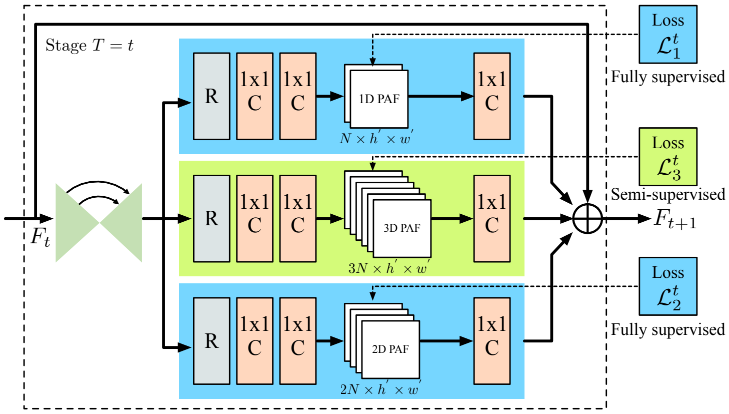


Figure 4. The architecture of our fully convolutional neural network (FCNN) for simultaneous 1D/2D/3D PAFs learning.

3.2.1. Semi-Supervised 3D PAFs Training Strategy

Following many previous works [18,21,24,32–34,38,39], we use a 2D pose dataset MPII [11] and a 3D pose dataset Human3.6M [12] to train the network. In training phase, each mini-batch is randomly sampled from the two datasets with equal probability.

The 1D and 2D branches are fully-supervised, because both datasets provide 2D annotations, from which we can generate the ground truth to fully supervise the learning of the 1D and 2D PAFs. For the 3D branch, on average only half of the training examples, that is, those come from the 3D dataset, have 3D PAFs supervision, so it is semi-supervised. We use the average gradient of the 3D training examples in each mini-batch to approximate the gradient of the whole mini-batch. When there are no 3D examples in a mini-batch, we simply set the gradients of parameters in this branch to zeros so that the weights in it are not updated in this single backward. In other words, the 3D branch *only* sees training examples from Human3.6M. Surprisingly, the automatically learnt features F_t, which is shared by the three branches, is domain-independent enough. As a result, despite the 3D branch gets supervisions obtained from a monitored indoor environment, the network generalizes pretty well to in-the-wild images.

It is worth noting that our method achieves a better generalization ability (See the results in Section 4.4), without applying any weakly supervised loss or GAN loss to the 2D training examples [32,33,38,39]. This greatly simplifies the training process and makes it easier to re-implement.

3.2.2. The Loss Functions

In this paper, the learning of 1D PAFs is taken to be a pixel-wise binary classification problem. The value at each pixel in the 1D PAF indicates the probability of this pixel lies inside the region of interest. In other words, the predicted 1D PAFs are also the limb region confidence maps, which can simplify the post-processing and make it differentiable. We use the *Binary Cross Entropy* (BCE) loss for 1D PAFs learning as follows:

$$\mathcal{L}_1^t = \sum_i \sum_n [q_n^i \log p_n^i + (1 - q_n^i) \log(1 - p_n^i)], \tag{1}$$

where p_n^i and q_n^i represents the predicted and ground truth probability at pixel n in training example i.

The 2D/3D PAF learning is taken to be a regression problem. As we mentioned above, the 2D/3D PAFs represent both limb regions and 2D/3D limb orientations. In orientation learning based 3D pose estimation, the major target is to learn the directions, rather than the regions, which means a PAF with correct orientation estimation but inaccurate region detection is absolutely acceptable. However, the *Mean Squared Error* (MSE) loss, used by previous PAF-related works, will give a large penalty to these acceptable prediction cases.

To this end, we propose a boundary-insensitive loss function for N-D PAF learning. The basic idea is that, we impose small weights to the pixels where the region prediction is incorrect, to reduce the impact of wrong region prediction. This is achieved by utilising the 1D PAF predictions, which represent region prediction confidence maps, to define a pixel-wise weight map as follows:

$$w(p_n^i, q_n^i) = 1 - (1 - w_0)e^{-\frac{[\ln(p_n^i + q_n^i) - \ln(2 - p_n^i - q_n^i)]^2}{2\sigma^2}}, \tag{2}$$

where w_0 is the lower bound of weights, σ is the standard deviation of the Gaussian distribution. We set $w_0 = 0.2$ in our experiments. The function above can be taken as a soft **ex**clusive **nor** (ex-NOR, or XNOR) function. That is, an output weight of 1 is obtained only if both of its inputs are at the same probability level, either high or low. Otherwise, if they are at different probability level, the weight approaches to the lower bound w_0. The ground truth probability q_n^i is binarized so that the case that prediction is correct but weight is small can never happen. Then the proposed boundary-insensitive loss is an MSE loss masked with the weight we define above:

$$\mathcal{L}_N^t = \sum_i \sum_n w(p_n^i, q_n^i) \|\mathbf{x}_n^i - \mathbf{y}_n^i\|_2^2, \tag{3}$$

where $N = 2, 3$, $\mathbf{x}_n^i$ and $\mathbf{y}_n^i$ represents the predicted and ground truth N-D directions at pixel n in training example i. The proposed loss function gives a small penalty to a pixel if its location prediction is wrong, no matter its direction prediction is correct or not, while it still gives a large penalty to a pixel if its location prediction is correct but the direction prediction is wrong. The lower bound w_0 can avoid the loss function $\mathcal{L}_N^t$ being trivially minimized by taking $p = 1 - q$. We want to make it clear that, the proposed boundary-insensitive loss does not boost the 3D pose prediction accuracy, but it could suppress the oscillation in training and thus speed up the convergence.

3.3. Differentiable Post-Processing

In this section, we introduce how we extract the 3D directions from the noisy PAF predictions in a differentiable way. Our post-processing consists of two parameter-free steps, 3D PAF refinement and 3D orientation injection.

3.3.1. 3D PAF Refinement

The 3D PAF refinement is further divided into two steps—denoising and 2D/3D PAF ensemble. As shown in Figure 2b, the predicted 1D and 2D PAFs are usually nice and clean, but the 3D PAFs can be very noisy, typically in pixels beyond the body region. This is what expected, since the 3D branch has never been trained with any in-the-wild images with 3D supervisions. We use the much cleaner 1D PAF predictions, to filter out the noise in the 3D PAFs by a masking operation. This is the one of our motivations to design the 1D PAF branch.

As we mentioned in Section 3.1, the directional vector represented by first two dimensions of the 3D PAFs are approximately parallel with that of the 2D PAFs, which can be used to further refine the 3D PAFs. Specifically, we resize each 2D limb directional vector so that its L2 norm equals that of the first two dimensions of the corresponding 3D PAFs. Then we replace the first two dimensions of the 3D directional vector with the average of the two 2D vectors. The rationale behind this is that we take the 2D PAFs and the first two dimensions of the 3D PAFs as predictions produced by two separate models. The final prediction is the ensemble of them, which often improves the prediction by voting.

3.3.2. 3D Orientation Injection

As discussed in Section 1, our method uses a skeleton-shaped puppet for producing the 3D pose output. The process of combining the predicted 3D directional vectors and the puppet makes a real 3D pose. We term this process as 3D orientation injection.

The human model in this paper is an articulated object that consists of several limbs and joints. A limb is a segment of fixed length, and a joint is the end point of a limb. Limbs can rotate among a conjunct joint (See Figure 2d). In this way, the human skeleton forms a tree structure. Usually the root of the tree is taken to be the pelvis, and is fixed at the origin. Following Human3.6M [12], the human model consists of 17 joints with 16 limbs. A limb has 0 degree of freedom (DOF) as its length is fixed, and a joint has 2 DOFs except for the root. So that there are 32 DOFs in our skeleton-shaped human model. Unlike Zhou et al. [40] that use a CNN to predict the rotation angle of joints, we estimate the 3D orientation directly. From the view of limb orientations, each limb 3D direction vector has 2 DOFs, which sums up to 32 DOFs as well.

3D orientation injection is a step that combines the predicted 3D limb orientations with the human model to generate the final 3D pose estimation. This process works like twisting the limbs of the human model to fit the predicted 3D direction vectors. For an arbitrary child node k in the skeleton tree, its 3D location prediction Y_k is determined by its parent node's 3D location prediction X_k, the predicted orientation $\mathbf{v}_k$, and the limb length L_k, in a recursive way as follows:

$$Y_k = X_k + L_k \mathbf{v}_k, \tag{4}$$

where we have $X_0 = \mathbf{0}$. L_ks are constant numbers, which are obtained by calculating the average of the limb lengths of subject S1, S6, S6, S7 and S8 in the training set of Human3.6M. This process stops when the locations of all the leaf nodes are determined.

During 3D orientation injection, the orientation errors do not accumulate, because the 3D direction vectors remain unchanged in this process. This is shown in Equation (4), where the 3D direction vector $\mathbf{v}_k$ is only scaled by a factor of L_k and translated by X_k, both of which do not alter the direction of the vector.

3.4. End-to-End training with 3D Pose Loss

In Section 3.2, we discussed that the 3D PAFs are learnt by minimizing the loss function in Equation (3). In this loss function, each 3D limb orientation is *independently* estimated. Although limb orientation errors do not accumulate, joint location errors still could propagate along the skeleton tree and possibly accumulate into large errors for joints at the leaf node. For example, a location shift in the left shoulder would lead to the same amount of location shift to both left elbow and left wrist.

To solve this problem, long-term objectives should be considered so that the 3D orientations are jointly optimized. In our method, since all the steps are designed to be differentiable, we can directly use the 3D pose loss as a long-term objective and train the model end-to-end. Here we use L1 loss for 3D pose:

$$\mathcal{L}_{\text{pose}} = \sum_i \sum_k |Y_k^i - \hat{Y}_k^i|, \tag{5}$$

where Y_k^i and $\hat{Y}_k^i$ represents the predicted and GT 3D locations for joint k in training example i.' In experiment, we find that end-to-end training can speed up the convergence, and improve the accuracy of estimation as well. In all, for a T stage model, the overall loss function is:

$$\mathcal{L} = \sum_{t=1}^{T} (\lambda_1 \mathcal{L}_1^t + \lambda_2 \mathcal{L}_2^t + \lambda_3 \mathcal{L}_3^t + \mathcal{L}_{\text{pose}}^t), \tag{6}$$

where λ_1, λ_2 and λ_3 control the relative importance of each objective. We set $\lambda_1 = 0.1$, $\lambda_2 = 1$ and $\lambda_3 = 1$ in our experiments.

4. Experiments

We provide here details on our experiments, including datasets and protocols used, training details, quantitative and qualitative results, robust analysis and ablation studies.

4.1. Datasets and Protocols

We evaluate our method on the following three popular human pose benchmarks.

Human3.6M [12] is a large-scale indoor 3D human pose dataset that comprises 3.6 million images and the corresponding 2D pose and 3D pose annotations. It features 7 subjects performing 15 everyday activities. We follow the standard protocol on Human3.6M to use S1, S5, S6, S7 and S8 for training, and use S9 and S11 for evaluation. Following [4,13,32,33], we down-sampled the original videos from 50 fps to 10 fps to remove redundancy in both training and evaluation.We report qualitative results on this dataset in terms of three evaluation metrics, i.e., the mean per joint position error (MPJPE), MPJPE after Procrustes alignment with the ground truth (PA-MPJPE), and the mean per limb orientation error (MPLORE), a metric for evaluating the 3D orientation prediction error as follows:

$$\mathrm{MPLORE} = \frac{1}{L}\sum_{l}\arccos(\frac{\mathbf{x}_l}{|\mathbf{x}_l|}\cdot\mathbf{y}_l), \tag{7}$$

where $\mathbf{x}_l$ and $\mathbf{y}_l$ are the predicted and GT direction vector of a limb l, respectively. L is the number of limbs.

MPII [11] is the most widely used benchmark for 2D human pose estimation. It contains 25K in-the-wild images with 2D annotations but no 3D ground truth. As a result, direct image-to-3D training is not a practical option with this dataset. We adopt this dataset for the learning of 1D and 2D PAFs, and also use it for the qualitative evaluation of our 3D pose estimation.

MPI-INF-3DHP [41] is a smaller 3D pose dataset constructed by the Mocap system with both constrained indoor scenes and complex outdoor scenes. We only use the test split of this dataset to evaluate the generalization capacity of our method quantitatively, as done in many prior works.

4.2. Implementation Details

The training of our network is handy and stable. We use a pre-trained *Stacked Hourglass* [2] model to initialize the common modules of our network and the stacked hourglass, including the first 7 × 7 convolutional layer, the following 3 residual blocks, and the hourglass sub-modules (see Figure 4). Then the network is trained for 40 epochs with RMSprop. The initial learning rate is 5×10^{-4} and decayed by 0.25 at the epoch of 20 and 30, respectively. The training examples are randomly sampled from Human3.6M and MPII with equal probability. Augmentations of random scale (1 ± 0.25) and random color jitter (1 ± 0.2), random rotation ($\pm 30°$, $p = 0.6$) and random horizontal flipping ($p = 0.5$) are used for both datasets. For fair comparison, we do not use multiple crops or flipping test for possible better performance score. The whole training procedure takes about 20 h on a single Tesla V100 GPU. The inference speed is about 70fps with a batch size of 6 on the same architecture.

4.3. Quantitative Results on Human3.6M

We first evaluate our method on Human3.6M using the metric MPJPE and PA-MPJPE in order to compare our method with state-of-the-art methods. The results are shown in Table 1. Our method slightly outperforms the state-of-the-art methods in terms of the average MPJPE and PA-MPJPE over all the 15 activities, even though our method is not designed in a way that optimizes these metrics.

Table 1. Detailed results on Human3.6M under the metrics of MPJPE and PA-MPJPE . Methods marked with * use ground truth camera parameters in post-processing. The results of all approaches are taken from the original papers, except for [5], which is taken from [42]. We also provide the results evaluated with ground truth limb lengths. Best results are marked in bold.

MPJPE	**Direct.**	**Discuss**	**Eat**	**Greet**	**Phone**	**Photo**	**Pose**	**Purch.**	**Sit**	**SitD.**	**Smoke**	**Wait**	**WalkD.**	**Walk**	**WalkT**	**Avg.**
Tekin et al. [43]	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6	70.1	107.3	69.3	70.3	74.3	51.8	74.3	69.7
Zhou et al. [32]	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.2	66.1	51.4	63.2	55.3	64.9
Martinez et al. [18]	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. [19]	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Fang et al. [23]	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. [38]	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
Pavlakos et al. [33]	48.5	54.4	54.4	**52.0**	59.4	65.3	49.9	52.9	65.8	**71.1**	56.6	52.9	60.9	44.7	47.8	56.2
Lee et al. [24]	**43.8**	**51.7**	48.8	53.1	**52.2**	74.9	52.7	**44.6**	**56.9**	74.3	56.7	66.4	68.4	47.5	45.6	55.8
Dabral et al. [39]	46.9	53.8	**47.0**	52.8	56.9	63.6	**45.2**	48.2	68.0	94.0	**55.7**	51.6	55.4	**40.3**	**44.3**	55.5
Chen et al. [42]	45.9	53.5	50.1	53.2	61.5	72.8	50.7	49.4	68.4	82.1	58.6	53.9	57.6	41.1	46.0	56.9
Sun et al. [5]*	46.5	48.1	49.9	51.1	47.3	43.2	45.9	57.0	77.6	47.9	54.9	46.9	37.1	49.8	41.2	49.8
Chen et al. [42]*	41.1	44.2	44.9	45.9	46.5	39.3	41.6	54.8	73.2	46.2	48.7	42.1	35.8	46.6	38.5	46.3
Ours ($T = 2$)	51.2	56.5	54.0	57.1	59.4	63.3	51.1	53.3	65.2	74.5	57.4	54.6	59.8	52.7	47.9	57.2
Ours ($T = 4$)	48.6	54.5	53.1	55.0	57.2	**60.8**	47.9	53.0	64.2	74.9	56.8	**51.1**	56.4	49.1	45.2	**55.2**
Ours (GT Length)	43.6	50.3	50.2	50.7	54.1	58.8	43.4	49.5	61.8	72.9	54.2	47.5	53.9	45.3	41.9	51.9
PA-MPJPE	**Direct.**	**Discuss**	**Eat**	**Greet**	**Phone**	**Photo**	**Pose**	**Purch.**	**Sit**	**SitD.**	**Smoke**	**Wait**	**WalkD.**	**Walk**	**WalkT**	**Avg.**
Moreno-Noguer [21]	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
Martinez et al. [18]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	59.4	49.2	45.0	49.5	38.0	43.1	47.7
Fang et al. [23]	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
Pavlakos et al. [33]	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	**50.7**	**56.8**	42.6	39.6	43.9	32.1	36.5	41.8
Lee et al. [24]	38.0	39.1	46.3	44.4	49.0	55.1	40.2	41.1	53.2	68.9	51.0	39.1	56.4	33.9	38.5	46.2
Dabral et al. [39]	**32.8**	**36.8**	42.5	**38.5**	**42.4**	49.0	35.4	**34.3**	53.6	66.2	46.5	**34.1**	42.3	**30.0**	39.7	42.2
Ours ($T = 2$)	37.3	40.3	39.9	41.2	43.4	43.7	37.3	38.7	50.7	56.9	42.9	37.9	43.7	38.7	35.1	41.8
Ours ($T = 4$)	36.2	39.4	**39.1**	40.1	43.1	**43.6**	**35.1**	38.6	**50.7**	57.2	**43.5**	36.5	41.8	36.3	**33.9**	**40.9**
Ours (GT Length)	33.7	37.5	38.0	37.7	42.0	42.4	32.7	37.1	50.4	56.4	42.5	34.6	40.6	33.7	31.1	39.4

Since objects with totally different sizes could project into similar 2D images, the absolute length estimation from a single color image is usually not reliable. To this end, we propose the *mean per limb orientation error* (MPLORE), defined in Equation (7), to evaluate the 3D pose estimation performance in the setting of limb lengths decoupled.

The MPLORE results are shown in Table 2. We compare our method with three state-of-the-art methods [18,32,34]. Our method achieves the best results on 14 of the 15 activities, and the best averaged result. The MPLORE results indicate that our method predicts much better 3D orientations of limbs.

Table 2. Comparison with state of the arts on Human3.6M in terms of MPLORE (lower the better). The best score is marked in bold. We achieve the best results across all the activities except for *Walking*, which is only slightly worse than [34].

MPLORE (°)	**Direct.**	**Discuss**	**Eat**	**Greet**	**Phone**	**Photo**	**Pose**	**Purch.**	**Sit**	**SitD.**	**Smoke**	**Wait**	**WalkD.**	**Walk**	**WalkT**	**Avg.**
Zhou et al. [32]	11.72	13.19	11.56	12.73	13.99	14.45	11.36	12.05	16.05	15.86	13.89	12.21	13.44	10.71	10.63	12.92
Martinez et al. [18]	8.73	9.29	8.96	9.20	10.85	11.86	9.27	8.70	12.10	15.14	10.20	9.98	10.82	8.10	8.97	10.14
Wang et al. [34]	8.64	8.94	8.88	9.08	10.38	10.90	9.01	8.31	11.43	11.84	10.04	9.28	10.14	**7.72**	8.74	9.56
Ours	**8.27**	**8.84**	**8.65**	**8.68**	**9.69**	**9.98**	**8.13**	**8.03**	**10.66**	**10.34**	**9.44**	**8.59**	**9.63**	7.76	**8.33**	**9.00**

4.4. Quantitative Results on MPI-INF-3DHP

This dataset is collected in indoor and outdoor with a multi-camera marker-less MoCap system. Because of this, the ground truth 3D annotations have some noise. To quantitatively show the generalization capacity of our method, we evaluate the 3D extension **P**ercentage of **C**orrect **K**eypoints (3DPCK) and **A**rea **U**nder **C**urve (AUC) score on the MPI-INF-3DHP without training with this dataset, as done in many previous works. The results are shown in Table 3. Our method achieves the second best score in terms of

3DPCK, and the best score in terms of AUC, demonstrating its good generalization capacity to unseen testing images.

Table 3. 3DPCK and AUC on the MPI-INF-3DHP dataset. Higher is better. The results for all approaches are taken from the original papers. ° represents our method without the auxiliary 2D orientation task, and *r*30 and *r*90 represent using random rotation augmentation of 30 and 90 degrees in the training. No training data from this dataset have been used. Our method achieves the best score in terms of AUC, and second best score in terms of 3DPCK.

	[41]	[32]	[33]	[38]	[44]	[42]	[35]	[45]	Ours °	Ours$_{r30}$	Ours$_{r90}$
3DPCK	64.7	69.2	71.9	69.0	69.6	68.7	64.6	67.9	69.4	70.5	71.1
AUC	31.7	32.5	35.3	32.0	35.5	34.6	32.1	-	37.3	37.4	38.3

Since our method learns the 3D limb orientations, of which the first two dimensions represent the 2D orientations, using large-angle random rotation augmentation on the image should help training a network with better generalization capacity. This is in fact validated by the experiment results in the last two columns in Table 3, in which the model trained with 90-degree random rotation augmentation has considerable improvement, compared to the one trained with 30-degree augmentation.

4.5. Qualitative Results on MPII

MPII is the most widely used 2D pose datasets which does not contain 3D annotations. In this section, we provide some qualitative results in Figure 5 on this dataset especially in some challenging scenes like images with missing body parts. In Figure 5, the images in the first two rows are truncated and the occluded in the last two rows. Our method can produce visually appealing results even in the presence of incomplete body parts, proving the robustness of the proposed method. These examples on MPII also demonstrate our method's generalization capacity on various in-the-wild images.

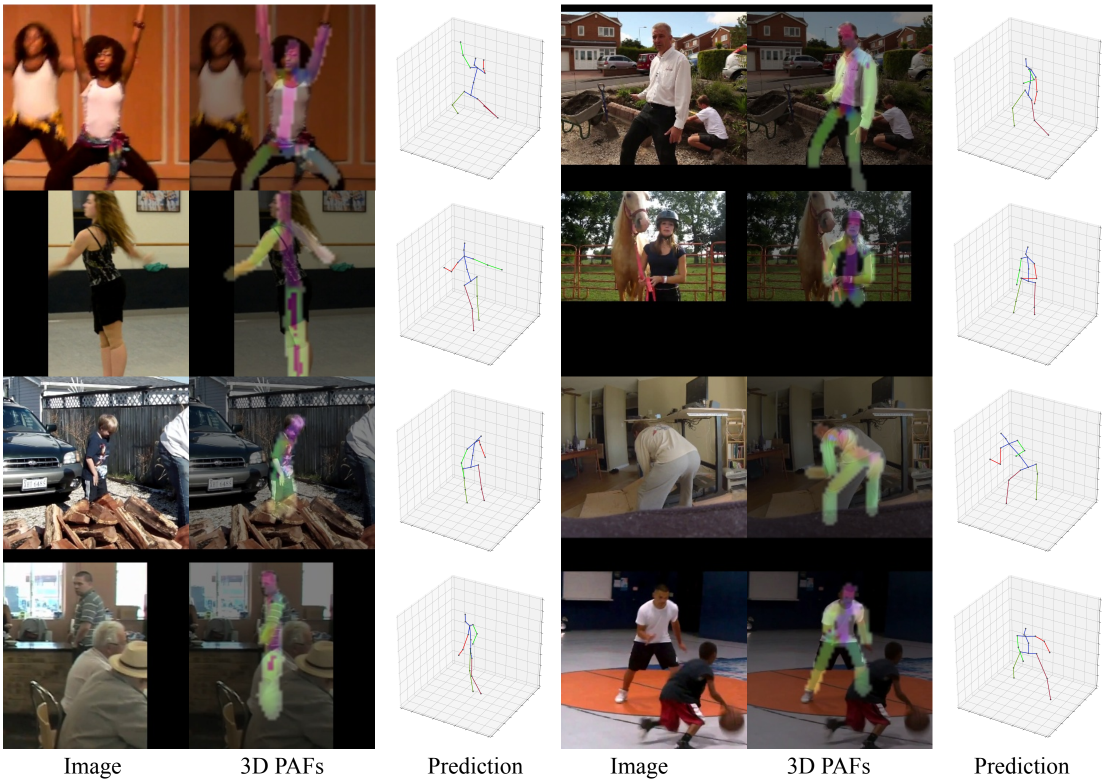
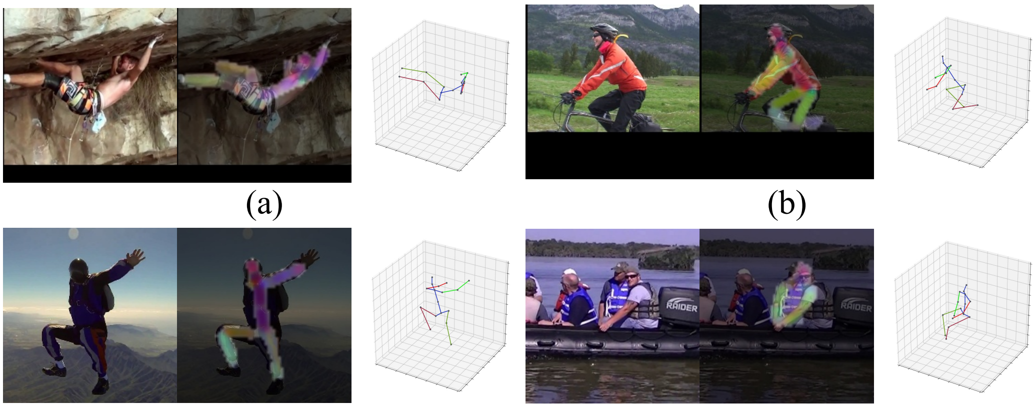


Figure 5. Qualitative results on truncated and occluded images from MPII. Best viewed in color and zooming in.

In Figure 6, we show four failure examples. In Figure 6a, the pose is rarely-seen as well as includes self-occlusions. In Figure 6b, the left lower leg is out-of-image and there is a bag next to it with the similar color, so the network takes the bag as the absent lower leg. In Figure 6c, the subject in it wears a black helmet that covers his/her face, in which case the network gets confused by the left and right side of the body. In Figure 6d, the subject is occluded by the another person, in which case the network takes the arm of the person in the front as the subject's in the back.


(a) (b)

(c) (d)

Figure 6. Failure cases.

4.6. Robustness Analysis: A Case Study

In Figure 7, we show the qualitative and quantitative results of a testing image under 5 synthetic disturbances including edge-erasing, rectangle-erasing, circle-erasing, partial-blurring and a composition of the above. To explain how the performance deteriorates, we also visualize the predicted 2D keypoint heatmaps and the 3D PAFs. Here only the four limbs are included for better visualization. In this case, our method is much less sensitive to these disturbances than those 2D keypoint detection based methods. The synthetic disturbances are generated at random. Our method achieves consistent better performance, which indicates that our method has the potential in improving the robustness of 3D human pose estimation.

The robustness of our method can be attributed to two aspects. First and foremost, the *pointless* method design enables us to predict the ROI and 3D orientation of a limb even when the limb is partially out-of-image or occluded. Second, the final 3D orientation of a limb is extracted by averaging all the predicted vectors in the ROI. The averaging operation in this step can be treated as an average filter, which suppresses noises and disturbances in the predicted vectors, making the prediction more stable.

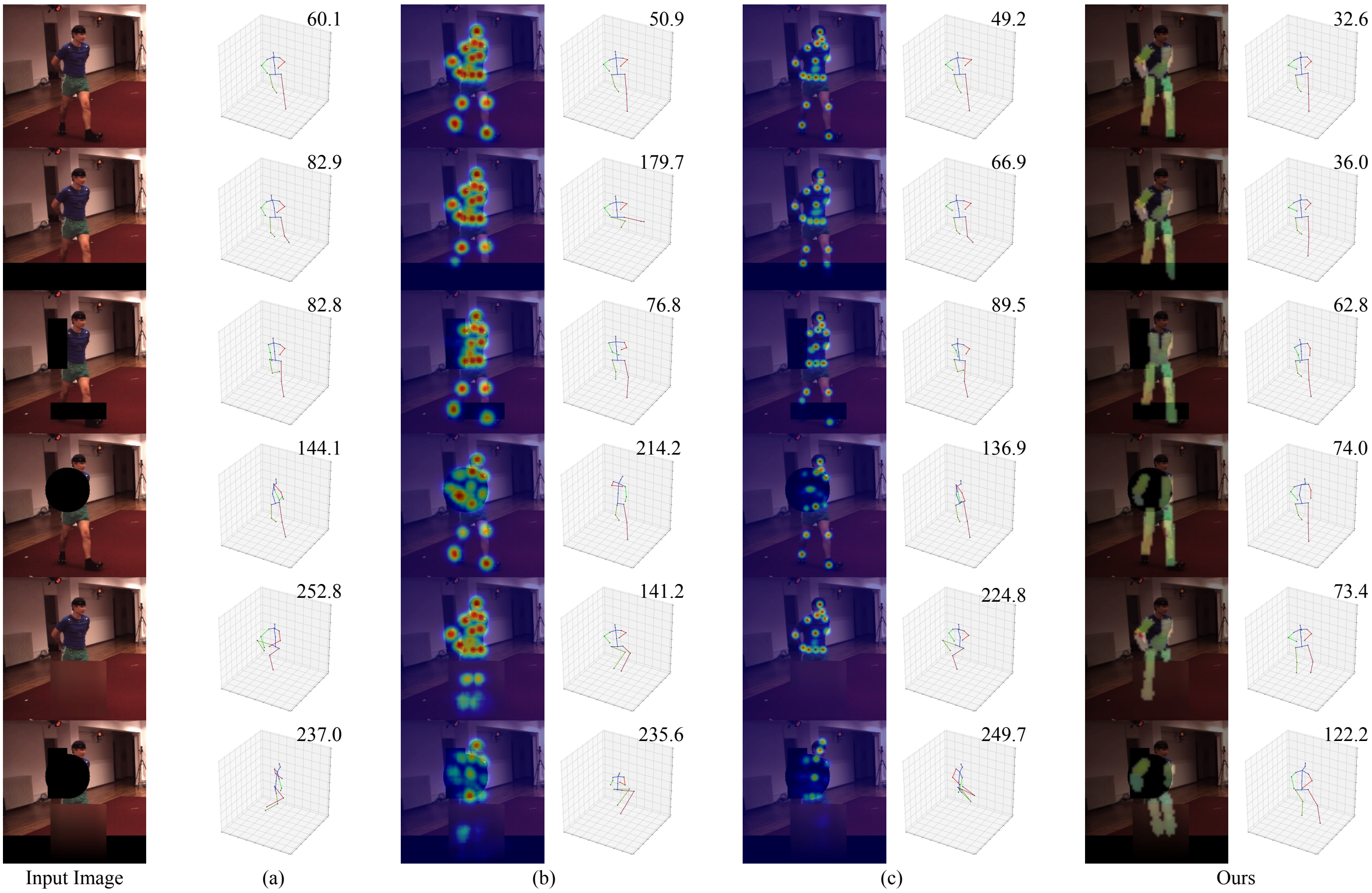


Figure 7. A case study on images with various geometric occlusions. We compare the results with three state-of-the-art methods: (**a**) [18], (**b**) [32], and (**c**) [34]. Our method is robust under missing key points, rectangular and circle-occlusions, as well as partial-blurring.

4.7. Ablation Study

To analyze the effectiveness of different steps in our method, we conduct ablation study on Human3.6M in terms of MPJPE. The results are reported in Table 4. *Baseline* refers to the approach that uses the original 3D PAFs without refinement. *Denoising* refers to using 1D/2D PAFs to remove the noise in the predicted 3D PAFs. *Flip* refers to using horizontal flipping test.

Table 4. Ablation study on Human3.6M in terms of MPJPE.

Methods	**MPJPE**
Baseline	59.3
Baseline + Denoising	58.9
Baseline + Denoising + $\mathcal{L}_{\text{pose}}$	56.7
Baseline + Denoising + $\mathcal{L}_{\text{pose}}$ + Flip	55.2

The performance gain by denoising might seem minor on Human3.6M. The reason is that the 3D branch is trained in a fully supervised manner on Human3.6M so that there is little noise in the predicted 3D PAFs on images from this dataset. However, when testing on in-the-wild images, there could be a lot noise in 3D PAF predictions (see Figure 2), making the denoising an indispensable step.

5. Conclusions

We propose in this paper a simple and effectual 3D human pose estimation method, termed *pointless* 3D pose estimation. Unlike prior methods that rely on 2D keypoint detection, which is prone to errors in the absence of body parts and joints, the proposed approach bypasses this stage and substitutes it with estimations that explicitly account for both the ROIs and 3D orientations. This allows us to robustly recover the poses, by taking advantage of the estimated 3D vector pointing from a neighboring body part, even when some 2D keypoints are out of scene or occluded. State-of-the-art results, in terms of both keypoint-based and angle-based evaluation metrics, have been achieved on standard benchmarks as well as in-the-wild data. Possible future work includes multi-person 3D pose estimation, person limb length estimation and so on.

Author Contributions: Conceptualization, J.W.; methodology, J.W.; software, J.W.; validation, J.W.and Z.L.; formal analysis, J.W.; investigation, J.W.; resources, J.W.; data curation, J.W.; writing–original draft preparation, J.W.; writing–review and editing, Z.L.; visualization, J.W.; supervision, Z.L.; project administration, Z.L.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly archived datasets used in the study are listed below. Human3.6M: http://vision.imar.ro/human3.6m/description.php; MPII: http://human-pose.mpi-inf.mpg.de/; MPI-INF-3DHP: http://gvv.mpi-inf.mpg.de/3dhp-dataset/.

Acknowledgments: We thank the anonymous reviewers whose comments helped improve the quality of this paper.

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

References

1. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional pose machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
2. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
3. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
4. Pavlakos, G.; Zhou, X.; Derpanis, K.G.; Daniilidis, K. Coarse-to-fine volumetric prediction for single-image 3D human pose. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
5. Sun, X.; Xiao, B.; Wei, F.; Liang, S.; Wei, Y. Integral Human Pose Regression. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
6. 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 Conference on Computer Vision and Pattern Recognition, 18–22 June 2018.
7. Sárándi, I.; Linder, T.; Arras, K.O.; Leibe, B. How robust is 3D human pose estimation to occlusion? *arXiv* **2018**, arXiv:1808.09316.
8. Sárándi, I.; Linder, T.; Arras, K.O.; Leibe, B. Synthetic occlusion augmentation with volumetric heatmaps for the 2018 eccv posetrack challenge on 3d human pose estimation. *arXiv* **2018**, arXiv:1809.04987.
9. Chen, X.; Lin, K.Y.; Liu, W.; Qian, C.; Lin, L. Weakly-Supervised Discovery of Geometry-Aware Representation for 3D Human Pose Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–21 June 2019.
10. Qiu, H.; Wang, C.; Wang, J.; Wang, N.; Zeng, W. Cross View Fusion for 3D Human Pose Estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
11. Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B. 2d human pose estimation: New benchmark and state of the art analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
12. 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]
13. Zhou, X.; Zhu, M.; Leonardos, S.; Derpanis, K.G.; Daniilidis, K. Sparseness meets deepness: 3D human pose estimation from monocular video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
14. Zhou, X.; Zhu, M.; Leonardos, S.; Daniilidis, K. Sparse representation for 3D shape estimation: A convex relaxation approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1648–1661. [CrossRef] [PubMed]

15. Zhou, X.; Zhu, M.; Pavlakos, G.; Leonardos, S.; Derpanis, K.G.; Daniilidis, K. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 901–914. [CrossRef] [PubMed]
16. Chen, C.H.; Ramanan, D. 3D Human Pose Estimation = 2D Pose Estimation + Matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
17. Bogo, F.; Kanazawa, A.; Lassner, C.; Gehler, P.; Romero, J.; Black, M.J. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
18. Martinez, J.; Hossain, R.; Romero, J.; Little, J.J. A Simple yet Effective Baseline for 3D Human Pose Estimation. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
19. 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.
20. Tome, D.; Russell, C.; Agapito, L. Lifting From the Deep: Convolutional 3D Pose Estimation From a Single Image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
21. Moreno-Noguer, F. 3D Human Pose Estimation from a Single Image via Distance Matrix Regression. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
22. Nie, B.X.; Wei, P.; Zhu, S.C. Monocular 3D human pose estimation by predicting depth on joints. In Proceedings of the International Conference on Computer Vision, 22–29 October 2017.
23. 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.
24. Lee, K.; Lee, I.; Lee, S. Propagating LSTM: 3D Pose Estimation based on Joint Interdependency. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
25. 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, 15–21 June 2019.
26. Arnab, A.; Doersch, C.; Zisserman, A. Exploiting Temporal Context for 3D Human Pose Estimation in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–21 June 2019.
27. Chen, W.; Wang, H.; Li, Y.; Su, H.; Wang, Z.; Tu, C.; Lischinski, D.; Cohen-Or, D.; Chen, B. Synthesizing Training Images for Boosting Human 3D Pose Estimation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 479–488. [CrossRef]
28. Bagiwa, M.A.; Wahab, A.W.A.; Idris, M.Y.I.; Khan, S.; Choo, K.K.R. Chroma key background detection for digital video using statistical correlation of blurring artifact. *Digit. Investig.* **2016**, *19*, 29–43. [CrossRef]
29. Aminu, M.; Wahid, A.; Idris, M.; Khan, S. Digital Video Inpainting Detection Using Correlation Of Hessian Matrix. *Malays. J. Comput. Sci.* **2016**, *29*, 179–195. [CrossRef]
30. Hossain, M.R.I.; Little, J.J. Exploiting temporal information for 3D human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
31. Pons-Moll, G.; Fleet, D.J.; Rosenhahn, B. Posebits for monocular human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2337–2344.
32. 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 International Conference on Computer Vision, 22–29 October 2017.
33. Pavlakos, G.; Zhou, X.; Daniilidis, K. Ordinal Depth Supervision for 3D Human Pose Estimation. In Proceedings of the Conference on Computer Vision and Pattern Recognition, 18–22 June 2018.
34. Wang, J.; Huang, S.; Wang, X.; Tao, D. Not All Parts Are Created Equal: 3D Pose Estimation by Modeling Bi-Directional Dependencies of Body Parts. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
35. Luo, C.; Chu, X.; Yuille, A. Orinet: A fully convolutional network for 3d human pose estimation. *arXiv* **2018**, arXiv:1811.04989.
36. Xiang, D.; Joo, H.; Sheikh, Y. Monocular Total Capture: Posing Face, Body, and Hands in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–21 June 2019.
37. Liu, D.; Zhao, Z.; Wang, X.; Hu, Y.; Zhang, L.; Huang, T. Improving 3D Human Pose Estimation Via 3D Part Affinity Fields. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 1004–1013.
38. 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 Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
39. Dabral, R.; Mundhada, A.; Kusupati, U.; Afaque, S.; Sharma, A.; Jain, A. Learning 3D Human Pose from Structure and Motion. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
40. Zhou, X.; Sun, X.; Zhang, W.; Liang, S.; Wei, Y. Deep kinematic pose regression. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
41. Mehta, D.; Rhodin, H.; Casas, D.; Fua, P.; Sotnychenko, O.; Xu, W.; Theobalt, C. Monocular 3d human pose estimation in the wild using improved cnn supervision. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 506–516.

42. Chen, X.; Lin, K.Y.; Liu, W.; Qian, C.; Lin, L. Weakly-Supervised Discovery of Geometry-Aware Representation for 3D Human Pose Estimation. *arXiv* **2019**, arXiv:1903.08839.
43. Tekin, B.; Marquez Neila, P.; Salzmann, M.; Fua, P. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
44. Habibie, I.; Xu, W.; Mehta, D.; Pons-Moll, G.; Theobalt, C. In the Wild Human Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations. *arXiv* **2019**, arXiv:1904.03289.
45. Li, C.; Lee, G.H. Generating Multiple Hypotheses for 3D Human Pose Estimation with Mixture Density Network. *arXiv* **2019**, arXiv:1904.05547.