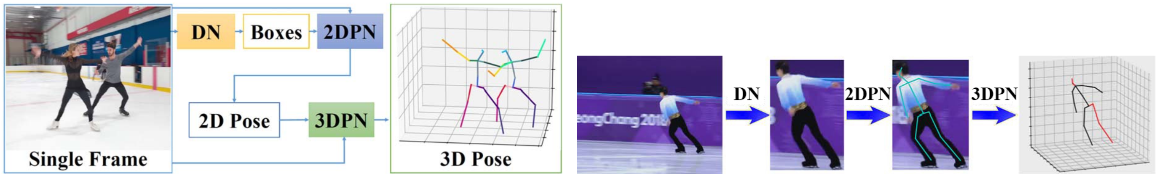
2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)

A practical framework of multi-person 3D human pose estimation with a single RGB camera

Le Ma[1,4],Sen Lian[1,4], Shandong Wang[3], Weiliang Meng[2,5,4,1][*], Jun Xiao[1][†], Xiaopeng Zhang[4,2,1] [‡]

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences,
[2]Zhejiang Lab,[3]Intel Labs China, [4]NLPR, Institute of Automation, Chinese Academy of Sciences
[5]The State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences



(a) The pipeline of our framework.

(b) 3D pose estimation process for one frame.

Figure 1: Our ’DN-2DPN-3DPN’ framework.DN:DetectNet,2DPN:2DPoseNet, 3DPN: 3DPoseNet.

Abstract

We propose a practical framework named ’DN-2DPN-3DPN’ for multi-person 3D pose estimation with a single RGB camera. Our framework performs three-stages tasks on the input video: our DetectNet(DN) firstly detects the people’s bounding box individually for each frame of the video, while our 2DPoseNet(2DPN) estimates the 2D poses for each person in the second stage, and our 3DPoseNet(3DPN) is finally applied to obtain the 3D poses of the people. Experiments validate that our method can achieve state-of-the-art performance for multi-person 3D human pose estimation on the Human3.6M dataset.

Keywords: Skeleton, Neural network, Detection, Real-time

Index Terms: Computing methodologies—Computer graphics—Animation—Motion processing; Computing methodologies—Artificial intelligence—Conputer vision tasks—Activity recognition and understanding

1 Introduction

In this paper, we mainly focus on the real-time multi-person 3D pose estimation from a monocular camera in various scenarios. This is a challenging topic as the 3D information can be ambiguous under this condition with more information loss because of the occlusion among multiple people.

We propose a practical framework named ‘DN-2DPN-3DPN’ for the real-time multi-person 3D pose estimation, consisting of DetectNet(DN), 2DPoseNet (2DPN),and 3DPoseNet (3DPN) as shown in Figure 1. Our DN detects the bounding-box of each person in each frame from the video, and outputs all the bounding-boxes to our 2DPN in order to predict the 2D pose for each person respectively based on the original image. Each of our 2D poses contains 17 keypoints following the Human3.6M dataset [2], and these coordinates are further input to our 3DPN for 3D pose keypoints prediction. We also design a simple strategy which can effectively conquer the occlusion issue, making the 3D pose estimation more effective and accurate, and reduce the computational complexity of the pose estimation. Besides, our framework can estimate the poses of small targets, and be superior to existing methods validated by our experiments.

2 Our Methods

2.1 The network

Our ‘DN-2DPN-3DPN’ framework consists of DetectNet (DN), 2DPoseNet (2DPN), and 3DPoseNet (3DPN).

DetectNet. In order to deal with multi-person simultaneously, the DetectNet(DN) should be used to obtain all the axis-aligned bounding boxes for each person individually in the frame and must be fast and accurate. We evaluated different advanced detection methods, and extract the detection part of CenterNet [10] as our DN, and we use the deep residual architecture networks ResNet101 and ResNet50 respectively as the pre-trained models, making human detection fast and accurate.

2DPoseNet. Our 2DPoseNet generates accurate 2D keypoints in high-speed base on our DN results. We have tested different 2D pose recognition methods and found that using HRNet [1] as our 2DPoseNet is better. HRNet can recognize 2D Pose directly on the image, but it often fails when the image has lots of irrelevant targets. Based on the results of DN, these interferences are removed, and the 2D pose recognition is more reliable for the small target.

3DPoseNet. Our 3DPoseNet(3DPN) predicts the 3D coordinates of each keypoint based on the 2D coordinates. Our 3DPN is a fully convolutional architecture with residual connections that take a sequence of 2D poses as the input, and output 3D poses referred to the network the structure proposed by Martinez et al. [4]. We modify the first layer of [5] for adapting to the input and design a new loss function L_{set} to train our 3DPN. Let $P_i = (X_i, Y_i, Z_i)$ be the estimated 3D coordinate keypoint by our 3DPN, and the corresponding ground-truth keypoint is P_i^g, we define the 3D joint MSE loss as Eq. (1) and the symmetric constraint as Eq. (2):

$$L_{mes} = \sum ||P_i - P_i^g||_2^2 \tag{1}$$

$$L_{sym} = \sum_{(i,j)\in E} (||P_i - P_j||_2^2 - ||P_i' - P_j'||_2^2) \tag{2}$$

where E is the set of all adjacent keypoints pairs, while P_i' and P_j' represent the keypoint and its symmetrical part respectively. Our

[*]Corresponding Author: weiliang.meng@ia.ac.cn
[†]Corresponding Author: xiaojun@ucas.ac.cn
[‡]Corresponding Author: xiaopeng.zhang@ia.ac.cn

978-0-7381-1367-8/21/$31.00 ©2021 IEEE
DOI 10.1109/VRW52623.2021.00092

Table 1: The comparison base on the Mean Per Joint Position Error(MPJPE)(taking the input 2D keypoints as the ground truth)/the Procrustes analysis MPJPE (P_MPJPE) on Human3.6M dataset. '-' means no value reported in the work, and our method obtains the best results.

Methods	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Martinez et al. [4]	37.7/39.5	44.4/43.2	40.3/46.4	42.1/47	48.2/51	54.9/56	44.4/41.4	42.1/40.6	54.6/56.5	58.0/69.4	45.1/49.2	46.4/45	47.6/49.5	36.4/38	40.4/43.1	45.5/47.7
Hossain et al. [7]	35.2/35.7	40.8/39.3	37.2/44.6	37.4/43	43.2/47.2	44.0/54	38.9/38.3	35.6/37.5	42.3/51.6	44.6/61.3	39.7/46.5	39.7/41.4	40.2/47.3	32.8/34.2	35.5/39.4	39.2/44.1
Zhao et al. [9]	40.2/-	49.2/-	47.8/-	52.6/-	50.1/-	75.0/-	50.2/-	43.0/-	55.8/-	73.9/-	54.1/-	55.6/-	58.2/-	43.3/-	43.3/-	43.8/-
Pavllo et al. [5]	36.2/34.1	42.3/36.1	36.8/34.4	40.0/37.2	41.8/36.4	50.1/42.2	44.6/34.4	39.9/33.6	50.3/45	52.6/52.5	40.8/37.4	42.0/33.8	42.7/37.8	31.6/25.6	33.9/27.3	41.1/36.5
Yang et al. [8]	-/26.9	-/30.9	-/36.3	-/39.9	-/43.9	-/47.4	-/28.8	-/29.4	-/36.9	-/58.4	-/41.5	-/30.5	-/29.5	-/42.5	-/32.2	-/37.7
Ours	**34.2/26.1**	**41.2/31.3**	**35.9/28.4**	**37.5/29.1**	**40.6/32.4**	**47.0/38.0**	**40.6/29.8**	**35.0/27.1**	**43.6/36.4**	**46.0/40.1**	**39.3/31.2**	**40.9/31.0**	**39.2/24.0**	**30.1/33.4**	**31.6/36.3**	**38.6/31.2**

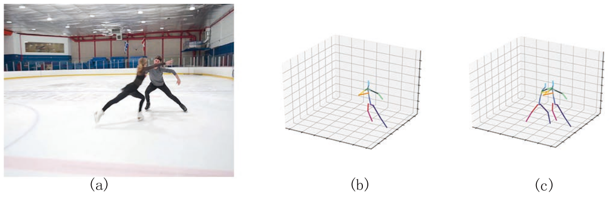

(a) (b) (c)

Figure 2: The detection loss can be solved in the occlusion case. (a) is a source frame; (b) the estimated 3D pose without our occlusion processing; (c) the estimated 3D pose with our occlusion processing.

training loss function is to minimize Eq. (3) based on the constant $\omega = 0.1$.

$$L_{set} = L_{mes} + \omega L_{sym} \tag{3}$$

2.2 Occlusion Processing

We predict the occluded keypoint based on the position information of the previous frame and the movement direction of the parent keypoint of the limb. If the entire limb is occluded, we deal with the arms and legs separately: if one leg is occluded, we can restore the occluded keypoint from the other leg according to the symmetry; if both legs are occluded, the body is in the half-blocked state and no extra processing is required; if the arm is occluded, there are self-occlusion and other blockings, meaning that we cannot restore the occluded keypoint based on the symmetry. In this case, we employ the Evolving Temporal Proposals [6] to determine the position information of the parent keypoint at the previous frame to obtain the occluded keypoint position(Eq. (4)).

$$P_i^t = P_i^{t-1} + \Delta \tag{4}$$

where P_i^t is predicted the 3D position of occluded keypoint i at time t, and P_i^{t-1} denotes the 3D position of the $t-1$ moment correspondingly. Δ is a vector representing the increment of the unoccluded parent keypoint. After our occlusion processing, the detection loss can be solved as shown in Figure 2.

2.3 Data Augmentation

When training our 3DPN model, we search for a 2D-vector λ to scale the ground truth 2D keypoints P_{2D} of the person, to distinguish the "big but far" person and the "small but near" person. Let the corresponding ground truth 3D keypoints be P_{3D}, we minimize Eq. (5) in which P_{3D}^{xy} denotes the (x, y) coordinates of the P_{3D}.

$$argmin_{\lambda} ||\lambda P_{2D} - P_{3D}^{xy}||^2 \tag{5}$$

After we obtain the λ, we use the λP_{2D} to train our 3DPN instead of P_{2D}, which can improve the accuracy 1% higher than our original results for the 3D human pose estimation.

3 Experiments

Training setting. For DN and 2DPN, the pre-trained model can work well for people detection and 2D pose recognition. We mainly use the Human3.6M dataset to train our 3DPN with the same settings as [4, 5, 7–9]. The initial learning rate is set to 0.001 with exponential decay every 10 epochs and a dropout rate $p = 0.25$. The training time is about 10 hours.

Evaluation. The quality evaluation results are given in the accompanying video. For quantitative evaluation, we mainly use Mean Per Joint Position Error(MPJPE)/the Procrustes analysis MPJPE (P_MPJPE)([5]) to evaluate our method as shown in Table 1. Some methods such as [3] report better quantitative results by using multi-frames or multi-views as the input to estimate the 3D poses. To the best of our knowledge, we get the best quantitative results for the 3D pose evaluation based on the single frame input.

4 Conclusion

As a top-down method, our 'DN-2DPN-3DPN' framework effectively combines the detection and human pose estimation methods and can generate real-time multi-person 3D poses based on a single RGB camera in practice. Experiments validate that our framework outperforms state-of-the-art methods in the Human3.6M dataset.

Acknowledgments

This work is being supported in part by the National Key R&D Program of China (No. 2020YFC2008500), in part by the National Natural Science Foundation of China (Nos. 61620106003, U2003109, 61971418, 61802406, and 61972459), and by Open Research Projects of Zhejiang Lab(No. 2021KE0AB07).

References

[1] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang. Bottom-up higher-resolution networks for multi-person pose estimation. *arXiv preprint arXiv:1908.10357*, 2019.

[2] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.

[3] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, and V. Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *CVPR*, pp. 5063–5072, 2020.

[4] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, pp. 2640–2649, 2017.

[5] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, pp. 7753–7762, 2019.

[6] H. Qiu, Y. Zheng, H. Ye, Y. Lu, F. Wang, and L. He. Precise temporal action localization by evolving temporal proposals. In *ICMR*, pp. 388–396, 2018.

[7] M. Rayat Imtiaz Hossain and J. J. Little. Exploiting temporal information for 3d human pose estimation. In *ECCV*, pp. 68–84, 2018.

[8] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. In *CVPR*, pp. 5255–5264, 2018.

[9] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, pp. 3425–3435, 2019.

[10] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.