2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506588

RGB-D FUSION FOR POINT-CLOUD-BASED 3D HUMAN POSE ESTIMATION

Jiaming Ying, Xu Zhao

Department of Automation, Shanghai Jiao Tong University
Key Laboratory of System Control and Information Processing, Ministry of Education of China

ABSTRACT

3D human pose estimation is an important and challenging task in computer vision. In this paper, we propose a method to estimate 3D human pose from RGB-D images. We adopt a 2D pose estimator to extract color features from the RGB image. The color features are integrated with the depth image in the form of point cloud. To fully exploit geometric information, we design a 3D learning module to extract point-wise features. To take advantage of local information as well as facilitate the convergence of the model, we design a dense prediction module. It estimates the offset vectors and closeness scores from points to target keypoints. The point-wise estimations are weighted and summed up to a final 3D pose. Experimental results show that our method achieves state-of-the-art performance on MHAD and SURREAL datasets.

Index Terms— 3D human pose estimation, RGB-D, point cloud, dense prediction

1. INTRODUCTION

3D human pose estimation aims to localize human body keypoints in 3D space. It is a fundamental and important computer vision task with diverse applications, including action recognition, human-robot interaction, and sports analysis.

Most recent approaches are based on a monocular RGB image. They are faced with challenges including depth ambiguity and the highly nonlinear 2D-to-3D mapping [1, 2, 3, 4]. Another drawback of RGB-based methods is that they have difficulty in estimating an absolute localization, which limits its application scenarios. Though some recent approaches [5, 6] try to predict absolute 3D poses by exploiting scale cues from 2D images, the results are just approximations.

To obtain absolute pose in real-world metric space, many algorithms take depth images as input. Shotton *et al.*[7] first segment body parts by a random forest, then localize 3D keypoints. Moon *et al.*[8] transform the depth map into a voxelized grid and perform a voxel-to-voxel prediction by a 3D CNN. One major defect of this approach is its high complexity of time and space. To reduce time consumption, Xiong *et al.*[9] propose an anchor-based method that directly applies 2D convolution to depth images. Point cloud is another 3D data format. In 3D hand pose estimation, Ge *et al.*[10, 11] regress hand joints from point clouds using PointNet [12, 13].

The RGB-only and depth-only methods have their respective pros and cons, which motivates us to combine these two kinds of methods. In addition, the advent of low-cost RGB-D sensors also makes it practical. Equipped with depth information, we are able to obtain absolute 3D pose and handle the problem of depth ambiguity. Depth information also facilitates 2D-to-3D mapping. Correspondingly, the color and texture information in the RGB image is crucial to keypoint localization. Moreover, the model can benefit from various RGB datasets including outdoor scenes. Despite the above advantages, RGBD-based methods have not been extensively studied [14, 15], probably due to the lack of RGB-D datasets.

In this work, we propose a method to regress 3D human keypoints from RGB-D images. The core of our approach is to integrate color features into the 3D point cloud. There are three reasons to estimate 3D pose based on point cloud: (1) learning in 3D space can better exploit geometric information; (2) comparing to voxelized grid, point cloud brings less information loss; (3) recent point-based 3D deep learning methods [13, 16] make it possible to extract fine-grained features from point clouds.

As depicted in Fig. 1, our model consists of three modules. A 2D pose estimator first generates heatmaps from the RGB image. The heatmaps act as a preliminary yet useful prediction that contributes to 3D estimation in later stages. Next, the 2D fusion module fuses heatmaps with the depth image and outputs a point cloud where each point is endowed with a color feature vector. The downsampled points are then fed into a 3D learning module to generate point-wise features. Finally, the model makes dense predictions, *i.e.* offset vectors and closeness scores from each point to target keypoints. The final result is voted by all points.

To summarize, the contributions of this work include:

- We extend an effective 3D neural network to 3D human pose estimation task.
- To better utilize local features, we design a point-wise

This work has been supported in part by the funding from NSFC (61673269, 61273285) and the project funding of the Institute of Medical Robotics at Shanghai Jiao Tong University. (Corresponding author: Xu Zhao. E-mail: zhaoxu@sjtu.edu.cn)

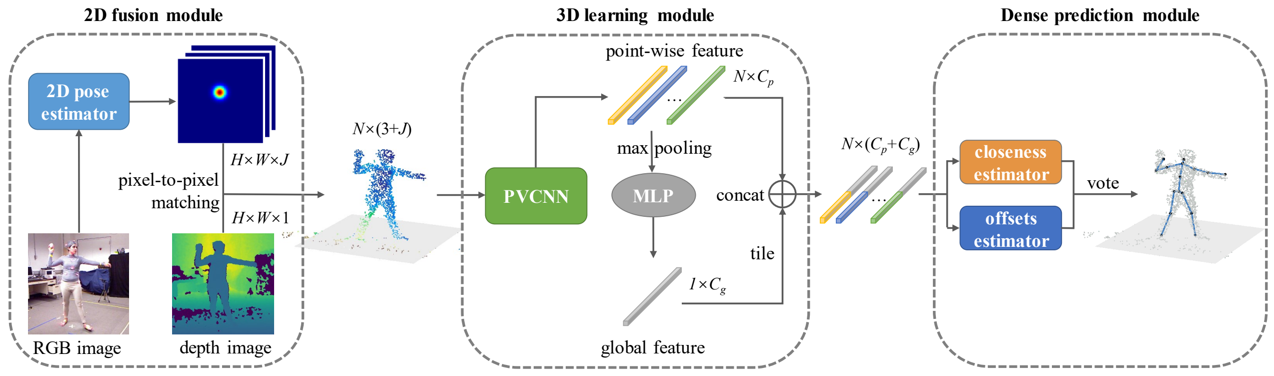

Fig. 1. The pipeline of our approach. The input to our model is a pair of aligned RGB and depth images. Color features extracted from the RGB image are fused with depth information. They are converted into a point cloud and passed through the 3D learning module to generate point-wise features. The final 3D pose is voted by all points based on dense prediction results.

voting mechanism that predicts closeness scores and offset vectors from each point to target keypoints.
- We demonstrate the state-of-the-art performance on two public datasets, MHAD [17] and SURREAL [18].

2. METHOD

Our goal is to estimate 3D human pose from a pair of aligned RGB image $\mathbf{R}\in \mathbb{R}^{H\times W\times 3}$ and depth image $\mathbf{D}\in \mathbb{R}^{H\times W\times 1}$. By combining color features from the RGB image and geometry features from the depth image, the model is able to localize J human keypoints $\mathbf{K} = (\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_J) \in \mathbb{R}^{3\times J}$ in the camera coordinate system. Fig. 1 illustrates the pipeline of our method. Details are described below.

2.1. 2D Fusion Module

Color feature extraction. With the release of diverse large-scale datasets, the state-of-the-art 2D pose estimation methods are able to generate accurate and robust results. Hence an off-the-shelf 2D pose estimator is adopted to exploit color and texture features from the RGB image. Considering both accuracy and speed, we choose to use SimpleBaseline [19] which consists of ResNet-50 [20] and a series of deconvolutional layers. It outputs a set of heatmaps $\mathbf{H}\in \mathbb{R}^{H'\times W'\times J}$ where the value of each pixel indicates the presence likelihood of a specific keypoint.

Color-depth fusion. The heatmaps are first upsampled to the original image size $H \times W$ by bilinear interpolation. Then each pixel on the heatmap is matched with its corresponding pixel on the depth map.

2.2. 3D Learning Module

This module aims to learn geometric features and further extract color features in 3D space. To better utilize the geometric information encoded in depth images, we transform the depth image to point cloud using camera intrinsic parameters. Before being fed to the downstream module, the point cloud is randomly downsampled to a fixed size N. It can be represented by a $N \times (3 + J)$ matrix, where each point has its own XYZ coordinate plus a J-dim vector of color feature.

Point-wise feature extraction. We take advantage of PVCNN [16] to exploit geometric information from the point cloud. PVCNN is an efficient 3D deep learning method which uses a multi-layer perceptron (MLP) to model fine-grained point-wise feature and a 3D CNN to capture coarse-grained voxel-based information. On the one hand, the coarse-grained feature provides context information. It helps to generate a feasible pose, since human body parts interact with each other. On the other hand, the fine-grained feature is essential for accurate localization. For each point, PVCNN generates a feature vector that is embedded with context information covering different spatial sizes.

Global-local feature fusion. Besides, the point-wise features are passed through a global pooling layer and an MLP to obtain a global feature vector. It is then tiled and concatenated to the point-wise features. This step aims to enable every point with the ability to "sense" the global context.

2.3. Dense Prediction Module

The point-wise features generated in the last stage provide a premise for dense prediction. In this way, the model can take advantage of local information to achieve accurate localization. Dense prediction is also conducive to the convergence of the network. Actually, we have tried a direct regression approach, but it failed to converge.

There are two branches in the prediction module. The first branch outputs the closeness score of each point to the target keypoint. Specifically, it predicts whether a point is in the neighborhood of a keypoint. Fig. 2(a) shows the neighboring points of the left knee in red. The ground truth closeness score

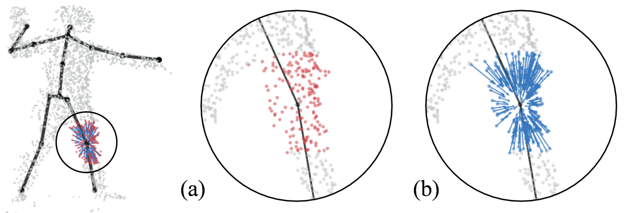

Fig. 2. The network output of the left knee. (a) Red points indicate the neighboring points of the left knee. (b) Arrows indicate the offsets from different points to the target keypoint. For better visualization, only a part of vectors are visualized.

is generated as

$$c_i^j = \begin{cases} 1, & \|\mathbf{k}_j - \mathbf{p}_i\|_2 \leq \mathrm{R}, \\ 0, & otherwise; \end{cases} \tag{1}$$

where $\mathbf{p}_i$ is the i-th point in the point cloud, $\mathbf{k}_j$ is the j-th human keypoint, R is the radius of neighborhood.

Another branch calculates the offset vectors from each point to each keypoint:

$$\mathbf{v}_i^j = \mathbf{k}_j - \mathbf{p}_i. \tag{2}$$

In other words, every point makes a guess at the keypoint location. Fig. 2(b) illustrates an example of the left knee.

To obtain one final output, scores from the closeness branch serve as weight factors. Because we assume that points close to the target keypoint tend to output more reliable offsets. The final keypoint location is a result of per-point voting, which can be formulated as

$$\hat{\mathbf{k}}_j = \frac{\sum_{i=1}^N \hat{c}_i^j (\mathbf{p}_i + \hat{\mathbf{v}}_i^j)}{\sum_{i=1}^N \hat{c}_i^j}, \tag{3}$$

where $\hat{c}_i^j$ and $\hat{\mathbf{v}}_j$ stand for the predicted closeness score and offset vector from point i to keypoint j respectively.

The loss function consists of three terms. The first term calculates the difference between the predicted pose and the ground truth:

$$\mathcal{L}_K = \frac{1}{J} \sum_{j=1}^{J} \mathcal{H}(\hat{\mathbf{k}}_j - \mathbf{k}_j), \tag{4}$$

where $\mathcal{H}$ means Huber loss. We choose Huber loss instead of MSE loss, for it is less sensitive to outliers while maintaining strong convexity near zero. The function will focus more on reducing small errors and help to generate a result as accurate as possible. Huber loss is defined as

$$\mathcal{H}(x) = \begin{cases} \frac{1}{2}x^2, & x \leq \delta, \\ \delta(|x| - \frac{1}{2}\delta), & otherwise; \end{cases} \tag{5}$$

where δ is a threshold factor.

MSE loss is used to reduce closeness estimation error:

$$\mathcal{L}_C = \frac{1}{N}\frac{1}{J} \sum_{i=1}^{N} \sum_{j=1}^{J} (\hat{c}_i^j - c_i^j)^2. \tag{6}$$

To further reduce the regression error, we also supervise the offset branch with Huber loss:

$$\mathcal{L}_O = \frac{1}{N}\frac{1}{J} \sum_{i=1}^{N} \sum_{j=1}^{J} \mathcal{H}(\hat{\mathbf{v}}_i^j - \mathbf{v}_i^j). \tag{7}$$

It plays a subsidiary role to keypoint regression since it treats offsets from different points equally regardless of their closeness scores.

Finally, the overall loss function is a weighted sum of the above three terms:

$$\mathcal{L} = \mathcal{L}_K + \lambda_C \mathcal{L}_C + \lambda_O \mathcal{L}_O. \tag{8}$$

3. EXPERIMENTS

3.1. Experimental Setting

Datasets. Experiments are conducted on MHAD [17] and SURREAL [18]. On MHAD, we use subjects 8, 11 for evaluation, and the other ten subjects for training. Only RGB-D frames from the front view are used. To reduce redundancy, the frame rate is downsampled to 10 FPS. On SURREAL, we only use subset *run2* and follow the original dataset partitioning. The frame rate is downsampled to 2 FPS. We pick 16 commonly evaluated keypoints in both datasets.

Evaluation metrics. We use MPJPE$_{abs}$ to measure the mean Euclidean distance between the predicted and target absolute coordinates. MPJPE$_{rel}$ is also used as root-relative MPJPE. Another metric is 3DPCK, which measures the percentage of correctly localized keypoints. The threshold is set to 150 mm in our experiments.

3.2. Implementation Details

RGB-D images are first cropped according to human bounding boxes. We resize RGB images to a fixed size of 256×256 as input to the 2D pose estimator. We delete background pixels from depth images by setting a depth threshold. For color feature extraction, we use the pretrained SimpleBaseline [19] model and fine-tune it on MHAD and SURREAL. The 3D learning module and the dense prediction module are trained from scratch for 40 epochs. The learning rate is set to 0.001 initially and reduced by a factor of 2 at epochs 5, 18, 30. We choose Adam as optimizer. The radius R of the keypoint neighborhood is set to 200 mm. In loss function, $\lambda_C = 200$, $\lambda_O = 0.01$. We train the 2D pose estimator and 3D modules separately. To train the 3D modules, we apply augmentation to ground truth heatmaps in an attempt to simulate the error distribution of predicted heatmaps. Similar to [6], we first analyze the 2D detection error and then synthesize noisy heatmaps as input to 3D modules.

3.3. Ablation Study

We first look into the impact of color information on 3D human pose estimation. As shown in the first three rows of Table 1, directly adding RGB values reduces the prediction error slightly. While integrating color information in the form of 2D heatmaps brings about a reduction of 4.56 mm in MPJPE$_{abs}$. This suggests the effectiveness of combining color features through a 2D pose estimator.

Table 1. Ablation study on MHAD.

Color	Loss	Training heatmap	MPJPE$_{abs}$
-	$\mathcal{L}_K + \mathcal{L}_C$	-	35.31
RGB	$\mathcal{L}_K + \mathcal{L}_C$	-	34.81
heatmap	$\mathcal{L}_K + \mathcal{L}_C$	GT w/ aug	30.75
heatmap	$\mathcal{L}_K + \mathcal{L}_C + \mathcal{L}_O$	GT w/ aug	29.77
heatmap	$\mathcal{L}_K + \mathcal{L}_C + \mathcal{L}_O$	GT w/o aug	33.32
heatmap	$\mathcal{L}_K + \mathcal{L}_C + \mathcal{L}_O$	prediction	35.01

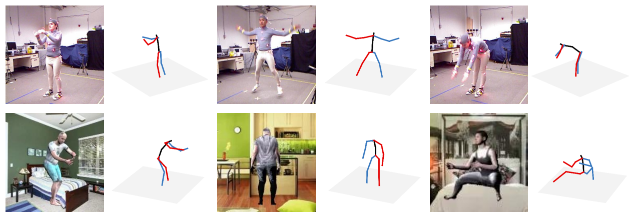

Fig. 3. Qualitative results. The first row shows results on MHAD. The second row shows results on SURREAL.

Table 2. Comparison with SOTA methods on MHAD.

Method	Input	MPJPE$_{abs}$
Shotton *et al.*[7]	depth	68
Makris *et al.*[15]	RGB+depth	66
Michel *et al.*[21]	RGB+depth	58
NaiveLifting	RGB+depth	121.35
Ours	RGB+depth	**29.77**
Ours (GT)	RGB+depth	20.41

Table 3. Comparison with SOTA methods on SURREAL.

Method	Input	MPJPE$_{rel}$	3DPCK
Tung *et al.*[22]	video+heatmap	64.4	-
Varol *et al.*[23]	RGB+2D pose +segmentation	46.1	-
Wang *et al.*[24]	RGB	37.1	97.3
Martinez *et al.*[4]	2D pose	70.0	89.3
NaiveLifting	RGB+depth	123.1	70.3
Ours	RGB+depth	**19.2**	**99.0**
Ours (GT)	RGB+depth	14.3	99.8

We also evaluate the impact of loss term $\mathcal{L}_O$. As described in Section 2.3, it plays a subsidiary role that penalizes offset prediction errors. Results in Table 1 prove that supervision on offset vectors contributes to the overall performance.

As for the training strategy, we compare three different ways to generate heatmaps. The input heatmaps to 3D modules are prediction results of the 2D pose estimator. They are supervised during training but unsupervised during evaluation. This makes the input data distribution varies at training time and test time. As shown in the last row of Table 1, using supervised estimations from the 2D estimator produces the worst result, which suggests that the data distribution deviation has a negative impact on training. By performing data augmentation as described in Section 3.2, the distribution deviation is reduced, leading to the lowest MPJPE$_{abs}$. This makes our model robust to noisy 2D heatmaps.

3.4. Comparison with State-of-the-art Methods

In this section, we compare our model with state-of-the-art methods. Additionally, we evaluate a simple method – NaiveLifting. It directly lifts 2D pose to 3D using the depth values at 2D landmarks. To distinguish 2D detection error from 2D-to-3D lifting error, we also test our method with ground truth 2D heatmaps (specified as *Ours (GT)*). It sets an upper bound on our 3D modules.

Table 2 shows the comparison results on MHAD. All methods take RGB-D images as input, except for the one proposed by Shotton *et al.*[7]. Since there have been no protocols for MHAD so far, the comparison may not be completely fair, yet the results can still tell us something. Our approach produces the lowest MPJPE$_{abs}$. The large error generated by NaiveLifting probably comes from the noise in depth images and self-occlusion. This demonstrates that extracting features from the whole depth image is essential.

We also compare with other methods on SURREAL. Notice the various inputs to different methods. We re-implement the method proposed by Martinez *et al.*[4] using the publicly released codes. Results of other methods are taken from the corresponding papers. As shown in Table 3, our method outperforms other methods by a large margin in MPJPE$_{rel}$. This proves the effectiveness of our RGB-D combination method. Qualitative results are shown in Fig. 3.

It is worth to note the difference between testing on ground truth heatmaps and estimated heatmaps (the last 2 rows in Table 2 and Table 3). It mainly stems from two parts: the data distribution deviation during training and evaluation, and the error from 2D detection itself.

3.5. Runtime Analysis

The average inference time is 21 ms for a frame on a single NVIDIA 2080 Ti GPU, including 10 ms for 2D detection and 11 ms for 3D regression. This reveals that our model can run in real-time at approximately 47.6 FPS.

4. CONCLUSION

In this work, we propose a method to estimate absolute 3D human pose from RGB-D input. Color features are first extracted by a 2D CNN and then fused with depth information. The 3D learning module converts depth image to point cloud, exploiting point-wise features and global features from it. We also design a dense prediction scheme and generate the final result by point-wise voting. Experimental results on two datasets demonstrate the superiority of our method.

5. REFERENCES

[1] Chen Li and Gim Hee Lee, "Generating multiple hypotheses for 3d human pose estimation with mixture density network," in *CVPR*, June 2019.

[2] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei, "Integral human pose regression," in *ECCV*, September 2018.

[3] Ching-Hang Chen and Deva Ramanan, "3d human pose estimation = 2d pose estimation + matching," in *CVPR*, July 2017.

[4] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, Oct 2017.

[5] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," in *ICCV*, October 2019.

[6] Ju Yong Chang, Gyeongsik Moon, and Kyoung Mu Lee, "Absposelifter: Absolute 3d human pose lifting network from a single noisy 2d human pose," *CoRR*, vol. abs/1910.12029, 2019.

[7] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011, pp. 1297–1304.

[8] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," in *CVPR*, June 2018.

[9] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan, "A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image," in *ICCV*, October 2019.

[10] Liuhao Ge, Zhou Ren, and Junsong Yuan, "Point-to-point regression pointnet for 3d hand pose estimation," in *ECCV*, September 2018.

[11] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan, "Hand pointnet: 3d hand pose estimation using point sets," in *CVPR*, June 2018.

[12] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, July 2017.

[13] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*. 2017, vol. 30, pp. 5099–5108, Curran Associates, Inc.

[14] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3d human pose estimation in rgbd images for robotic task learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1986–1992.

[15] A. Makris and A. Argyros, "Robust 3d human pose estimation guided by filtered subsets of body keypoints," in *2019 16th International Conference on Machine Vision Applications (MVA)*, 2019, pp. 1–6.

[16] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han, "Point-voxel cnn for efficient 3d deep learning," in *Advances in Neural Information Processing Systems*. 2019, vol. 32, pp. 965–975, Curran Associates, Inc.

[17] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley mhad: A comprehensive multimodal human action database," in *WACV*, 2013, pp. 53–60.

[18] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid, "Learning from synthetic humans," in *CVPR*, July 2017.

[19] Bin Xiao, Haiping Wu, and Yichen Wei, "Simple baselines for human pose estimation and tracking," in *ECCV*, September 2018.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, June 2016.

[21] Damien Michel, Ammar Qammaz, and Antonis A. Argyros, "Markerless 3d human pose estimation and tracking based on rgbd cameras: An experimental evaluation," in *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, 2017, p. 115–122.

[22] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki, "Self-supervised learning of motion capture," in *Advances in Neural Information Processing Systems*, 2017, vol. 30, pp. 5236–5246.

[23] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid, "Bodynet: Volumetric inference of 3d human body shapes," in *ECCV*, September 2018.

[24] Zhe Wang, Daeyun Shin, and Charless C Fowlkes, "Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation," *arXiv preprint arXiv:2004.03143*, 2020.