

PersPose: 3D Human Pose Estimation with Perspective Encoding and Perspective Rotation

Xiaoyang Hao* Han Li†

Southern University of Science and Technology, China

{haoxy2022, lih2022}@mail.sustech.edu.cn

Abstract

Monocular 3D human pose estimation (HPE) methods estimate the 3D positions of joints from individual images. Existing 3D HPE approaches often use the cropped image alone as input for their models. However, the relative depths of joints cannot be accurately estimated from cropped images without the corresponding camera intrinsics, which determine the perspective relationship between 3D objects and the cropped images. In this work, we introduce Perspective Encoding (PE) to encode the camera intrinsics of the cropped images. Moreover, since the human subject can appear anywhere within the original image, the perspective relationship between the 3D scene and the cropped image differs significantly, which complicates model fitting. Additionally, the further the human subject deviates from the image center, the greater the perspective distortions in the cropped image. To address these issues, we propose Perspective Rotation (PR), a transformation applied to the original image that centers the human subject, thereby reducing perspective distortions and alleviating the difficulty of model fitting. By incorporating PE and PR, we propose a novel 3D HPE framework, PersPose. Experimental results demonstrate that PersPose achieves state-of-the-art (SOTA) performance on the 3DPW, MPI-INF-3DHP, and Human3.6M datasets. For example, on the in-the-wild dataset 3DPW, PersPose achieves an MPJPE of 60.1 mm, 7.54% lower than the previous SOTA approach. Code is available at: <https://github.com/KenAdamsJoseph/PersPose>.

1. Introduction

Estimating 3D human pose from a single image is a crucial task in computer vision, with numerous practical applications such as AR/VR and human-computer interaction. The objective of 3D human pose estimation (HPE) is to recover

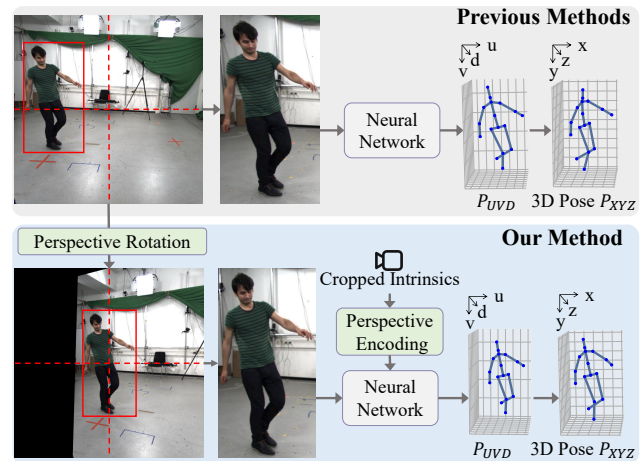


Figure 1. Comparison of previous 3D HPE frameworks and ours.

the 3D coordinates of human joints from images, whereas 3D human pose and shape estimation (HPS) aims to predict the 3D coordinates of human mesh vertices. Some 3D HPS methods [4, 16, 24, 29, 42] treat 3D HPE as a sub-task, first estimating the 3D positions of the joints (or a larger set of mesh vertices) and then reconstructing the human mesh.

Existing 3D HPE methods [15, 23, 28, 32, 36] and 3D HPS methods [4, 16, 17, 24, 29, 42] typically use cropped images as input to estimate 3D joint positions, meaning that the region corresponding to the human bounding box is cropped from the original image, as illustrated in Figure 1. However, the CLIFF method [18] demonstrates that using cropped images alone cannot accurately estimate global rotation (i.e., the orientation of the human in the camera coordinate system), thus it employs both bounding box information and the cropped images as inputs to the neural network. In this paper, we demonstrate through the example in Figure 2 that when only the cropped image is provided as input, the relative depths of joints cannot be accurately estimated either. Additionally, the example in Figure 3 reveals that even when the full image is used, the absence of field of view angle (FOV) information results in an inaccurate estimation of the joints' relative depths. Moreover, cropping can be con-

*Equal contribution

†Corresponding author

sidered as a modification of the camera intrinsics and sensor resolution; in this context, the camera intrinsics corresponding to the cropped image effectively encapsulate both the cropping and FOV information. We refer to the camera intrinsics corresponding to the cropped image as cropped intrinsics, denoted as K^{crop} , which include the focal length f^{crop} and the principal point coordinates c_x^{crop} and c_y^{crop} . We propose a Perspective Encoding (PE) module that encodes the cropped intrinsics as a 2D PE map, which is then jointly fed into the CNN with the cropped image.

Since a person can appear at any location within an image, the principal points of the cropped intrinsics c_x^{crop} and c_y^{crop} vary significantly across different samples. This reflects substantial changes in the perspective relationship between the 3D scene and the cropped image. Additionally, the further the human subject deviates from the image center, the more pronounced the perspective distortions of the cropped image become. These factors increase the difficulty of model fitting. To mitigate these issues, we propose a Perspective Rotation (PR) module, which centers the human subject and generates a centered image from the original image, thereby ensuring that c_x^{crop} and c_y^{crop} remain fixed across samples while also reducing perspective distortions.

In this study, we propose a 3D HPE framework, PersPose, which incorporates the proposed PE and PR modules, as illustrated in Figure 1. First, the original image undergoes the PR module, yielding a centered image (see Section 4.2 for details), which is then cropped from the center. Subsequently, the cropped intrinsics K^{crop} , calculated using Eq. 4, are encoded by the PE module to form a PE map, as detailed in Section 4.1. The PE map and the cropped image are then fed into a CNN backbone, and then a decoder is used to predict the 3D Pose P_{XYZ} . The contributions of this paper can be summarized as follows:

- Without camera intrinsics information, the relative depths of joints cannot be accurately estimated. To mitigate this previously overlooked limitation, we propose an innovative PE module that encodes the cropped intrinsics as a 2D PE map, which is then jointly fed into the CNN with the cropped image.
- Due to the variation in a person’s location within the original images, the perspective relationships between the 3D scene and the cropped image vary significantly, which complicates model fitting. To address this, we propose a novel PR module that centers the human subject, thereby reducing perspective distortions and alleviating the difficulty of model fitting.
- We propose a 3D HPE framework, PersPose, incorporating our PE and PR modules. To evaluate our framework, we conduct comprehensive experiments on 3DPW [33], Human3.6M [7], and MPI-INF-3DHP [25] datasets. Experimental results show that our framework outperforms existing state-of-the-art (SOTA) methods. Ablation stud-

ies further validate the effectiveness of the proposed PE and PR modules.

2. Related Work

Some 3D HPE methods [15, 23, 28, 32] directly estimate 3D joint coordinates. Pavlakos et al. [28] proposed to voxelize the 3D space into a 3D probabilistic heatmap. Sun et al. [32] proposed a differentiable soft-argmax operation to derive joint coordinates from estimated heatmaps. Ma et al. [23] proposed an attention-based context modeling framework that propagates structural cues across joints and effectively suppresses absurd 3D pose estimates. Li et al. [15] designed RLE from a maximum-likelihood perspective to learn flexible output distributions for joint positions, thereby facilitating the training process.

Other methods [16, 24, 29, 36], including ours, first estimate 2D pixel coordinates and relative depths of joints, and then convert these into 3D joint coordinates based on camera intrinsics. This approach facilitates joint training with 2D keypoint datasets. The principal point is usually near the image center. As for focal length, some studies assume it to be unknown. Some methods [8, 12] employ a weak perspective model, which effectively corresponds to using a large fixed focal length. Kissos et al. [10] proposed estimating a rough focal length using image resolution. The SPEC method [13] incorporates a camera calibration sub-network to estimate the focal length directly from images, while the Zolly method [35] estimates the human-to-camera distance and scale from images to derive the focal length. However, the focal length of a camera is usually obtainable (e.g., through device APIs/SDKs) and is essential for estimating the relative depths of joints, as illustrated in Figure 3.

The CLIFF method [18] demonstrates that global rotation, i.e., the orientation of the human in the camera coordinate system, cannot be accurately inferred from cropped images alone. Their work underscores the necessity of incorporating bounding box information. To address the adverse effects of cropping on global orientation, Yao et al. [39] proposed applying a compensatory rotation within their self-supervised 3D HPE framework to correct the global orientation of the estimated 3D skeleton.

Furthermore, some 3D HPE methods utilize 2D keypoint sequences as input. The recent Ray3D method [41] leverages camera intrinsics to convert 2D keypoints into 3D rays, and subsequently employs a temporal network to recover the 3D pose sequence from the 3D ray sequence. Additionally, methods like ElePose [34] and EPOCH [5] estimate camera pose within unsupervised 3D HPE frameworks.

3. Importance of Camera Intrinsics in 3D HPE

In the following, we first introduce camera intrinsics and FOV. Then, we demonstrate the limitation of using cropped

images alone for 3D HPE. Finally, we discuss the importance of FOV for 3D HPE when images are not cropped. Since the cropping operation can be considered as a modification of the camera intrinsics and resolution, the cropped intrinsics are inherently linked to the cropping operation and FOV; this underscores the significance of camera intrinsics in 3D HPE.

Camera Intrinsics and FOV. Camera intrinsics are used to map 3D coordinates onto 2D image coordinates, playing a pivotal role in 3D HPE. The intrinsic matrix K is a 3×3 matrix:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where f denotes the effective focal length in pixel units. If the virtual sensor* moves along the z -axis, f scales accordingly. The coordinates (c_x, c_y) represent the principal point, which is typically located near the image center. Adjusting (c_x, c_y) is equivalent to moving the virtual sensor within the x - y plane.

The FOV quantifies the extent of the observable world through a camera. FOV is determined by the effective focal length f and the image size:

$$\text{FOV}_x = 2 \arctan\left(\frac{w}{2f}\right), \text{FOV}_y = 2 \arctan\left(\frac{h}{2f}\right), \quad (2)$$

where FOV_x and FOV_y are the horizontal and vertical fields of view, respectively. h and w represent the height and width of the captured image in pixels.

Limitation of Depth Estimation from Cropped Images.

In the preprocessing stage of existing 3D HPE methods [15, 23, 32, 36] and 3D HPS methods [4, 16, 24, 29, 42], original images are typically cropped based on the bounding boxes of human subjects. However, using only cropped images for 3D HPE introduces significant difficulty due to the lack of important information about the position of the human subject within the camera’s view frustum. This information directly relates to how human subjects are projected onto images from the camera’s perspective.

Figure 2 gives visual examples demonstrating the limitation of using only cropped images for 3D HPE. A camera captures three human subjects. The original image is displayed in the upper right, while the lower right shows three cropped images corresponding to the three human subjects. For subjects (a) and (b), their cropped images are very similar despite having different relative depth labels. Similarly,

*The virtual sensor is a mathematical abstraction positioned on the virtual image plane, located at a distance equal to the focal length in front of the optical center.

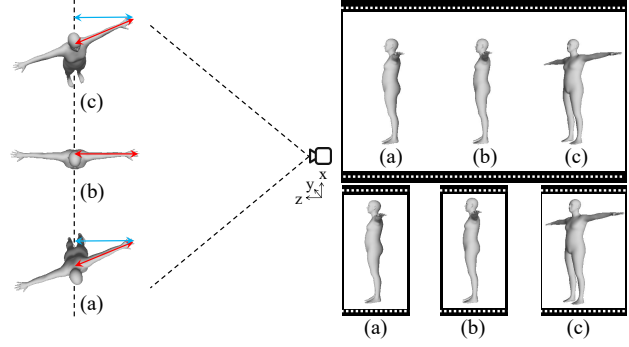


Figure 2. Examples that illustrate the limitation of depth estimation using cropped images alone. A single camera captures three human subjects with the identical body shape, and the resulting images are subsequently cropped. The blue lines denote the relative depths (i.e., to the pelvis) of a particular joint, while the red lines measure the arm lengths. In addition, the red lines for (a) and (c) are parallel. For cropped images, two key observations are noted: 1) subjects (a) and (b), despite having different relative depths, yield the same cropped images; 2) subjects (a) and (c), which share the same relative depths, produce distinct cropped images.

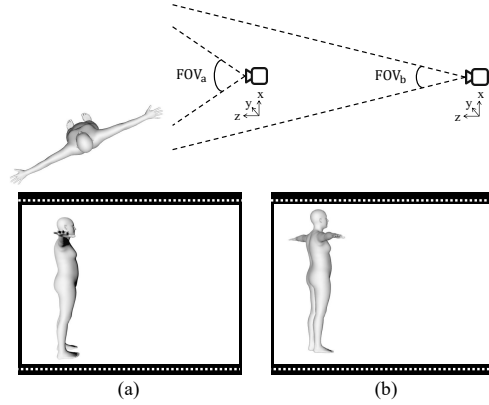


Figure 3. Examples that illustrate the importance of FOV to depth estimation. Two cameras at varying distances from a human subject capture images, resulting in images (a) and (b). Both cameras have the same optical axes but different FOV. Although there are significant visual differences between the two images, they share identical relative depth labels.

the cropped images of subjects (a) and (c) are quite different, even though they share the same relative depth labels.

Furthermore, cropping an image is equivalent to capturing a photo using another camera with the same camera extrinsics but with different camera intrinsics and resolution. The same camera extrinsics mean keeping the position of the pinhole and the camera’s orientation fixed in the world coordinate system, and the modification of the camera intrinsics from K to K^{crop} can be geometrically viewed as a translation of the virtual sensor. Specifically, K^{crop} is calculated by: $K^{\text{crop}} = AK$, where A is the affine transformation

matrix corresponding to the cropping operation. The matrix A is defined as

$$A = \begin{bmatrix} s & 0 & t_u \\ 0 & s & t_v \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where $s = w^{\text{bbox}}/w^{\text{crop}}$ (or $s = h^{\text{bbox}}/h^{\text{crop}}$), $t_u = w^{\text{crop}}/2 - s c_u^{\text{bbox}}$, $t_v = h^{\text{crop}}/2 - s c_v^{\text{bbox}}$. Here, w^{crop} and h^{crop} denote the resolution of the cropped image, w^{bbox} and h^{bbox} represent the bounding box resolution, and c_u^{bbox} and c_v^{bbox} are the center coordinates of the bounding box. Given the original camera intrinsics in Eq. 1, the cropped intrinsics K^{crop} become

$$K^{\text{crop}} = AK = \begin{bmatrix} f^{\text{crop}} & 0 & c_x^{\text{crop}} \\ 0 & f^{\text{crop}} & c_y^{\text{crop}} \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

where $f^{\text{crop}} = sf$, $c_x^{\text{crop}} = sc_x + t_u$, and $c_y^{\text{crop}} = sc_y + t_v$.

Consequently, the cropped intrinsics K^{crop} can be utilized to map 3D coordinates onto the 2D coordinate system of the cropped image.

Importance of FOV to Depth Estimation. In Figure 3, two cameras with the same optical axis capture the same human subject. The left camera has a larger field of view angle, i.e., $FOV_a > FOV_b$, and it is closer to the human subject compared to the right camera. These two cameras produce images (a) and (b). Note that (a) and (b) look quite different: for image (a), the depth of the right wrist relative to the pelvis seems larger than in image (b). However, images (a) and (b) actually have the same relative depth labels. This example demonstrates that the FOV must also be considered to infer relative depths accurately.

4. Method

In this section, we detail the proposed PersPose, illustrated in Figure 4. First, the original image I undergoes the PR process, during which a rotation matrix R is applied to center the human subject, yielding a centered image I' (see Section 4.2 for details). Then, the image I' is cropped from the center, resulting in the cropped image I^{crop} . Subsequently, the cropped intrinsics K^{crop} , calculated using Eq. 4, is encoded to form a 2D map M^{xy} using PE, as detailed in Section 4.1. This PE map M^{xy} and the cropped image I^{crop} are separately processed through distinct convolutional layers before being element-wise added. The fused features are further passed through a backbone network, and then a decoder is used to predict 2D joint coordinates and relative depths of joints P_{UVD} , as well as a scale factor $\hat{s} = d_0^{\text{abs}}/f^{\text{crop}}$. Here, d_0^{abs} is the absolute depth of the pelvis joint in the camera coordinate system, and scale factor \hat{s} is expressed in mm/pixel. These outputs are then used to compute the rotated 3D pose P'_{XYZ} . For a joint

index by i , the calculation from $P_{\text{UVD},i} = [u_i, v_i, d_i]^\top$ to $P'_{\text{XYZ},i} = [x'_i, y'_i, z'_i]^\top$ is given by

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = d_i^{\text{abs}} (K^{\text{crop}})^{-1} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}, \quad (5)$$

where $d_i^{\text{abs}} = d_i + \hat{s} f^{\text{crop}}$. Finally, the inverse rotation R^\top is applied to P'_{XYZ} to derive P_{XYZ} , the 3D pose for the original image I .

$$P_{\text{XYZ}} = R^\top P'_{\text{XYZ}}. \quad (6)$$

4.1. Perspective Encoding

We propose PE to encode cropped intrinsics. As illustrated in Figure 5, three virtual sensors have different cropped intrinsics $K_{\text{red}}^{\text{crop}}$, $K_{\text{green}}^{\text{crop}}$ and $K_{\text{blue}}^{\text{crop}}$. The red virtual sensor has a larger focal length compared to the green and blue virtual sensors. Additionally, the red and green virtual sensors are aligned along the optical axis, while the blue virtual sensor is placed off-axis, which is common for cropped images. In this setup, the principal points of $K_{\text{red}}^{\text{crop}}$ and $K_{\text{green}}^{\text{crop}}$ are located at the image center, while the principal point of $K_{\text{blue}}^{\text{crop}}$ is not. We project virtual sensors at different positions onto a fixed reference plane at $z = 1$. The red virtual sensor has a larger focal length than the other two virtual sensors, thus its corresponding projected region is smaller. Similarly, the principal point of the blue virtual sensor deviates from the optical axis, and its corresponding projected region also shifts away from the axis. In other words, for each sensor, the corresponding projected area geometrically represents the unique view frustum determined by its cropped intrinsics. Consequently, we employ this projected area as the encoded representation of the cropped intrinsics.

Specifically, we construct the PE map M^{xy} by projecting each pixel coordinates (u_i, v_i) of the cropped image onto the plane at $z = 1$, as shown in Figure 5, and the projected coordinates are computed as:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = (K^{\text{crop}})^{-1} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}. \quad (7)$$

Consequently, the pixel coordinate map M^{uv} , defined as

$$M^{uv} = \begin{bmatrix} (1, 1) & (1, 2) & \cdots & (1, w) \\ (2, 1) & (2, 2) & \cdots & (2, w) \\ \vdots & \vdots & \ddots & \vdots \\ (h, 1) & (h, 2) & \cdots & (h, w) \end{bmatrix}, \quad (8)$$

is transformed into the PE map M^{xy} :

$$M^{xy} = \begin{bmatrix} (x_1, y_1) & (x_1, y_2) & \cdots & (x_1, y_w) \\ (x_2, y_1) & (x_2, y_2) & \cdots & (x_2, y_w) \\ \vdots & \vdots & \ddots & \vdots \\ (x_h, y_1) & (x_h, y_2) & \cdots & (x_h, y_w) \end{bmatrix}. \quad (9)$$

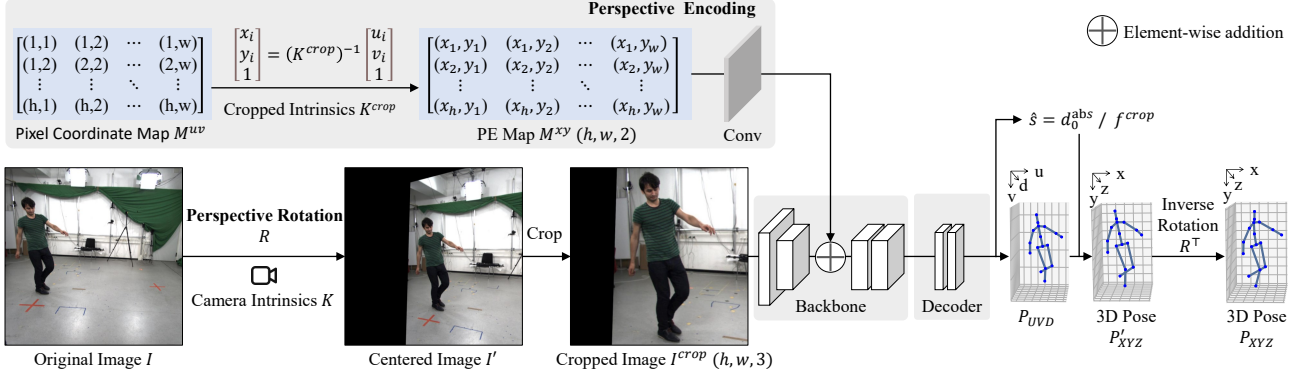


Figure 4. **Overview of the PersPose.** PersPose introduces two key components: 1) PE, which encodes the cropped intrinsics as a 2D map M^{xy} ; and 2) PR, which centers the human subject in the image, reducing the difficulty of model fitting.

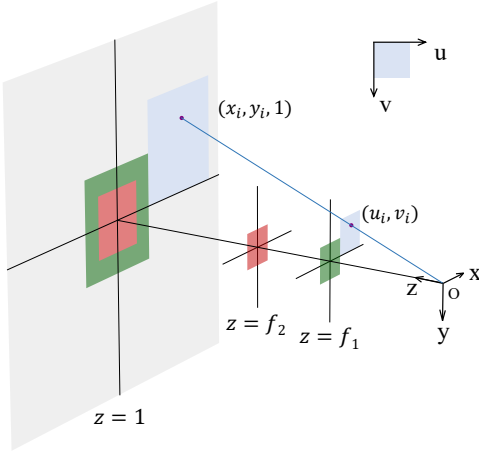


Figure 5. **Perspective Encoding.** The grey region denotes the plane at $z = 1$ in the camera coordinate system. The red, green, and blue virtual sensors, corresponding to different cropped intrinsics, are placed on the planes at $z = f_1$ and $z = f_2$. The focal lengths f_1 and f_2 , measured in meters, correspond to two effective focal lengths measured in pixel units. The projected areas of these three virtual sensors on the plane at $z = 1$ are also denoted in red, green, and blue. (u_i, v_i) denotes a pixel coordinate on the virtual sensor, and the corresponding 3D point projected onto the $z = 1$ plane is $(x_i, y_i, 1)$. For each virtual sensor, the corresponding projected area on the plane at $z = 1$ geometrically encodes its unique view frustum. Consequently, we use the projected area as the encoding result of the cropped intrinsics.

The PE map M^{xy} encodes the view frustum and is subsequently fed into the 3D HPE model, enabling the model to capture the perspective variations induced by different cropped intrinsics.

4.2. Perspective Rotation

In Section 3, we explain the importance of camera intrinsics in 3D HPE. Then, we introduce PE to encode cropped intrinsics K^{crop} as part of model input in Section 4.1. Given a

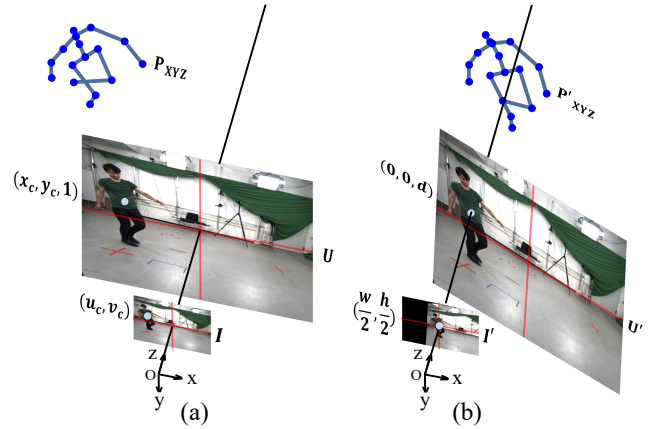


Figure 6. **Perspective Rotation.** In Figure (a), a virtual sensor captures an image I that is marked with the human bounding box center (u_c, v_c) . An upscaled image U is added to the scene, placed on the plane at $z = 1$, and the human bounding box center $(x_c, y_c, 1)$ on U is also marked. Additionally, a 3D human skeleton P_{XYZ} is added to the scene. The scene, including the 3D skeleton P_{XYZ} and the upscaled image U , is rotated around the optical center O from the setting in (a) to the setting in (b), so that the optical axis points to the human subject. After applying this rotation, the upscaled image U' in (b) is then reprojected onto the virtual sensor in (b), and the centered image I' in Figure 4 is obtained.

cropped image I^{crop} and the corresponding cropped intrinsics K^{crop} (composed of f^{crop} , c_x^{crop} , and c_y^{crop}), a 3D HPE model, parameterized by θ , aims to learn the following mapping function:

$$f_\theta : (I^{crop}, f^{crop}, c_x^{crop}, c_y^{crop}) \rightarrow P_{XYZ}. \quad (10)$$

Notably, the principal point (c_x^{crop}, c_y^{crop}) may exhibit considerable variance across different images, as the human subject can appear at any region within the original image. This substantially increases the difficulty of model fitting. To mitigate this challenge, we introduce PR to center the

human subject, as shown in Figure 4. This ensures that the cropped principal point $(c_x^{\text{crop}}, c_y^{\text{crop}})$ is fixed to the center of the cropped image across different images. Therefore, the mapping function changes from Eq. 10 to

$$\tilde{f}_\theta : (I^{\text{crop}}, f^{\text{crop}}) \rightarrow P_{XYZ}, \quad (11)$$

thus reducing the model’s fitting difficulty.

In the following, we describe the process of PR. As illustrated in Figure 4, through PR, the human subject in the original image I is reprojected to the center in the image I' . The reprojection process of this example is shown in Figure 6.

In Figure 6(a), the virtual sensor, the upscaled image U , and the 3D skeleton P_{XYZ} form a perspective relationship: if a ray is projected from a particular joint of the P_{XYZ} towards the optical center O , this ray passes through the corresponding joint in the original image I on the sensor, and also through the corresponding joint in the upscaled image U .

The coordinates (u_c, v_c) and $(x_c, y_c, 1)$ in Figure 6(a) denote the center of the human bounding box on the sensor and on the upscaled image U , respectively. Here, $(x_c, y_c, 1)$ is calculated as

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix}, \quad (12)$$

where K is the camera intrinsics of the original image I .

The goal of PR is to center the bounding box in the centered image I' . To achieve this, we rotate the U around the optical center such that the bounding box center $(x_c, y_c, 1)$ is brought onto the z -axis at $(0, 0, d)$, where $d = \|(x_c, y_c, 1)\|_2$, as shown in Figure 6(b). Therefore, the rotation axis \mathbf{n} and rotation angle ϕ can be computed as follows:

$$\mathbf{n} = \frac{(x_c, y_c, 1) \times (0, 0, d)^\top}{\|(x_c, y_c, 1) \times (0, 0, d)^\top\|}, \quad (13)$$

$$\phi = \arccos \left(\frac{(x_c, y_c, 1) \cdot (0, 0, d)}{\|(x_c, y_c, 1)\| \|(0, 0, d)\|} \right), \quad (14)$$

where \times denotes the cross product, and \cdot represents the dot product. Applying Rodrigues’ rotation formula, the rotation matrix is:

$$R = \text{Rodrigues}(\mathbf{n}, \phi) \quad (15)$$

This rotation R is first applied to the upscaled image U in Figure 6(a) and obtain U' in Figure 6(b). U' is then reprojected onto the virtual sensor in (b) and the centered image I' is obtained, with the bounding box center located at image center $(\frac{w}{2}, \frac{h}{2})$.

The 3D pose corresponding to the centered image I' can be derived by applying the same rotation R to P_{XYZ} :

$$P'_{XYZ} = RP_{XYZ}. \quad (16)$$

Since both U and P_{XYZ} undergo the same 3D rotation about the optical center, the perspective relationship of the centered image I' , the rotated upscaled image U' , and the rotated 3D skeleton P'_{XYZ} is maintained, as illustrated in Figure 6(b). Consequently, the rotated 3D skeleton P'_{XYZ} maintains geometric consistency with the new image I' .

Finally, the perspective transformation matrix M used to warp the original image I into the centered image I' is computed as follows:

$$M = KRK^{-1}, \quad (17)$$

where K is the camera intrinsics of the original image I .

5. Experiments

5.1. Experimental setup

We set the resolution of the cropped images to 256×256 . The HRNet-W48 [31] is used as the backbone, which outputs a feature map of dimensions $64 \times 64 \times 48$ and a global feature vector of length 2048. The Perspose decoder comprises one convolutional layer and two linear layers. The convolutional layer processes the feature map to produce a 2D heatmap, from which the 2D pose is obtained using the soft-argmax operation. Simultaneously, the two linear layers take the global feature vector as input to predict the relative depths of joints and the scale factor \hat{s} , respectively. Additionally, the element-wise addition in Figure 4 is performed before the second stage of HRNet.

To compare with HPS methods, we additionally add two linear layers into the Perspose decoder to estimate human shape and twist, and the analytical IK algorithm in HybriK [16] is used to derive SMPL parameters prediction.

The loss function is defined as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{uvd}} + \lambda_2 \mathcal{L}_{\hat{s}} + \lambda_3 \mathcal{L}_{\text{beta}} + \lambda_4 \mathcal{L}_{\text{tw}}, \quad (18)$$

where \mathcal{L}_{uvd} is the L1 loss for the P_{UVD} predictions, $\mathcal{L}_{\hat{s}}$ is the L1 loss for the estimated scale factor \hat{s} , $\mathcal{L}_{\text{beta}}$ is the L1 loss for the estimated human shape parameters, \mathcal{L}_{tw} is the L2 loss for the twist predictions, and $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are weight hyperparameters, set to 1.0, 0.5, 0.01 and 0.1 respectively.

The batch size is set to 96. The AdamW [22] optimizer is used with an initial learning rate of $3e-4$. The training lasts for 70 epochs, and the learning rate is reduced by a factor of 0.75 every 6 epochs. This process utilizes one NVIDIA 3090 GPU and is completed in around 60 hours.

To evaluate the estimated 3D pose, we report Mean Per Joint Position Error (MPJPE), Procrustes Aligned MPJPE (PA-MPJPE), Percentage of Correct Keypoints (PCK), and Area Under Curve (AUC). Additionally, Per Vertex Error (PVE) is computed to evaluate all vertices on the human mesh. Furthermore, we report on the depth error in our ablation experiments to evaluate the estimated relative depths.

| | | 3DPW | | | Human3.6M | | | MPI-INF-3DHP | | |
|----------------------|---------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|
| | | PA-MPJPE↓ | MPJPE↓ | PVE↓ | PA-MPJPE↓ | MPJPE↓ | PVE↓ | PCK↑ | AUC↑ | MPJPE↓ |
| HMR [8] † | CVPR'18 | 81.3 | 130.0 | 152.7 | 56.8 | 88.0 | 96.1 | 72.9 | 36.5 | 124.2 |
| SPIN [14] † | ICCV'19 | 59.2 | 96.9 | 116.4 | 41.1 | - | - | 76.4 | 37.1 | 105.2 |
| I2L-MeshNet [26] † | ECCV'20 | 57.7 | 93.2 | 110.1 | 41.1 | 55.7 | 65.1 | - | - | - |
| Pose2Mesh [3] † | ECCV'20 | 58.3 | 88.9 | 106.3 | 46.3 | 64.9 | 85.3 | - | - | - |
| Mesh Graphormer [19] | ICCV'21 | 45.6 | 74.7 | 87.7 | 34.5 | 51.2 | - | - | - | - |
| HybrIK [16] ‡ | CVPR'21 | 45.0 | 74.1 | 86.5 | 34.5 | 54.4 | 65.7 | 87.5 | 46.9 | 93.9 |
| CLIFF [18] | ECCV'22 | 43.0 | 69.0 | 81.2 | 32.7 | 47.1 | - | - | - | - |
| FastMETRO [2] | ECCV'22 | 44.6 | 73.5 | 84.1 | 33.7 | 52.2 | - | - | - | - |
| IKOL [42] ‡ | AAAI'23 | 45.5 | 73.3 | 86.4 | - | - | - | 87.9 | 48.1 | 88.8 |
| VirtualMarker [24] ‡ | CVPR'23 | 41.3 | 67.5 | 77.9 | 32.0 | 47.3 | <u>58.0</u> | - | - | - |
| ProPose [4] ‡ | CVPR'23 | 40.6 | 68.3 | 79.4 | <u>29.1</u> | <u>45.7</u> | - | - | - | - |
| PLIKS [29] ‡ | CVPR'23 | 42.8 | 66.9 | 82.6 | 34.7 | 49.3 | - | <u>91.8</u> | <u>52.3</u> | <u>72.3</u> |
| Zolly [35] | ICCV'23 | 39.8 | <u>65.0</u> | <u>76.3</u> | 32.3 | 49.4 | - | - | - | - |
| Gwon et al. [6] | CVPR'24 | 44.3 | 73.2 | 80.3 | - | - | - | - | - | - |
| GLNet-W48 [37] | ECCV'24 | <u>39.5</u> | 66.9 | 77.9 | 29.4 | 48.8 | - | - | - | - |
| PostoMETRO [38] | WACV'25 | 39.8 | 67.7 | 76.8 | - | - | - | - | - | - |
| PersPose ‡ | | 39.1 | 60.1 | 72.4 | 28.3 | 43.0 | 52.7 | 94.0 | 55.2 | 72.1 |

Table 1. **Comparison between our method and SOTA HPS methods.** ↓ indicate that lower values are better, while ↑ indicate that higher values are better. † denotes that 3DPW training split is not used. ‡ denotes methods that estimate UVD coordinates as an intermediate representation prior to obtaining 3D XYZ positions.

The depth error is defined as the mean absolute difference between the estimated and ground-truth relative depths of joints. Following [16, 24], We evaluate 14 joints on 3DPW and Human3.6M datasets and 17 joints on MPI-INF-3DHP dataset. For MPJPE, PA-MPJPE, PCK, and AUC, the joints regressed from the human mesh are evaluated, while the relative depths predicted by the PersPose decoder is used to compute depth error.

Our experiments employ the following 3D datasets: 1) 3DPW [33], an in-the-wild dataset annotated with SMPL parameters, which can be used to derive 3D pose labels. 2) Human3.6M [7], which was captured in a controlled environment. We use the SMPL parameters derived from Mosh [21]. Following [9, 11], we use 5 subjects (S1, S5, S6, S7, S8) for training, and 2 subjects (S9, S11) for evaluation. 3) BEDLAM [1], a large and realistic synthetic dataset annotated with accurate SMPL parameters. 4) MPI-INF-3DHP [25], a 3D human pose dataset that includes both controlled indoor settings and complex outdoor scenes.

5.2. Comparison with the State-of-the-art

Table 1 presents a comparison between our method and SOTA HPS methods on 3DPW, Human3.6M, and MPI-INF-3DHP datasets. In this study, we employ HRNet-W48 [31] as the backbone, and results utilizing the same backbone from other works will be preferred when available. The results for the proposed PersPose in Table 1 were obtained by training on a combination of datasets, including 3DPW [33], Human3.6M [7], MPI-INF-3DHP [25], and

COCO [20]. Note that while some methods may have incorporated additional 2D keypoint datasets during training, none of the results in Table 1 were derived using synthetic 3D datasets.

PersPose outperforms other methods across all three datasets. On MPI-INF-3DHP benchmark, PersPose surpasses the second-best approach by 2.2 on PCK and 2.9 on AUC. Furthermore, on 3DPW dataset, PersPose outperforms the second-best approach by 4.9 mm (7.54%) in MPJPE. The test splits of MPI-INF-3DHP and 3DPW datasets both contain complex outdoor scenes, demonstrating the robustness of the proposed PersPose under challenging real-world conditions.

5.3. Ablation Study

Experimental Setup. To evaluate the effectiveness of our PE and PR modules, we conduct ablation studies comprising the following configurations: 1) our framework without PE and PR, 2) our framework without PR, and 3) our framework with both PE and PR.

For evaluation metrics, we employ PA-MPJPE and MPJPE to assess the accuracy of 3D pose estimation. In addition, Section 3 highlights the importance of camera intrinsics for relative depth estimation with two examples (Figure 2 and Figure 3). Section 4.1 introduces PE to transfer camera intrinsics to the neural model. Section 4.2 proposes PR to reduce the difficulty of model fitting. To evaluate the effectiveness of our proposed PE and PR in estimating relative depths, we additionally report on the depth error.

| PR | PE | Dataset | 3DPW | | | MPI-INF-3DHP | | |
|----|----|---------|--------------|-------------|-------------|--------------|-------------|-------------|
| | | | Depth error↓ | PA-MPJPE↓ | MPJPE↓ | Depth error↓ | PA-MPJPE↓ | MPJPE↓ |
| - | - | R | 45.1 | 39.8 | 62.4 | 57.3 | 57.3 | 80.1 |
| - | ✓ | | 44.5 | 39.7 | 62.2 | 53.7 | 56.3 | 76.6 |
| ✓ | ✓ | | 43.8 | 39.1 | 60.1 | 51.0 | 54.4 | 71.9 |
| - | - | R+B | 41.5 | 37.8 | 58.4 | 54.2 | 55.5 | 76.8 |
| - | ✓ | | 41.2 | 37.8 | 58.1 | 51.0 | 55.1 | 73.4 |
| ✓ | ✓ | | 40.0 | 37.3 | 57.2 | 48.6 | 53.7 | 70.2 |

Table 2. Ablation experiments on 3DPW [33] and MPI-INF-3DHP [25]. R denotes the use of real datasets, and B denotes BEDLAM.

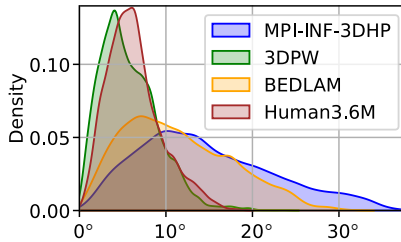


Figure 7. **Distribution of PR rotation angle ϕ .** We analyze the distribution of ϕ (measured in degrees) across various datasets. A broader spread in ϕ reflects greater variability in the centre of the bounding box’s coordinates (u_c, v_c) . The values of ϕ for 3DPW and Human3.6M concentrate around 5° . In contrast, BEDLAM and MPI-INF-3DHP exhibit a wider spread, suggesting these datasets feature more diversity and complexity.

Prior work has demonstrated that training on synthetic data improves accuracy [1, 29, 30]. In this study, we further investigate the impact of incorporating synthetic data into the training phase. Specifically, in Table 2, “R” represents training on real-image datasets (3DPW, Human3.6M, MPI-INF-3DHP, and COCO), while “B” denotes training on BEDLAM synthetic dataset.

Ablation Study Results. Table 2 displays the results of our experiments. The addition of PE or PR consistently leads to performance improvements across both datasets, regardless of whether the additional synthetic dataset is used for training, indicating the effectiveness of the proposed PE and PR in relative depth estimation and 3D pose estimation.

On MPI-INF-3DHP dataset, the addition of PE or PR results in a considerable reduction in depth error. Specifically, PE reduces depth error by 3.6mm and 3.2mm for training settings “R” and “R+B”, respectively. Similarly, PR achieves reductions in depth error of 2.7mm and 2.4mm for “R” and “R+B”, respectively.

On 3DPW dataset, incorporating PE or PR leads to a relatively modest performance improvement compared to MPI-INF-3DHP dataset. To investigate this effect, we analyzed the distribution of the rotation angle ϕ in PR across different datasets.

Distribution of the rotation angle ϕ across datasets.

Figure 7 displays the distribution of ϕ , using kernel density estimation to estimate the density curve. The rotation angle ϕ for MPI-INF-3DHP spans a broader range than 3DPW. This distribution quantifies the variability in the person’s position within the dataset’s images: a wider spread in ϕ indicates greater variation in the center of the human bounding box (u_c, v_c) , reflecting significant changes in the perspective relationship between the 3D scene and the cropped image; in such cases, the benefits of PE and PR become even more pronounced.

Effect of Synthetic Training Data. Comparing the training settings “R+B” and “R” across various model configurations, we observe that the model trained with “R+B” consistently outperforms the one trained with “R” under identical model configurations. For instance, by incorporating additional synthetic training data, the MPJPE of our framework with PE and PR decreases from 60.1 mm to 57.2 mm on 3DPW dataset and from 71.9 mm to 70.2 mm on MPI-INF-3DHP dataset. This underscores the beneficial impact of incorporating additional synthetic datasets during training.

6. Conclusion

In this paper, we propose a novel 3D HPE framework, PersPose, integrating PE and PR. We first underscore the necessity of camera intrinsics for accurately inferring the relative depths of joints in 3D HPE and then introduce the PE module to encode camera intrinsics as a 2D map, which is jointly fed into the CNN alongside the cropped image. Considering that the human subject may appear anywhere in an image, we propose the PR module to center the human subject, thereby reducing perspective distortion and the difficulty of model fitting. Experimental results on the multiple datasets demonstrate that PersPose achieves considerable improvement over existing SOTA methods. Ablation studies confirm the effectiveness of PE and PR and demonstrate that the performance enhancements provided by these modules become more pronounced when perspective relationships are more variable.

References

- [1] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023. 7, 8
- [2] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *European Conference on Computer Vision*, pages 342–359. Springer, 2022. 7
- [3] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 769–787. Springer, 2020. 7
- [4] Qi Fang, Kang Chen, Yinghui Fan, Qing Shuai, Jiefeng Li, and Weidong Zhang. Learning analytical posterior probability for human mesh recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8781–8791, 2023. 1, 3, 7
- [5] Nicola Garau, Giulia Martinelli, Niccolò Bisagno, Denis Tomè, and Carsten Stoll. Epoch: Jointly estimating the 3d pose of cameras and humans. *arXiv preprint arXiv:2406.19726*, 2024. 2
- [6] Mi-Gyeong Gwon, Gi-Mun Um, Won-Sik Cheong, and Wonjun Kim. Instance-aware contrastive learning for occluded human mesh reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10553–10562, 2024. 7
- [7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian 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, 2013. 2, 7
- [8] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018. 2, 7
- [9] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019. 7
- [10] Imry Kissos, Lior Fritz, Matan Goldman, Omer Meir, Edward Oks, and Mark Kliger. Beyond weak perspective for monocular 3d human pose estimation. In *Computer Vision—ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 541–554. Springer, 2020. 2
- [11] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5252–5262, 2020. 7
- [12] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. Pare: Part attention regressor for 3d human body estimation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11107–11117, 2021. 2
- [13] Muhammed Kocabas, Chun-Hao P. Huang, Joachim Tesch, Lea Müller, Otmar Hilliges, and Michael J Black. Spec: Seeing people in the wild with an estimated camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11035–11045, 2021. 2
- [14] Nikos Kolotouros, Georgios Pavlakos, Michael Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2252–2261, 2019. 7
- [15] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11025–11034, 2021. 1, 2, 3
- [16] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3382–3392, 2021. 1, 2, 3, 6, 7
- [17] Jiefeng Li, Siyuan Bian, Qi Liu, Jiasheng Tang, Fan Wang, and Cewu Lu. Niki: Neural inverse kinematics with invertible neural networks for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12933–12942, 2023. 1
- [18] Zhihao Li, Jianzhuang Liu, Zhenqiang Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *European Conference on Computer Vision*, pages 590–606. Springer, 2022. 1, 2, 7
- [19] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12939–12948, 2021. 7
- [20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7
- [21] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220–1, 2014. 7
- [22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6
- [23] Xiaoxuan Ma, Jiajun Su, Chunyu Wang, Hai Ci, and Yizhou Wang. Context modeling in 3d human pose estimation: A unified perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6238–6247, 2021. 1, 2, 3
- [24] Xiaoxuan Ma, Jiajun Su, Chunyu Wang, Wentao Zhu, and Yizhou Wang. 3d human mesh estimation from virtual markers. In *Proceedings of the IEEE/CVF Conference on Com-*

- puter Vision and Pattern Recognition*, pages 534–543, 2023. [1](#), [2](#), [3](#), [7](#)
- [25] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017. [2](#), [7](#), [8](#)
- [26] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII*, page 752–768, 2020. [7](#)
- [27] Priyanka Patel, Chun-Hao P Huang, Joachim Tesch, David T Hoffmann, Shashank Tripathi, and Michael J Black. Agora: Avatars in geography optimized for regression analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13468–13478, 2021.
- [28] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7025–7034, 2017. [1](#), [2](#)
- [29] Karthik Shetty, Annette Birkhold, Srikrishna Jaganathan, Norbert Strobel, Markus Kowarschik, Andreas Maier, and Bernhard Egger. Pliks: A pseudo-linear inverse kinematic solver for 3d human body estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 574–584, 2023. [1](#), [2](#), [3](#), [7](#), [8](#)
- [30] Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J. Black. Wham: Reconstructing world-grounded humans with accurate 3d motion. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. [8](#)
- [31] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. [6](#), [7](#)
- [32] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European conference on computer vision (ECCV)*, pages 529–545, 2018. [1](#), [2](#), [3](#)
- [33] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 601–617, 2018. [2](#), [7](#), [8](#)
- [34] Bastian Wandt, James J Little, and Helge Rhodin. Elepose: Unsupervised 3d human pose estimation by predicting camera elevation and learning normalizing flows on 2d poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6635–6645, 2022. [2](#)
- [35] Wenjia Wang, Yongtao Ge, Haiyi Mei, Zhongang Cai, Qingping Sun, Yanjun Wang, Chunhua Shen, Lei Yang, and Taku Komura. Zolly: Zoom focal length correctly for perspective-distorted human mesh reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3925–3935, 2023. [2](#), [7](#)
- [36] Zhe Wang, Daeyun Shin, and Charless C Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *Computer vision–ECCV 2020 workshops: Glasgow, UK, August 23–28, 2020, proceedings, part II 16*, pages 523–540. Springer, 2020. [1](#), [2](#), [3](#)
- [37] Yabo Xiao, Mingshu He, and Dongdong Yu. Global-to-pixel regression for human mesh recovery. In *European Conference on Computer Vision*, pages 481–497. Springer, 2024. [7](#)
- [38] Wendi Yang, Zihang Jiang, Shang Zhao, and S Kevin Zhou. Postometro: Pose token enhanced mesh transformer for robust 3d human mesh recovery. *arXiv preprint arXiv:2403.12473*, 2024. [7](#)
- [39] Yao Yao, Yixuan Pan, Wenjun Shi, Dongchen Zhu, Lei Wang, and Jiamao Li. Rotated orthographic projection for self-supervised 3d human pose estimation. In *European Conference on Computer Vision*, pages 422–439. Springer, 2024. [2](#)
- [40] Yingxuan You, Hong Liu, Ti Wang, Wenhao Li, Runwei Ding, and Xia Li. Co-evolution of pose and mesh for 3d human body estimation from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14963–14973, 2023.
- [41] Yu Zhan, Fenghai Li, Renliang Weng, and Wongun Choi. Ray3d: ray-based 3d human pose estimation for monocular absolute 3d localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13116–13125, 2022. [2](#)
- [42] Juzhe Zhang, Ye Shi, Yuexin Ma, Lan Xu, Jingyi Yu, and Jingya Wang. Ikol: Inverse kinematics optimization layer for 3d human pose and shape estimation via gauss-newton differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3454–3462, 2023. [1](#), [3](#), [7](#)