FasterVoxelPose+: Fast and Accurate Voxel-based 3D Human Pose Estimation by Depth-wise Projection Decay

Zonghuang Zhuang ZHUANGZONGHUANG@SJTU.EDU.CN
Yue Zhou* ZHOUYUE@SJTU.EDU.CN
Department of Automation, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

In terms of multi-person multi-view 3D pose estimation, voxel-based methods gain promising accuracy by directly manipulating features in 3D space. Since their high computational cost prevents them from practical applications, Faster VoxelPose was proposed to address this complication by re-projecting the 3D feature volume onto coordinate planes, which greatly improved the efficiency of the model. However, it suffers from an obvious performance drop, especially when there are fewer cameras. In this paper, we propose a more accurate real-time 3D pose estimation method, FasterVoxelPose+, to address the above problem. We have made two improvements to the previous methods. First, we propose a novel method for constructing voxel feature volume called Depth-wise Projection Decay (DPD). It introduces extra depth information to the projection to alleviate depth ambiguity. Second, we design an Encoder-Decoder Network for processing the re-projected voxel features to further push up the performance of the model. Our method obtains 17.42mm MPJPE on Panoptic with real-time speed and can be easily used in other voxel-based models.

Keywords: Human Pose Estimation, Real-time, Voxel, Multi-view

1. Introduction

In this paper, we tackle the problem of multi-person multi-view 3D pose estimation from RGB images. The goal is estimating the locations of all joints of each person in world coordinate system in a scene. It not only lays the groundwork for many human-centric downstream tasks like human-robot interaction [Shi et al. (2019)] and action recognition, but also can be used in a wide range of applications, such as 3d animation production and sports analysis.

In recent years, a series of voxel-based methods like Qiu et al. (2019); Iskakov et al. (2019); Tu et al. (2020) achieve promising results by back-projecting the 2D pose heatmaps or features of each view to the 3D space and directly predicting 3D poses from the constructed feature volume. Despite their effectiveness, the voxel-based methods suffer from slow inference speed due to the high computation complexity on 3D features. To address this problem, Ye et al. (2022) proposed Faster VoxelPose, which re-projects the 3D feature volumes to 2D coordinate planes, i.e., xy, xz, yz planes. This allows the model to using

* Corresponding author

Method	5 views	4 views	3 views	2 views	1 views
VoxelPose [Tu et al. (2020)]	17.68mm	20.01mm	24.29mm	38.94mm	66.95mm
Faster VoxelPose [Ye et al. (2022)]	18.26mm	21.12mm	26.13mm	52.23mm	133.05mm

Table 1: Comparison of MPJPE between VoxelPose and Faster VoxelPose on Panoptic.

2D CNN instead of 3D CNN to process the features, so that it can run much faster than previous voxel-based models and reach real-time speed.

However, this voxel-wise feature re-projection method brings a significant accuracy drop, especially when there are fewer cameras. Specifically, as Table 1 shows, compared with previous method Tu et al. (2020) on the CMU Panoptic dataset, the accuracy drop of Faster VoxelPose rises from 0.58mm with five views to 66.10mm with one view. Due to the high cost of deploying a large number of synchronized and calibrated cameras, in practical applications, a smaller number of cameras is beneficial to simplify the system and reduce overhead. So, improving the accuracy of real-time pose estimation models without compromising their accuracy with fewer cameras has practical application value. In this paper, we propose a new multi-person multi-view 3D pose estimation model, FasterVoxelPose+, which addresses the above problem of real-time voxel-based work in two step.

First, we introduce a novel projection method for constructing voxel features, i.e., **the Depth-wise Projection Decay**, **DPD**. We argue that the reason why the model accuracy collapses with decreasing number of views is that, in previous voxel-based methods, the way of simply projecting the 2D pose heatmaps to the 3D space loses the depth information of the original images. This leads to the subsequent networks not being able to access the spatial context information of the scene in the original images, e.g., the appearance of the human body and the background. As a result, the performance of the model is overly dependent on the cross-blending of heatmap projections from multiple views, making the model unable to cope well with a small number of views, and this issue become even worse for the re-projected features in Faster VoxelPose. Thus, in our method, we use the Depth-wise Projection Decay to incorporate depth information in the projection process.

Specifically, as shown in Figure 1, previous methods project the pixel values of the 2D heatmap to 3D space in the form of rays, where the values on a ray are equal. If 2D pose estimation is accurate, the rays of the same joint projected from multiple views will intersect at the ground truth 3D position. But if 2D pose prediction is not perfect and there are fewer views, it will be more difficult to recover 3D point. To improve this projection method, we applied a Gaussian decay centered on the root depth of the human body to the values on a ray. That is, values near the root depth have larger values, and those farther or closer will decay. Actually, this method introduces additional depth information into voxel features from the original image. In our experiments, compared with Faster VoxelPose, DPD considerably reduces the MPJPE on CMU Panoptic by 19% and 46% under two and one views respectively.

Second, we design a powerful yet efficient 2D CNN to process 2D re-projected voxel features, called **EDN**, namely **Encoder-Decoder Network**. Two EDNs are deployed to estimate 3D bounding boxes and 2D poses respectively in different stages, which will be detailed in Section 3.1. Overall, this network adopts the U-Net [Ronneberger et al. (2015)]

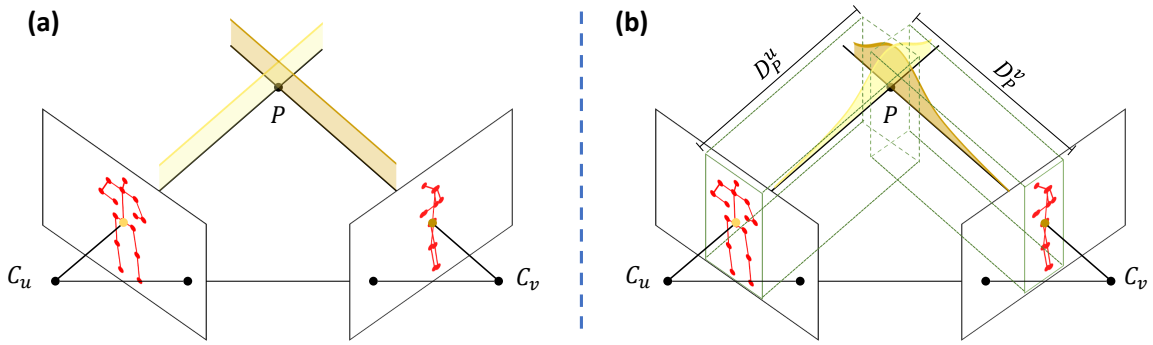

Figure 1: The Depth-wise Projection Decay, DPD. (a) The projection method in previous voxel-based model. (b) Our method. C_u and C_v are camera parameters of view u and v, D_P^u and D_P^v is the estimated depth in view u and v.

architecture to fuse multi-scale information. But we have made two effective improvements: First, we proposed a parallel decoder with deconvolution skip-connections, which passes the fine-tuned and un-fine-tuned features from the encoder to the decoder for richer multi-scale information fusion. Second, inspired by Wang et al. (2022), we adopt inverted bottleneck blocks [Sandler et al. (2018)] as the basic element in EDN, where the kernel size of the depth-wise convolution layer is increased. This 2D network reduces the MPJPE from 18.90mm to 17.54mm with negligible speed drop on CMU Panoptic.

As Figure 2 shows, our method, FasterVoxelPose+, deal with the 3D pose estimation in three stages. First, estimate 2D pose heatmaps, 2D bounding boxes and root depths of people from multi-view images. Second, construct feature volume in the range of whole scene by DPD and estimate 3D bounding boxes proposals. Third, construct feature volumes for each person proposal corresponding to the 3D bounding boxes, and recover the 3D poses from them.

The contributions of this paper are summarized as follows:

1. We propose an efficient model, FasterVoxelPose+, to further push the performance of voxel-based methods for real-time multi-person multi-view 3D pose estimation.

2. We put forward a novel method for constructing the discrete 3D feature volumes, namely **the Depth-wise Projection Decay**, **DPD**, which introduces helpful depth information hidden in the images to alleviate the depth ambiguity problem.

3. We design a lightweight encoder-decoder network to handle re-projected 2D features, and further improve the performance of the model, showing the effectiveness of large convolution kernels and multi-scale fusion for processing discrete voxel features.

4. We implement a list of experiments to evaluate our method. Our model gets 96.4 and 97.7 PCP3D on Campus and Shelf respectively and get 17.42mm MPJPE and 30.5 FPS on CMU Panoptic, which competitive results compared with SOTA methods.

2. Related Works

2.1. Multi-Person Multi-View 3D Pose Estimation

Estimating 3D poses of multiple people from multiple views is a challenging task due to the difficulty of association of people in different views. To address this problem, several methods like Re-ID features [Dong et al. (2019)], plane sweep stereo [Lin and Lee (2021)] are proposed. However, their accuracy would drop in the case of crowded scenes as 2D pose estimations are noisy. To handle occlusions in the crowd, wang et al. (2021) propose a transformer-based method with projection attention to estimate pose by direct regression.

In recent years, voxel-based methods like Ye et al. (2022), Zhang et al. (2023) and Ye et al. (2022) are proposed and cast a new light on 3D pose estimation. Voxel-based methods directly process features in 3D space, thus directly detecting human bodies in 3D space without correlating them between different views. However, due to the use of 3D CNN, the huge computational load makes the voxel-based method far away from practical applications. Recently, Faster VoxelPose [Ye et al. (2022)] greatly reduces the amount of computation by re-projecting 3D features to the coordinate planes so that they can be processed by 2D CNN. But this method also brings obvious performance degradation, which reduces its value in practical scenarios. Our method addresses exactly this problem and can be used in other voxel-based methods.

2.2. Efficient Pose Estimation

For practical application, many works reduce the computational load and memory footprint of the pose estimation models to operate on edge devices or achieve real-time speed. In terms of multi-view 3D pose estimation, Bultmann and Behnke (2021) achieves real-time speed by semantic feedback to smart edge sensors. Fabbri et al. (2020) present a heatmap compression method to reduce the computation. Lin and Lee (2021) uses plane sweep stereo to estimate 3D pose efficiently. However, the scalability of these methods is limited, making them difficult to apply to diverse scenarios. The voxel-based architecture of our model allows training with generated data and can be used in large-scale scenarios.

3. Method

3.1. Overview

The architecture of our method is shown in Figure 2. Our pipeline consists of three parts. First, given images from multiple views, we estimate 2D poses, 2D bounding boxes and body root depths. Second, a feature volume is constructed by back-projecting the 2D pose heatmaps to 3D voxel space with DPD, and then its orthographic projection onto the xy plane is fed into an EDN to predict the center and size of each person's bounding box on the plane, which form the 3D bounding box proposals with a fixed height. Finally, for each 3D proposal, build a finer-grained feature volume and input its orthographic projections on three coordinate planes into another EDN to predict the 2D poses on the planes and then recover the 3D pose.

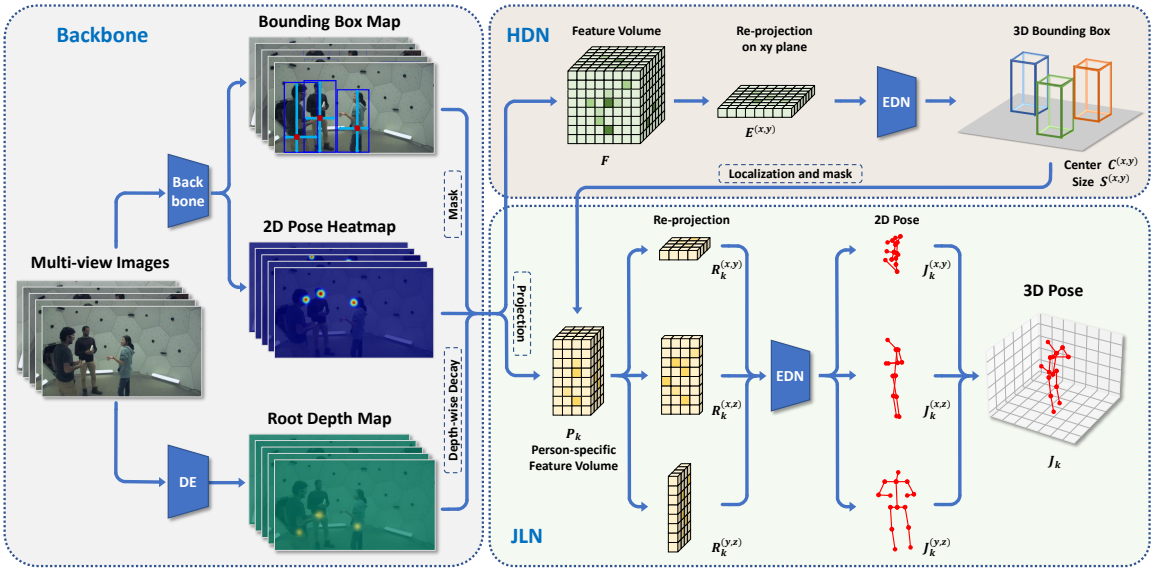

Figure 2: The architecture of our method. It is composed of three parts: Backbone, Human Detection Network (HDN) and Joint Localization Network (JLN).

In this section, we first describe our core contributions: the Depth-wise Projection Decay and the Encoder-Decoder Network, and then introduce the three parts of the model, namely the Backbone, the Human Detection Network and the Joint Localization respectively.

3.2. The Depth-wise Projection Decay

In previous voxel-based methods, each pixel intensity on the 2D pose heatmaps is projected onto a ray in 3D space with the same values when constructing feature volumes. This approach can fuse information from multiple views and inference directly in 3D space, but it still suffers from the problem of most two-stage 3D pose estimation method, that is, using only 2D pose as an intermediate representation loses the contextual information in the original images. In other words, it is more difficult for the subsequent networks to inference depth without accessing the original images. To address this issue, we propose the Depth-wise Projection Decay to introduce depth information to the process of the projection.

The Depth-wise Projection Decay, DPD is a projection method for constructing 3D voxel features. First, we discretize the 3D space into X×Y×Z locations $\{G^{x,y,z}\}$ and construct a feature volume F by back-projecting the 2D pose heatmaps into the 3D space using camera parameters. Specifically, denote the heatmap of view v as $M_v \in \mathbb{R}^{K\times H\times W}$, where K is the number of body joints. And for every location $G^{x,y,z}$, we compute its projection on view v as $P_v^{x,y,z}$, and the heatmap intensity at $P_v^{x,y,z}$ can be obtained by bilinear interpolation, which is denoted as $M_v^{x,y,z} \in \mathbb{R}^K$.

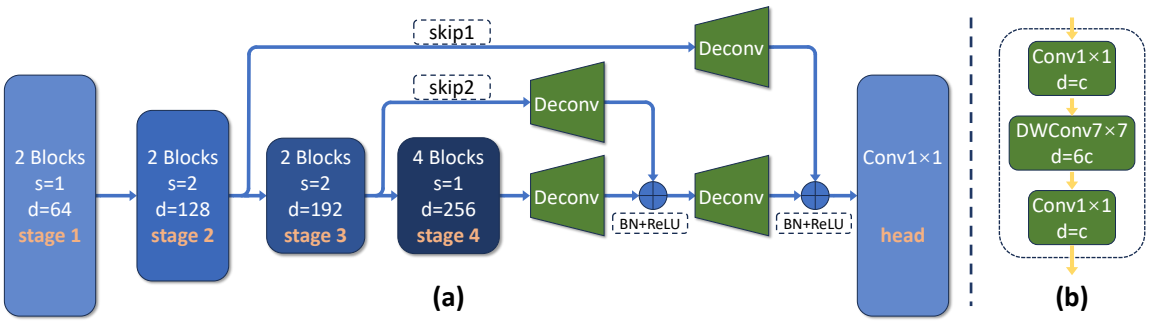

Figure 3: The Encoder-Decoder Network. (a) The architecture of EDN. (b) The basic block of EDN.

Different from the previous works [Tu et al. (2020), Zhang et al. (2023)], we do not directly use $M_v^{x,y,z}$ to calculate the voxel feature vector, instead, we decay the back-projection value $M_v^{x,y,z}$ along the depth direction according to the estimated root depth of the bounding box where the location $P_v^{x,y,z}$ is in. Specifically, if a location $G^{x,y,z}$ whose projected point $P_v^{x,y,z}$ is in the n_{th} bounding box of view v, its back-projection value $\tilde{M}_v^{x,y,z}$ is calculated by

$$\tilde{M}_v^{x,y,z} = M_v^{x,y,z} \cdot exp(-\frac{(d_v^{x,y,z} - \tilde{D}_{v,n})^2}{2\sigma^2}), \tag{1}$$

where $d_v^{x,y,z}$ is the distance between $G^{x,y,z}$ and the plane of view v, $\tilde{D}_{v,n}$ is the estimated root depth of the n_{th} person in view v, and σ is set to 200 empirically. In this formula, we decay the heatmap values so that projected values closer to the estimated root depth will be larger and those farther away will be smaller or even disappear. This introduces additional depth information for voxel features. After that, we can compute the feature vector $F^{x,y,z}$ in feature volume F corresponding to spatial location $G^{x,y,z}$ by averaging the heatmap intensities over all views:

$$F^{x,y,z} = \frac{1}{V} \sum_{v=1}^{V} \tilde{M}_v^{x,y,z}, \tag{2}$$

where V is the number of views. The feature volume is constructed once in the Human Detection Network and N (the number of people) times in the Joint Localization Network for each person, which will be detailed later.

3.3. The Encoder-Decoder Network

As in Ye et al. (2022), after the 3D feature volume is constructed, it will be orthographically projected onto the coordinate planes through maximum pooling, so that we can use 2D CNN for calculation, which greatly speeds up the inference speed of the model. To further improve the performance of the model, we design a 2D network processing the 2D voxel feature in a multi-scale manner.

As Figure 3 (a) shows, Our network adopts the encoder-decoder paradigm to fuse multi-scale information in the 2D voxel feature. It consists of a four-stage encoder and a decoder with two transposed convolutional layers for upsampling. As the size of 2D voxel feature is relatively small, i.e., typically 80×80 or 64×64 in our implementation, too many down-sampling layers will do harm to the performance. Thus, we only place two downsampling layers in the encoder, which not only ensures the accuracy but also reduces the size of the decoder. Following the common practices, we allocate more channels and blocks for later stages to deal with high-level features and gain larger receptive field. At the end, multiple prediction heads with 1×1 convolution are used to modify the channels to output.

In the experiment (see Section 4.3), we find that two improvements are important for our network. First, we do not use the regular ResNet block, instead, we employ the inverted bottleneck block in Sandler et al. (2018) for its efficiency at the low-computation region. As illustrated in Figure 3 (b), the inverted bottleneck block utilizes two point-wise convolutions to lift and compress the number of channels and apply a depth-wise in the middle to exchange information within channels, which largely reduce the parameters and calculations. In addition, following up on the findings of Liu et al. (2022), we enlarge the kernel size of the depth-wise convolutional layer, which further enhance the performance with acceptable computation increasing.

Second, motivated by HRNet [Wang et al. (2021)], we design a novel parallel decoder structure presented in Figure 3 (a). Previous work usually transfers features at the same scale with skip connections, while we use upsampling layers to add smaller-sized features in the next stage to the features in the decoder. This approach can utilize the high-level information in the refined feature and fuse information in different scale, which essentially does the same thing as the single-branch version of HRNet, but much more efficiently.

3.4. The Backbone

Unlike previous voxel-based methods, we extend the intermediate representation from only 2D pose to three parts, i.e., 2D pose, 2D bounding box and root depth to allow to implement DPD.

The 2D bounding boxes provide masks for specific 2D pose predictions to filter out other people's joints, to avoid them being projected near the wrong depth. In our implementation, following CenterNet [Zhou et al. (2019)], the bounding box and pose estimation share a ResNet-50 as backbone and are predicted by different heads. As shown in Figure 2, given a RGB image as input, the backbone predicts a 2D pose heatmap $\hat{M} \in \mathbb{R}^{H \times W \times K}$ where K is the number of joints, and a bounding box map $\hat{B} \in \mathbb{R}^{H \times W \times 4}$ where the four channels represent the distances from the body root joint to the four sides of the bounding box. The backbone is trained by

$$\mathcal{L}_{2D} = \mathcal{L}_{heat} + \lambda_{bbox}\mathcal{L}_{bbox}, \tag{3}$$

$$\mathcal{L}_{heat} = ||M - \hat{M}||_2^2, \tag{4}$$

$$\mathcal{L}_{bbox} = \sum_{p \in \mathcal{P}} ||B_p - \hat{B}_p||_1, \tag{5}$$

where M is the GT 2D pose heatmap and λ_{bbox} is a hyper-parameter. Note that we just supervise the estimated bounding boxes at the locations of GT root joints $\mathcal{P}$, where the

estimated and GT bounding box at the root joint p are denoted as $\hat{B}_p$ and $B_p \in \mathbb{R}^4$ respectively.

To implement Depth-wise Projection Decay for reducing the depth ambiguity when projecting 2D heatmaps to the 3D space, the backbone predicts a coarse depth map of the root joints by an independent ResNet-18 called Depth Estimator, DE. The primary reason for not predicting depth by adding a head to ResNet-50 is that it would lead to a decrease in the performance of 2D pose estimation. Furthermore, since our method does not heavily rely on the accuracy of depth estimation, a lightweight network such as ResNet-18 suffices. The output of DE is denoted as $\hat{D} \in \mathbb{R}^{H\times W\times 1}$, which is trained by

$$\mathcal{L}_{depth} = \sum_{p\in\mathcal{P}} ||D_p - \hat{D}_p||_1, \tag{6}$$

where D_p and $\hat{D}_p$ is the GT and estimated root depth of the GT root joint p. As above, the depth is also only supervised at the location of the GT root joints $\mathcal{P}$.

3.5. The Human Detection Network

The task of the Human Detection Network is to detect the bounding boxes of people from bird's-eye view. We first construct an aggregated 3D feature volume $F \in \mathbb{R}^{X\times Y\times Z}$ in the range of whole scene (8m×8m×2m) by back-projecting the 2D pose heatmaps into the 3D voxel space and conducting the DPD. In our implementation, X, Y and Z are set to 80, 80 and 20, respectively. Then, we project feature volume onto xy plane by max-pooling along z axis and get a 2D feature $E^{(x,y)}$. Next, $E^{(x,y)}$ is fed into an Encoder-Decoder Network (EDN) introduced in Section 3.3 to estimate the bounding boxes from bird's-eye view.

The output of EDN includes two parts, namely a center map $\hat{C}^{(x,y)} \in \mathbb{R}^{X\times Y\times 1}$ and a size map $\hat{S}^{(x,y)} \in \mathbb{R}^{X\times Y\times 4}$, predicted by two heads. The former encodes the probability that the person's root joints is located at the positions, while the latter encodes the distance between the person's root joints and the four side of the bounding boxes.

The Center Map To supervise the center map $\hat{C}^{(x,y)}$, we generate its GT confidence map $C^{(x,y)}$ by placing Gaussian distributions at the GT locations of people's root joints. Specifically, the 2D confidence value at location (i, j) is computed by

$$C_{i,j}^{(x,y)} = \max_{1\le n\le N} exp\{-\frac{(i-\bar{u}_n)^2 + (j-\bar{v}_n)^2}{2\sigma^2}\}, \tag{7}$$

where $(\bar{u}_n, \bar{v}_n)$ represents the GT position of the n_{th} person's root joint, and N is the number of people. We empirically set σ to 200. The loss function for training the center map is

$$\mathcal{L}_{center} = ||C^{(x,y)} - \hat{C}^{(x,y)}||_2, \tag{8}$$

The Bounding Box Map To train the bounding box estimation, we compute a GT bounding box map according to the GT 3D poses. For a GT root joint on xy plane, the GT size of its bounding box is obtained by expanding the circumscribed rectangle of the 3D pose

on the xy plane by 200 mm. Similar to the backbone, we only supervise bounding box predictions at locations within the root joints' neighborhood. Specifically, we denote the set of all the locations within the neighborhood of the root joints as U, and compute the loss of bounding box estimation by

$$\mathcal{L}_{size} = \frac{1}{N} \sum_{p \in U} ||S_p^{(x,y)} - \hat{S}_p^{(x,y)}||_1, \tag{9}$$

where $S_p^{(x,y)}$ and $\hat{S}_p^{(x,y)}$ are the GT and estimated size map of bounding boxes, and N denote the number of people.

Finally, we pick the bounding boxes with higher confidence scores after NMS and lift them into 3D bounding box proposals by a fixed height, i.e., 2000mm. These proposals will be passed into the Joint Localization Network for estimating the 3D poses.

3.6. The Joint Localization Network

In the Joint Localization Network, we estimate 3D poses from a list of person-specific feature volumes. First, for each body proposal center provided by previous stage, we construct a finer-grained feature volume at the center within a 2m×2m×2m space. The volume is divided into $X' \times Y' \times Z'$ voxels, where $X' = Y' = Z' = 64$. After being processed by DPD, the feature volumes are masked by the corresponding 3D bounding boxes to zero out the voxel features outside the boxes, which is important for avoiding feature overlap between people. Next, we orthographically re-project all the person-specific feature volumes onto the coordinate planes, namely xy, xz and yz planes, by max-pooling. These 2D voxel features will be fed into an EDN to estimate the 2D poses on the corresponding planes, which finally weighted by a lightweight scoring network to recover the 3D pose.

Joint Localization To reduce the quantization error when computing the joint positions, we compute the center of mass on 2D pose heatmaps instead of directly taking the maximum response of the heatmaps. Specifically, we denote the outputs of EDN as $T^{(x,y)} \in \mathbb{R}^{X' \times Y' \times K}$, $T^{(x,z)} \in \mathbb{R}^{X' \times Z' \times K}$, $T^{(y,z)} \in \mathbb{R}^{Y' \times Z' \times K}$ corresponding to the planes respectively, and the estimated positions of joints $\hat{J}^t$ ($t \in \{(x,y),(x,z),(y,z)\}$) is calculated by

$$\hat{J}^{(x,y)} = \sum_{i=1}^{X'} \sum_{j=1}^{Y'} (i,j) \cdot T_{i,j}^{(x,y)}, \tag{10}$$

$$\hat{J}^{(x,z)} = \sum_{i=1}^{X'} \sum_{k=1}^{Z'} (i,k) \cdot T_{i,k}^{(x,z)}, \tag{11}$$

$$\hat{J}^{(y,z)} = \sum_{j=1}^{Y'} \sum_{k=1}^{Z'} (j,k) \cdot T_{j,k}^{(y,z)}, \tag{12}$$

We compute a L_1 loss to train the EDN:

$$\mathcal{L}_{joint} = \sum_{t} \sum_{k=1}^{K} ||J_k^t - \hat{J}_k^t||_1, \tag{13}$$

where J_k^t is the GT heatmaps generated by projecting the GT 3D poses.

Scoring Weight Fusion Despite the 3D pose can be easily obtained by averaging the 2D pose estimations on three coordinate planes, it may introduce more errors due to the imperfect predictions in each plane. Thus, as in [Ye et al. (2022)], we use a lightweight CNN to score every joint's heatmap in $T^{(x,y)}$, $T^{(x,z)}$ and $T^{(y,z)}$. Then we weight every 2D joint location $\hat{J}_k^t$ with normalized scores to compute the 3D pose. The scoring network is trained by

$$\mathcal{L}_{score} = \sum_{k=1}^{K} ||J_k - \tilde{J}_k||_1, \tag{14}$$

where J_k is the GT 3D pose coordinates.

4. Experiments

4.1. Implementation Details

Datasets We adopt three datasets to evaluate our method: Campus [Belagiannis et al. (2014)], Shelf [Belagiannis et al. (2014)] and CMU Panoptic [Joo et al. (2015)]. The Campus dataset captures 3D poses of three people outdoors in three views. The Shelf dataset captures four people surrounding a shelf by five cameras. For Campus and Shelf datasets, we split the training set and validation set following Tu et al. (2020). Due to they are relatively small and their incomplete annotations, we do not use them to train our backbone, instead, only use them to train the HDN and JLN in the corresponding experiments. The CMU Panoptic dataset captures 3D poses of multiple people interacting with each other by numerous cameras. We select five HD cameras and image sequences following Ye et al. (2022).

Metrics As in previous works, we apply the Percentage of Correct Parts (PCP3D) metric to evaluate our method on Campus and Shelf, which only find the closest 3D estimation for each GT 3D pose and do not penalize the false-positive estimation due to their incomplete annotations. For CMU Panoptic, we use the Mean Per Joint Position Error (MPJPE) and the Average Precision (AP_K) metrics for evaluation. If MPJPE of a pose is larger than K mm, it will be regarded as a wrong estimation in AP_K.

Training Details We use GeForce RTX 2080 Ti GPU to train our models and implement all the experiments. We first train the backbone (ResNet-50) on the dataset mixed by COCO [Lin et al. (2014)] and CMU Panoptic [Joo et al. (2015)] for estimating the 2D poses and bounding boxes from views, and we train it for 40 epochs, where the learning rate is 1×10^{-4} and the batch size is 16. And the Depth Estimator (ResNet-18) is trained on CMU Panoptic by generated depth annotations for 20 epochs. For evaluation, we train the HDN and JLN in an end-to-end manner on the dataset corresponding to each experiment, where the batch size is set to 8. For Campus and Shelf, the HDN and JLN are trained for 10 epochs by generated heatmaps as in Ye et al. (2022). For CMU Panoptic, they are trained for 20 epochs.

Method	#Views	AP_{25}	AP_{50}	AP_{100}	AP_{150}	MPJPE
VoxelPose	5	83.59	98.33	99.76	99.91	17.68mm
Faster VoxelPose		85.22	98.08	99.32	99.48	18.26mm
Ours		86.25	98.45	99.77	99.82	17.42mm
VoxelPose	4	78.62	98.03	99.35	99.51	20.01mm
Faster VoxelPose		73.95	97.02	99.21	99.35	21.12mm
Ours		75.92	97.86	99.32	99.40	20.95mm
VoxelPose	3	58.94	93.88	98.45	99.32	24.29mm
Faster VoxelPose		53.68	91.89	97.40	98.30	26.13mm
Ours		57.23	92.21	97.83	98.32	24.98mm
VoxelPose	2	36.83	71.94	93.44	97.21	38.94mm
Faster VoxelPose		19.63	54.94	91.27	94.36	52.23mm
Ours		31.25	65.63	93.51	96.12	42.53mm
VoxelPose	1	0.86	23.47	80.69	93.32	66.95mm
Faster VoxelPose		0.00	1.40	27.84	55.84	133.05mm
Ours		0.44	18.33	64.89	87.95	71.33mm

Table 2: Comparing our method with VoxelPose [Tu et al. (2020)] and Faster VoxelPose [Ye et al. (2022)] (Baseline) in different number of views on CMU Panoptic. Our method largely reduces the gap between the Faster VoxelPose and VoxelPose, especially when the number of cameras is small, demonstrating the effectiveness of DPD.

4.2. Evaluation of DPD

Comparison with Baseline To test the effect of DPD on pose estimation performance, we compare our method with VoxelPose [Tu et al. (2020)] and Faster VoxelPose [Ye et al. (2022)] on CMU Panoptic dataset. Note that VoxelPose is the baseline of Faster VoxelPose, which serves as the baseline of our method. We evaluate three models under different number of views and report the results on Table 2. It reveals that our method significantly bridges the gap between Faster VoxelPose and its base method VoxelPose. Especially in the case of fewer views, our method reduces the MPJPE of Faster VoxelPose by 19% and 46% under two and one views respectively, which implies that our method is more applicable in real-world scenarios with limited budget. In addition, it is worth noting that our method outperforms VoxelPose when the number of cameras is five, which further demonstrates the benefit of our introduction of depth information to the pose estimation network via DPD.

Different Decay Coefficient Since DPD applies a Gaussian decay in the process of the projection, the setting of the coefficient σ (Equation 1) will affect the intensity of DPD. In theory, a larger coefficient σ would bring DPD closer to the original projection method, while a smaller one will do the opposite. Thus, in this experiment we assess the effect of different σ settings and record the AP_K and MPJPE on Table 3. We try three settings of σ, which roughly revolve around the width of one voxel in HDN, i.e., 200mm. The result shows that setting σ to 200mm obtains the best accuracy, while both larger and smaller values lead to performance degradation.

σ	AP_{25}	AP_{50}	AP_{100}	AP_{150}	MPJPE
100mm	84.22	98.03	99.18	99.56	18.03mm
200mm	86.25	98.45	99.77	99.82	17.42mm
300mm	83.58	98.64	99.54	99.59	17.64mm

Table 3: Evaluation of decay coefficient σ in DPD. We attempt three settings of σ on CMU Panoptic to assess its effect to our method and report the result by AP_K and MPJPE. It shows that 200mm is better than other settings.

Method	Inverted Bottleneck	Parallel Decoder	Kernel Size	AP_{25}	AP_{50}	MPJPE
(a) Baseline	✗	✗	3×3	82.66	98.01	18.90mm
(b)	✓	✗	3×3	85.42	97.83	18.44mm
(c)	✓	✓	3×3	84.33	98.36	17.81mm
(d)	✓	✓	5×5	84.26	98.22	17.70mm
(e) Ours	✓	✓	7×7	**86.25**	**98.45**	**17.42mm**
(f)	✓	✓	9×9	85.64	98.21	17.54mm

Table 4: Ablation study on EDN. We evaluate the performance of the model on CMU Panoptic with a couple of model settings. The results demonstrate the effectiveness of our design of adopting inverted bottleneck blocks, parallel decoders, and large convolution kernels.

4.3. Evaluation of EDN

As mentioned in Section 3.3, we have three important designs for EDN: inverted bottleneck, parallel decoders, and large convolution kernels. For revealing their impact on the performance, we conduct an ablation study on EDN, and record the results on Table 4. As illustrated, first, replacing the ResNet block with inverted bottleneck brings benefit to the model, which demonstrate the effectiveness of inverted bottleneck in lightweight networks. Second, our parallel decoder gains an obvious improvement on the performance as expected, which reduces the MPJPE from 18.44mm to 17.81mm. This confirms our conjecture that it is effective to use the multi-branch network approach to fuse multi-scale information on a single-branch network. Third, we try a list of kernel size from 3×3 to 9×9, finding that larger convolution kernel in inverted bottleneck block can increase the receptive field and capacity of the network, and kernel size of 7×7 get the best accuracy for our method.

4.4. Comparison with SOTA

Evaluation on Campus and Shelf We compare our method with SOTA multi-view multi-person 3D pose estimation methods in recent years on Campus and Shelf datasets. In this experiment, we use PCP3D as metrics and the result is recorded in Table 5. On these two

Method	Campus				Shelf			
	Actor 1	Actor 2	Actor 3	Average	Actor 1	Actor 2	Actor 3	Average
Ershadi-Nasab et al. (2018)	94.2	92.9	84.6	90.6	93.3	75.9	94.8	88.0
Dong et al. (2019)	97.6	93.3	98.0	96.3	98.8	94.1	97.8	96.9
Huang et al. (2020)	98.0	94.8	97.4	96.7	98.8	96.2	97.2	97.4
Tu et al. (2020)	97.6	93.8	98.8	96.7	99.3	94.1	97.6	97.0
Lin and Lee (2021)	98.4	93.7	99.0	97.0	99.3	96.5	98.0	97.9
(Faster VoxelPose)Ye et al. (2022)	96.5	94.1	97.9	96.2	99.4	96.0	97.5	97.6
Ours	97.4	93.6	98.1	96.4	99.0	96.3	97.7	97.7

Table 5: Comparison with SOTA methods on Campus and Shelf. Taking PCP3D as metrics, our method gets competitive performance with SOTA methods. Especially, we outperform the baseline of our method, Faster VoxelPose on both datasets.

Method	AP_{25}	AP_{50}	AP_{100}	AP_{150}	MPJPE	Time	FPS
Tu et al. (2020)	83.59	98.33	99.76	99.91	17.68mm	290.1ms	3.5
Lin and Lee (2021)	92.12	98.96	99.81	99.84	16.75mm	107.6ms	9.4
wang et al. (2021)	92.28	96.6	97.45	97.69	15.76mm	170.6ms	5.9
(Faster VoxelPose)Ye et al. (2022)	85.22	98.08	99.32	99.48	18.26mm	29.6ms	33.9
Ours	86.25	98.45	99.77	99.82	17.42mm	33.3ms	30.5

Table 6: Comparison with SOTA methods on CMU Panoptic. We measure the average per-sample inference time on 2080-Ti GPU to compare the efficiency. The results show that our method get competitive performance with a real-time speed.

small datasets, we achieve competitive accuracy with the SOTA methods, and we outperform Faster VoxelPose on both datasets.

Evaluation on CMU Panoptic We carry out comparison between our method and SOTA multi-view multi-person 3D pose estimation methods on CMU Panoptic. And the average per-sample inference time is measured to evaluate the speed of the models, where all of them run on a single 2080-Ti GPU and batch size is 1. The result is illustrated in Table 6. It shows that our method is competitive with SOTA methods in terms of accuracy. On the one hand, our method outperforms two other voxel-based works, VoxelPose [Tu et al. (2020)] and Faster VoxelPose [Ye et al. (2022)], indicating that the superiority of our proposed method for constructing and manipulating voxel features, namely DPD and EDN. On the other hand, compared with other CNN-based methods, Lin and Lee (2021) and wang et al. (2021), the gap between our method and them is smaller than 2mm, which is acceptable in application in real world, while our model runs much faster than them.

4.5. Qualitative Study

We report the qualitative result on CMU Panoptic in Figure 4, which shows 3D pose estimation in the first column and 2D pose estimations in three coordinate planes in the

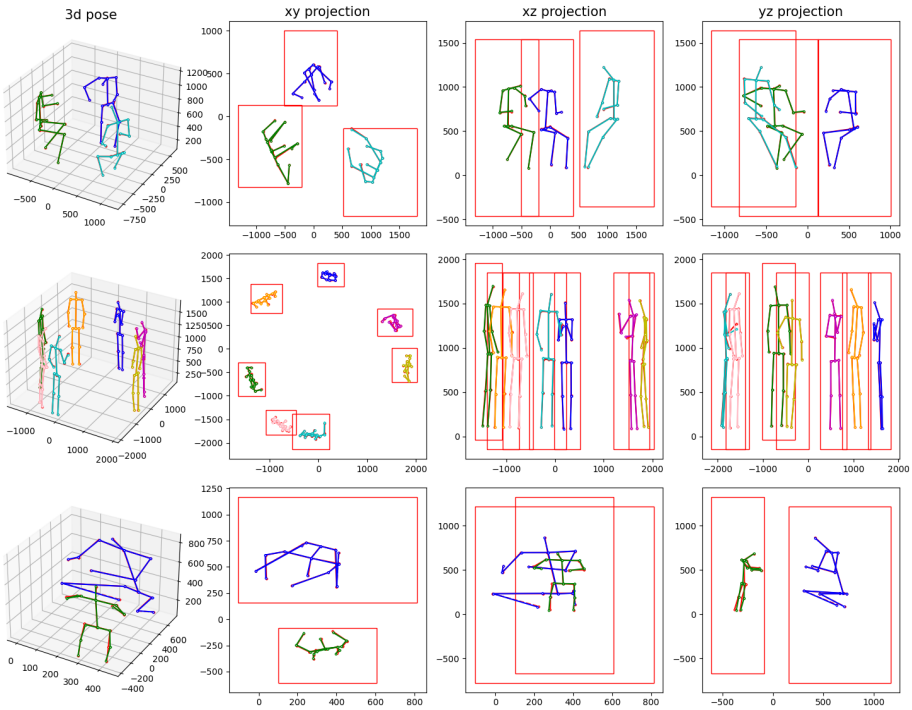

Figure 4: The qualitative result of our method on CMU Panoptic. The first row shows 3D pose estimation, and the next three column shows 2D pose estimation in three coordinate planes, respectively. Each row shows a sample in test set.

next three columns, where the GT annotations are drawn with red. From the second column, we can find that our method precisely locates every person in the scene from the bird's-eye view and prevent the overlap of features as much as possible. As illustrated in the first and second rows, our method can easily handle the case of multiple people standing or sitting, especially the crowd scene shown in the second row indicates the robustness of our method to occlusion between people. More importantly, the poses of the kid and the adult in the third row are both estimated accurately, which put the evidence for our method's capacity to deal with bodies in different scales, i.e., different ages and heights.

5. Conclusion

In this paper, we propose a real-time voxel-based method for multi-person multi-view 3D pose estimation. Our main contributions focus on two points. First, we propose a new projection method for constructing feature volume, namely the Depth-wise Projection Decay, which largely improves the accuracy of model in the case of fewer cameras. Second, we design a powerful yet efficient Encoder-Decoder Network to process the 2D features, considerably improving the performance of the model. The experiments prove that our model have a good trade-off between accuracy and speed compared with SOTA methods.

References

Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Simon Bultmann and Sven Behnke. Real-time multi-view 3d human pose estimation using semantic feedback to smart edge sensors. *arXiv preprint arXiv:2106.14729*, 2021.

Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Sara Ershadi-Nasab, Erfan Noury, Shohreh Kasaei, and Esmaeil Sanaei. Multiple human 3d pose estimation from multiview images. *Multimedia Tools and Applications*, 77:15573–15601, 2018.

Matteo Fabbri, Fabio Lanzi, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Compressed volumetric heatmaps for multi-person 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Congzhentao Huang, Shuai Jiang, Yang Li, Ziyue Zhang, Jason Traish, Chen Deng, Sam Ferguson, and Richard Yi Da Xu. End-to-end dynamic matching network for multi-view multi-person 3d pose estimation. In *Computer Vision – ECCV 2020*, pages 477–493, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58604-1.

Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Jiahao Lin and Gim Hee Lee. Multi-view multi-person 3d pose estimation with plane sweep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11886–11895, June 2021.

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*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, June 2022.

Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7904–7913, 2019. doi: 10.1109/CVPR.2019.00810.

Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *Computer Vision – ECCV 2020*, pages 197–212, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3349–3364, 2021. doi: 10.1109/TPAMI.2020.2983686.

tao wang, Jianfeng Zhang, Yujun Cai, Shuicheng Yan, and Jiashi Feng. Direct multi-view multi-person 3d pose estimation. In *Advances in Neural Information Processing Systems*, volume 34, pages 13153–13164. Curran Associates, Inc., 2021.

Yihan Wang, Muyang Li, Han Cai, Wei-Ming Chen, and Song Han. Lite pose: Efficient architecture design for 2d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13126–13136, June 2022.

Hang Ye, Wentao Zhu, Chunyu Wang, Rujie Wu, and Yizhou Wang. Faster voxelpose: Real-time 3d human pose estimation by orthographic projection. In *Computer Vision – ECCV 2022*, pages 142–159, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20068-7.

Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenyu Liu, and Wenjun Zeng. Voxeltrack: Multi-person 3d human pose estimation and tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2613–2626, 2023. doi: 10.1109/TPAMI.2022.3163709.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. URL http://arxiv.org/abs/1904.07850.