Sequential 3D Human Pose Estimation Using Adaptive Point Cloud Sampling Strategy

Zihao Zhang[1,2†], **Lei Hu**[1,2†], **Xiaoming Deng**[3†] and **Shihong Xia**[1,2*]

[1]Institute of Computing Technology, Chinese Academy of Sciences

[2]University of Chinese Academy of Sciences, [3]Institute of Software, Chinese Academy of Sciences

{zhangzihao, hulei19z, xsh}@ict.ac.cn, xiaoming@iscas.ac.cn

Abstract

3D human pose estimation is a fundamental problem in artificial intelligence, and it has wide applications in AR/VR, HCI and robotics. However, human pose estimation from point clouds still suffers from noisy points and estimated jittery artifacts because of handcrafted-based point cloud sampling and single-frame-based estimation strategies. In this paper, we present a new perspective on the 3D human pose estimation method from point cloud sequences. To sample effective point clouds from input, we design a differentiable point cloud sampling method built on density-guided attention mechanism. To avoid the jitter caused by previous 3D human pose estimation problems, we adopt temporal information to obtain more stable results. Experiments on the ITOP dataset and the NTU-RGBD dataset demonstrate that all of our contributed components are effective, and our method can achieve state-of-the-art performance.

1 Introduction

3D human pose estimation from point clouds has been a fundamental research field in recent years, and it can be applied to many applications such as human-computer interaction, motion retargeting and virtual avatar control. Regarding the input of 3D human pose estimation, depth maps or point clouds are often preferable. First, point clouds contain 3D spatial information of humans, which can make the estimated human pose to be scale correct. Second, the point clouds' quality is generally invariant under the ambient light changes, which leads to more potential application scenarios such as indoor augmented reality. Finally, depth sensors are widely available in cellphones and tablet PCs, which require robust algorithms to utilize these depth sensors.

Although great progress has been made in the field of 3D human pose estimation, there are still several challenges. First, the noisy point clouds from depth cameras may cause difficulties in learning a proper human pose model. Second, the 3D human pose estimation task is difficult due to the ambiguity caused by occlusion and self-occlusion. Third, existing point cloud-based methods mainly focus on pose estimation from a single frame. However, due to the lack of temporal smoothness enforcement, current methods may generate results with shaking artifacts on continuous point cloud sequences.

In this paper, we propose a 3D human pose estimation method from point cloud sequences. Inspired by the point cloud-based framework from a single frame [Zhang *et al.*, 2020], we design a new two-stage human pose estimation pipeline using point cloud sequences. The point cloud sampling module is used to select effective point clouds in an adaptive manner that can be informative to the human pose estimation task. To address noisy point cloud issues, we observe that our point sampling strategy can enhance the quality of the input point clouds. Therefore, we first estimate the point cloud sampling centers based on a density-guided attention mechanism, and use these center points to sample the pose-aware point clouds. To address jittery artifacts and occlusion problems, we use temporal consistency to constrain the results, thereby generating accurate human pose results. Experiments demonstrate that our method can achieve state-of-the-art performance efficiently on ITOP and NTU-RGBD datasets.

The main contributions of our work can be summarized as follows:

1. We propose a density-guided and attention-based differentiable point cloud sampling method. The sampling method can help us select the point clouds in the human foreground and provide more effective point cloud input for human pose estimation.

2. We propose a 3D human pose estimation method for point clouds using temporal sequences. Compared to existing human pose estimation methods, our method can achieve better and smoother human pose results.

3. Based on our method, we build a real-time human capture system that can enable smooth human motion capture. Experiments demonstrate that our method can achieve state-of-the-art performance in both accuracy and efficiency.

† indicates equal contributions. * indicates corresponding author. supplementary material https://github.com/Hmslab/Adapose

2 Related Works

2.1 Human Pose Estimation

In the following part, we review related state-of-the-art methods that use a single image as input and several representative 3D human pose estimation methods that leverage temporal information.

The State-of-the-Art Methods. For 3D human pose estimation from depth maps, previous methods mainly use statistic models to estimate the human pose from the depth maps [Ding and Fan, 2014]. Recently, the state-of-the-art methods have mainly been based on representations of depth maps [Moon *et al.*, 2018; Zimmermann *et al.*, 2018; Zhang *et al.*, 2020]. The method [Moon *et al.*, 2018] treats depth maps as point clouds and converts them into 3D voxel grids; a 3D CNN is then used to estimate the 3D human poses. However, this method requires prior extraction of the background points. Similarly, the work [Zimmermann *et al.*, 2018] first uses an RGB image to voxelize the point clouds and uses a Res2Net-like [Gao *et al.*, 2019] network to estimate the 3D human pose. Recently, Zhang *et al.* [Zhang *et al.*, 2020] have proposed the use of a hybrid 2D/3D representation of depth maps and a generative-like method.

Methods using Videos as Input. The 3D human pose estimation methods using videos can be classified into two categories. The first type uses the temporal information afterwards and uses it to smooth the estimated results. The work [Lin *et al.*, 2017] proposes a multistage sequential refinement network to estimate 3D human pose sequences. Dabral *et al.* [Dabral *et al.*, 2017] use a fully connected network to refine coarse input poses. In the work [Rayat Imtiaz Hossain and Little, 2018], the authors use temporal coherent 2D poses to estimate a sequence of 3D poses. The second type uses the temporal information and extracts the temporal-related features from the sequence. Kanazawa *et al.* [Kanazawa *et al.*, 2019] design a semisupervised pipeline to learn the 3D human dynamics from videos. In [Pavllo *et al.*, 2019], the authors propose a fully convolutional architecture that utilizes temporal convolution to estimate 3D human poses in videos. In [Arnab *et al.*, 2019], the authors first estimate the 2D joint position and the SMPL model parameters for each frame and use the bundle adjustment to estimate smooth results. In [Bertasius *et al.*, 2019], the authors focus on learning features from both the labeled frame and the unlabeld frame to perform dense temporal pose propagation and estimation. Recently, Liu and colleagues [Liu *et al.*, 2021] propose a human pose estimation method for a multiframe scenario, in which they leverage the temporal information between video frames to facilitate keypoint detection.

2.2 Deep Learning on Point Clouds

Recent 3D deep learning research on processing 3D objects, such as point clouds and mesh models, can be roughly divided based on the representation of the 3D object. In the following part, we only review several point cloud-based 3D deep learning methods due to the efficiency and excellent performance of point cloud representations.

The point cloud-based methods mainly use the point clouds as input and can extract features from the input point cloud coordinates and other information such as the surface normal. These methods were originally designed for point cloud segmentation or classification tasks [Qi *et al.*, 2017a; Qi *et al.*, 2017b]. Recently, there have also been works using point cloud learning methods for object detection tasks [Zhou and Tuzel, 2018; Qi *et al.*, 2019]. Qi *et al.* [Qi *et al.*, 2017a] propose an end-to-end network named PointNet, which uses the point coordinates and the surface normal as inputs and maps them to a higher-dimensional space using a multilayer perceptron. In the work [Qi *et al.*, 2017b], the authors further use a partition-sampling module and feed the output recursively to this module. On the other hand, they also propose to use the 2D information to accelerate the 3D detection in a hybrid camera [Qi *et al.*, 2018]. Liu and colleagues [Liu *et al.*, 2019b] propose a new method that uses the points to represent the point clouds while performing the convolution on the voxel-based representation. Recently, Lang and colleagues propose a differentiable point cloud sampling method [Lang *et al.*, 2020] that predicts the sampling centers and achieves state-of-the-art performance.

The differences between our method and the other methods are twofold. First, we propose the first 3D human pose estimation framework from point cloud sequences, and we propose pose consistency loss to smooth the pose estimation results. Second, we introduce an attention mechanism into the sampling phase, which improves the performance of the 3D pose estimation task.

3 Method

We use the two-stage pipeline for human pose estimation. As can be seen from Fig. 1, our pose estimation method can be roughly divided into two parts, the density-based point cloud sampling module and the sequential 3D human pose estimation module. In the point cloud sampling module, we present a differentiable point cloud sampling using a density-guided attention mechanism. In the 3D human pose estimation module, we estimate 3D human poses from the sampled point cloud sequences. The point sampling network and 3D human pose estimation network are trained in an end-to-end manner to obtain optimal human pose estimation performance.

3.1 Point Cloud Sampling Module

This module mainly aims to sample an effective subset from the input point clouds, which can benefit the human pose estimation task. The original depth maps contain redundant and noisy pixels, which may increase the computational cost and reduce the human pose estimation accuracy. For example, if the background pixels or the noise pixels are mixed into the point clouds for human pose estimation, then the human pose estimation can often exhibit significant estimation errors. Inspired by [Lang *et al.*, 2020], we resolve this problem by designing a new differentiable sampling strategy with a density-guided attention mechanism. The input of our point cloud sampling module is a depth map that contains persons and backgrounds, and the output is the pose-aware point cloud.

Our density-guided and attention-based point cloud sampling strategy has two steps. First, we aim to generate a set of sampling centers R with input point clouds P such that the

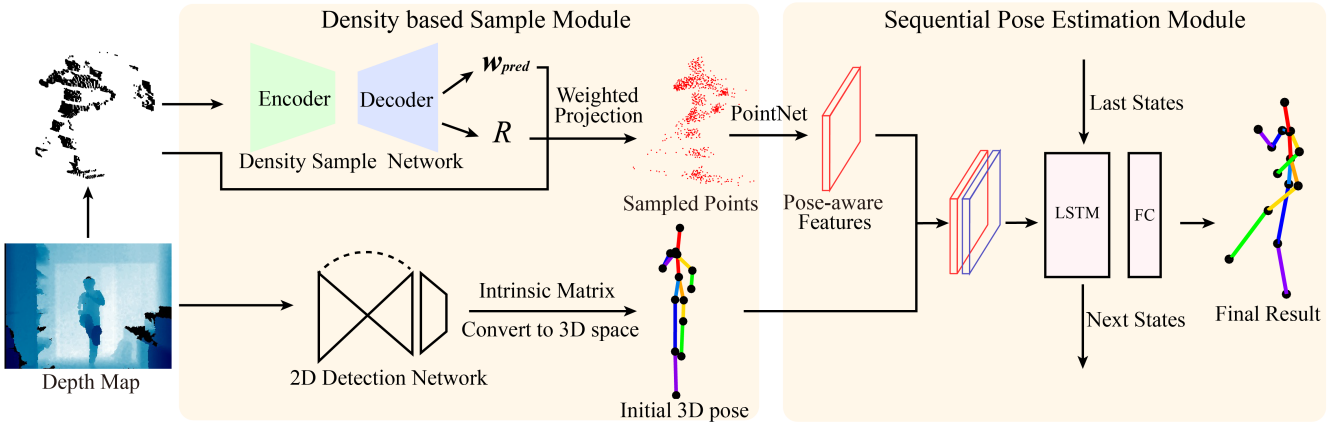


Figure 1: Overall pipeline of our 3D human pose estimation method. The network consists of two modules. The density-based point cloud sampling module obtains the downsampled pose-aware point clouds and the original point cloud weights. The sequential pose estimation module extracts the spatiotemporal features of the pose-aware point cloud sequences and learns 3D human pose estimation.

neighboring points of R in the original point clouds P perform better in the human pose estimation task. Second, we sample the point clouds with the guidance of the sampling centers R, predicted weights w_{pred} and original point clouds. The process can be seen in the upper left of Fig. 1.

Sampling Center Generation. The aim of sampling center generation is to obtain a subset of point clouds to serve as the sampling centers. We expect that the neighboring point clouds of sampling centers in the input point clouds are in the human foreground instead of background points. To obtain effective point cloud sampling centers, we take the relationship between the sampling center points and their neighborhood into consideration, and design a density-guided attention mechanism to adaptively generate the sampling results.

The key idea is built upon two definitions, *core points* and *boundary points*. As shown in Fig. 2, the core points are often the points inside the human surface, and the boundary points often belong to the human boundary. The points that are neither the core points nor the boundary points are usually less important or noisy because these points are sparse in 3D space and contain little human pose-aware information.

A point r is called a core point unless there are M points in the neighborhood of r within the distance of ϵ. This is shown as the red solid circles in the middle of Fig. 2. The red solid circles always have M original points (in the figure, we use $M = 1$) within the given radius of ϵ (shown as the circle with the dashed line.). The boundary points are the points that have less than M points but more than one core point in the distance threshold ϵ. The boundary points are shown as green solid circles in Fig. 2. The core point and boundary point can be formulated as follows: A point r is a core point if $|U^o(r, \epsilon)| \geq M$, where $|U^o(r, \epsilon)|$ is the number of points within a sphere of radius ϵ and a center of sampled point r. A point r is a boundary point if there exists $r' \in U^o(r, \epsilon)$ such that $I_C(r') = 1$. The definition of the indicator function I_C that determines whether point r belongs to the core points C

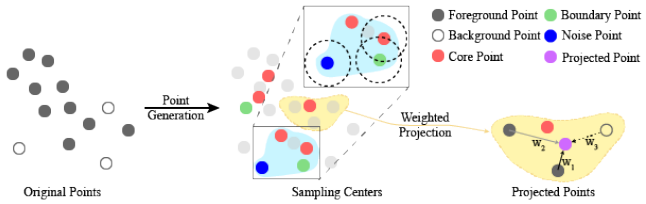


Figure 2: Illustration of our point cloud sampling module. Using the original points as input (shown as solid and hollow gray circles in the left), we first generate a set of intermediate sampling centers (shown in red, green and gray colors) and then use the predicted weight (Eq.(3)) with the sample network to obtain the final sampling center (shown in purple) with Eq.(2).

is given as follows:

$$I_C(r) = \begin{cases} 1, & if\ r\ is\ a\ core\ point \\ 0, & if\ r\ is\ not\ a\ core\ point \end{cases} \quad (1)$$

Point Cloud Sampling. The density-based sampling strategy can generate compact sampling point centers from the original point clouds, but it overlooks the human body context and may still obtain points on background points with high density. Next, we present a point cloud sampling method to adaptively select the human-aware point clouds.

As shown in the left of Fig. 2, the original point $p_i \in P$ has a ground truth binary label w_i^{gt} based on whether they are background points. Our sample network will simultaneously predict original point clouds' weights w^p while generating sampling centers.

The sampling centers are just an approximate subset of original point clouds P. Therefore, to obtain the final sampled point clouds, we use the soft projection operation, similar to the work [Lang *et al.*, 2020]. The soft projection operation is shown in the right of Fig. 2. In the training process, the projected point $r^* \in R^*$ is obtained by:

$$r^* = \sum_{i \in \mathcal{N}_P(r)} \omega_i p_i \quad (2)$$

where r is the generated point in R, $\mathcal{N}_P(r)$ represents the k-nearest neighbor of point r in P, and r^* is the projected generated point cloud set for the pose estimation module. Different to [Lang *et al.*, 2020] that uses only the distance as the weight parameter, our parameter ω_i is defined by the predicted weight w_j together with the distance between r and its neighbors as follows:

$$\omega_i = \frac{w_i^p e^{-d_i^2/t^2}}{\sum_{j \in \mathcal{N}_P(r)} w_j^p e^{-d_j^2/t^2}} \tag{3}$$

where d_i represents the Euclidean distance between point r and neighbor p_i. If w_i^p is close to zero (shown as the dash arrow in Fig. 2), which means the point i is a noisy point, it will get smaller ω even if it is closer than the other neighborhood points in Euclidean distance. The parameter t will convergence to zero as the training time increases. During the testing stage, for each $r \in R$ we directly sample the nearest neighbor from the original point clouds P as the projected point with the weighted distance.

Loss Function. We present a density loss function to obtain the generated sampling centers to be the core points or boundary points. Following [Lang *et al.*, 2020], we introduce the distance loss function to constrain the generated points to have sufficiently large coverage. We also use a mask loss to constrain the generated points to be human foreground points.

Density Loss. The density loss function aims to minimize the generated noisy points' distance to the density cluster in the space. The density loss function can be defined as follows:

$$L_{den} = \frac{1}{|R|} \sum_{i=1}^{|R|} (1 - I_C(r_i))(1 - I_B(r_i)) D(r_i) \tag{4}$$

where $I_B(r)$ is the indicator function showing whether point r belongs to the boundary point B, the definition is similar to that in equation 1, $D(r)$ is the L2 distance function between point r and its nearest core point $C(r)$, which is activated if the point r is a noisy point. The optimal situation is that the generated points are either core points or boundary points, so that the value of $(1 - I_C(r))(1 - I_B(r))$ is zero.

Distance Loss. Inspired by [Lang *et al.*, 2020], we also use the average nearest neighbor loss(the Hausdorff distance) L_H and the maximal nearest neighbor loss(the Chamfer distance) L_C to enforce the generated sampling centers R to be close to the original point clouds P

$$\begin{aligned} L_H(R, P) &= \frac{1}{|R|} \sum_{r \in R} \min_{p \in P} \|r - p\|_2^2 \\ L_C(R, P) &= \max_{r \in R} \min_{p \in P} \|r - p\|_2^2 \end{aligned} \tag{5}$$

where r and p are two points in point cloud sets R and P.

Mask Loss. Only using the density or the distance as the constraints of the network may not be sufficient since the background may also satisfy such constraints. To make our network sample the human foreground points, we use the mask loss using the cross-entropy loss function as follows:

$$L_{mask} = \frac{1}{|P|} \sum_{j=1}^{|P|} -w_j^{gt} \log(w_j^p) - (1 - w_j^{gt}) \log(1 - w_j^p) \tag{6}$$

After we have the definitions of all the loss terms, we can give the overall loss function of our point cloud sampling module as follows:

$$L_{sample} = \alpha_1 L_{den} + \alpha_2 L_H + \alpha_3 L_C + \alpha_4 L_{mask} \tag{7}$$

where the constant α_i is the loss weight. In our experiment, we set $\{\alpha_i\}_{i=1}^4$ to 1.

3.2 3D Human Pose Estimation Module

The pose encoding stage aims to encode the pose-aware point clouds to learn 3D human poses. The right part of Fig. 1 shows our prediction network. We first feed the sampled point clouds of each frame into PointNet to extract pose-aware features. Then, we concatenate them with the initial poses and use a long short-term memory (LSTM) network to model the temporal correlation and estimate 3D human poses.

Pose-aware Feature Extraction. Similar to the work [Zhang *et al.*, 2020], we treat the joint offset between the initial pose and the final pose as our regression target so that we can easily encode the temporal sequence and use a weakly supervised manner for the sequential data.

In our network, we extract the features frame by frame. For frame t, we feed the normalized point clouds p_{norm} to PointNet to extract the pose-aware features. PointNet is a deep neural network that focuses on solving the point cloud-related tasks. The PointNet takes n points as input and applies a 3D transformation to the points. Then, a multilayer perceptron is used to extract the per-joint features, and the max pooling is used to aggregate the features. In our problem, we directly use the three-layer perceptron to extract the human-aware features. The sizes of the layers are set as $64, 128, 1024$ separately. The reason for using PointNet is that the input point clouds are already pose-aware, which means they only contain local information. Moreover, the model size using PointNet is much smaller than that using PointNet++, which enables it to be used in more potential applications.

Long Short-Term Memory Module. After we obtain the pose-aware features, we add a recurrent connection between the features of the neighboring frames. Since we focus on pose estimation, we use the sequential data before frame t to ensure the consistency $f_{estimate} : (\phi_0, ...\phi_t) \mapsto q_t$. As illustrated in the left lower corner of Fig. 1, we first calculate the initial 3D pose by backprojecting the estimated 2D joints $q_{2d} = \{m_i, n_i\}, i = 1, .., J$ into 3D space using the intrinsic matrix of the depth maps. Then, the point cloud features of consecutive frames extracted by PointNet are concatenated with the initial pose and fed into the LSTM module. For the first LSTM stage, we use zero vectors as the initial states. The stage size is set to 256 in our network. To predict the final 3D pose, we adopt a fully connected layer to map the out states of the LSTM module to the offset of each joint and add it to the initial pose to get the final result.

3.3 Loss function

We use both fully labeled data ("valid data" in the ITOP dataset) and weakly labeled data ("invalid data" in the ITOP dataset) for training our model. For the fully labeled data, *i.e.*

the data with 3D pose labels, we use 3D joint loss L_{3D} to enforce the poses generated by our network to be consistent with the ground truth 3D poses. We also use 2D joint loss L_{2D} to enforce the projected 2D poses of the generated 3D poses to be close to the ground truth 2D poses. For the weakly labeled data, we use only 2D joint loss L_{2D} to enforce the projected 2D poses of the generated 3D poses to be consistent with the ground truth 2D poses. In addition to these pose constraints of a single frame, we also use consistency loss L_{con} to make the generated motion sequence continuous and smooth. Since 3D human pose estimation is a downstream task of the point sampling, we also use the point cloud sampling loss L_{sample} defined in Eq. (7).

The total loss function of our network is given as follows:

$$L = \mathbf{I}\lambda_{3D}L_{3D}+\lambda_{2D}L_{2D}+\lambda_{con}L_{con}+\lambda_{sample}L_{sample} \quad (8)$$

where $\mathbf{I}$ is an indicator function to activate the 3D joint loss term L_{3D}, and the constants $\lambda_{3D}, \lambda_{2D}, \lambda_{con}$, and λ_{sample} are loss weights.

3D Joint Loss. The 3D joint loss L_{3D} is computed with the Euclidean distance between the estimated joint positions and the ground truth joint positions:

$$L_{3D} = \|q^* - (q_{init} + \Delta q)\|^2 \quad (9)$$

where q^* is the ground truth 3D human pose, q_{init} is the predicted initial pose, and Δq is the predicted offset between the initial pose and the ground truth pose.

2D Joint Loss. The 2D joint loss L_{2D} is computed with the Euclidean distance between the 2D projection of the predicted joint positions and the ground truth 2D joint positions:

$$L_{2D} = \|q^*_{2D} - q_{2D}\|^2 \quad (10)$$

where q^*_{2D} is the ground truth 2D joint locations, $q_{2D} = \mathbf{K}(q_{init} + \Delta q)$ is the estimated 2D joint locations and $\mathbf{K}$ is the intrinsic matrix of the depth camera.

Consistency Loss. The key idea of consistency loss is that we assume that the velocity of the joint movement should remain constant in a short time period. Therefore, we compute consistency loss L_{con} with the difference of the first-order derivative between consecutive frames as follows:

$$L_{con} = \sum_{t=3}^{T} \|(q_t - q_{t-1}) - (q_{t-1} - q_{t-2})\|^2 \quad (11)$$

The constant velocity assumption in the consistent loss is only valid for a short time period, so we use both 3D joint loss and consistent loss to balance the estimation accuracy and motion smoothness.

3.4 Implementation Details

During the training process, we use Adam optimizer with a learning rate of 0.0005 which is set to decay 0.05% every 1000 iterations. The bounding box size L is [1.8, 2, 1.5]. In our experiments, we set the weights λ_{3D}, λ_{2D}, λ_{consis} and λ_{sam} as 10, 0.1, 1e-3 and 1. In the point cloud sampling module, we choose $\epsilon = 0.025$, $M = 4$ in the sampling center generation step and 8-nearest neighbors in the projection step.

Experiment	Detail	Result(mAP/MJE)
Baseline	our full method	**93.38/3.97**
Sampling Strategy-1	w/ sample + Stage 2	91.96/4.35
	w/o sample + Stage 2	89.59/5.51
Sampling Strategy-2	FPS-based sampling	90.58/4.98
	original SampleNet	92.29/4.33
Supervision	w/o weak supervision	92.53/4.14
	1/2 fully labeled data	92.48/4.21
	1/3 fully labeled data	92.28/4.33

Table 1: Ablation study results. We show the mAP (%) with the 10 cm error threshold and the mean joint error (mJErr) (cm).

In density-based sampling module, the original point clouds are fed into five 1D convolution layers, each of which is followed by a ReLU activation layer. The output dimensions of the five convolution layers are 64, 128, 256, 512, and 128, respectively. Then we use a fully connected layer that has 512 neurons to generate the sampling centers and the weights of original point clouds.

4 Experiments

In this section, we first provide the implementation details and introduce the datasets and evaluation metrics we use. Then we will provide systematic evaluation results of our method with ablation study and comparisons with state-of-the-art methods.

4.1 Datasets and Evaluation Metrics

In our experiment, we use the ITOP dataset [Haque *et al.*, 2016] and NTU-RGBD dataset [Shahroudy *et al.*, 2016; Liu *et al.*, 2019a] to evaluate our method. The ITOP dataset is built for the 3D human pose estimation problem from depth maps, and the NTU-RGBD dataset is built mainly for action recognition problems. To evaluate the performance of our human pose estimation method, we follow the work [Zhang *et al.*, 2020] and use two types of evaluation metrics. The first is the overall precision of the pose estimation methods, and it includes the percentage of correct keypoints (PCK) and mean average precision (mAP). The PCK value means the percentage of detected keypoints in a given threshold, and mAP is the mean PCK over all joints. The threshold we use in this set of comparisons is 10 cm. The second type is the mean joint error (MJE), which is the average error between the estimated results and the ground truth.

4.2 Ablation Study and Self-Comparison

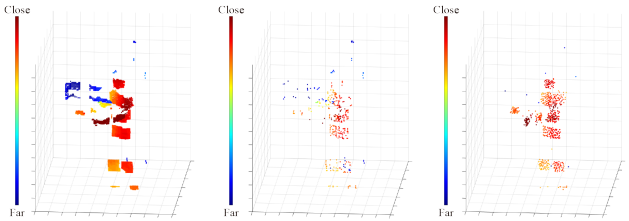
To investigate the effect of different components of our network, we conduct an ablation study of our model. We use the ITOP dataset for the following experiments. The results are shown in Table 1.

Effect of the Weak Supervision. We evaluate the effect of the weak supervision by comparing the mAP value of models trained with fully labeled data and models trained with fully labeled data plus weakly labeled data. As illustrated in Table 1, the mAP values are 92.53% and 93.38% on the ITOP test dataset. To further demonstrate the importance of weak supervision, we train the model with a fixed amount of weakly

	mAP (ITOP)				mAP (NTU)	
Body part	V2V	WSM	V3D	Ours	WSM	Ours
Head	98.25	98.15	**99.61**	98.42	84.54	**89.34**
Neck	98.8	**99.47**	98.93	98.67	**92.63**	91.80
Spine	-	-	-	-	**93.98**	92.83
Mid-spine	-	-	-	-	**96.59**	95.33
Shoulders	98.25	94.69	**98.31**	95.39	81.66	**92.00**
Elbows	78.73	82.80	73.21	**90.74**	66.75	**84.08**
Wrists	-	-	-	-	64.13	**72.73**
Hands	67.21	69.10	59.20	**82.15**	64.40	**68.83**
Thumbs	-	-	-	-	64.75	**65.46**
Hands-tip	-	-	-	-	**61**.62	63.21
Torso	98.29	99.67	92.80	**99.71**	**97.94**	95.19
Hips	90.25	95.71	77.24	**96.43**	**96.79**	94.69
Knee	91.68	91.00	73.16	**94.41**	77.19	**87.56**
Ankles	-	-	-	-	63.06	**83.44**
Feet	85.87	89.96	53.41	**92.84**	58.98	**76.21**
Mean	87.69	89.59	77.17	**93.38**	74.57	**81.64**

Table 2: Comparison of joint mAP with other methods(V2V:[Moon *et al.*, 2018], WSM:[Zhang *et al.*, 2020], V3D:[Pavllo *et al.*, 2019]).

labeled data and different proportions of fully labeled data. As illustrated in Table 1, a sharp decrease of the fully labeled data proportion leads to only a slight decline.

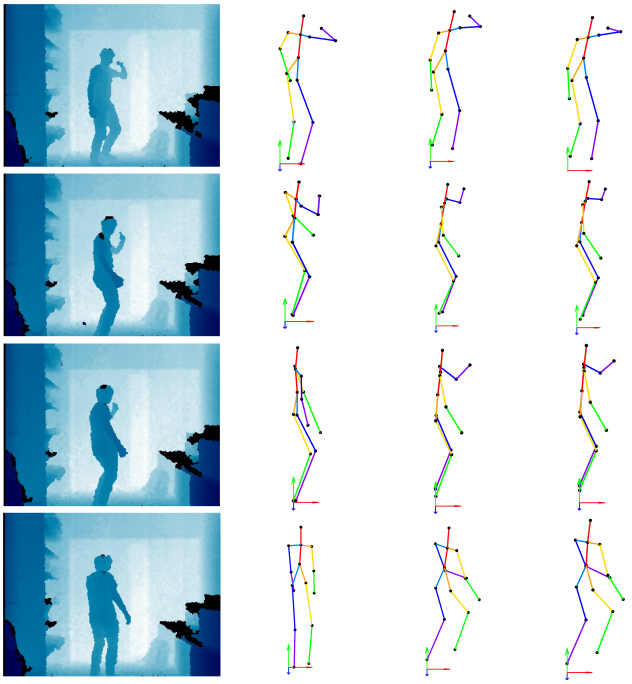


(a) (b) (c)

Figure 3: Qualitative results of our point cloud sampling strategy. From left to right, we show the point cloud sampled by the traditional FPS method, by SampleNet and by our density-guided and attention-based differentiable sampling method.

Effect of Point Sampling Strategy. To investigate the effectiveness of the sampling strategy, we conduct a comparison by replacing the LSTM in our method with one-frame regression. Specifically, we use Stage 1 of our method with the density-guided point sampling strategy and Stage 2 of WSM to train the model. We then compare it with the model trained with Stage 1 without any point sampling strategy and Stage 2 of WSM. Table 1 shows the comparison results with and without point sampling using one-frame pose regression (See "Sampling Strategy-1"). We can observe that our sampling strategy is effective.

As shown in "Sampling Strategy-2" of Table 1, we conduct experiments based on different sampling strategies (SampleNet and FPS-based sampling [Zhang *et al.*, 2020]). We keep the remaining parts of the architecture fixed and only replace the sampling strategy. The mAP value with our sampling strategy in our full model (93.38) is 1.09 and 2.80 percentage points higher than that with the SampleNet (92.29) and FPS-based strategies (90.58), respectively. Fig. 3 illustrates qualitative results of our point cloud sampling strategy. We observe that the noisy points with our method are almost removed. Intuitively, these results can be interpreted by the fact that the point clouds of the human body can be properly selected based on whether they can form a cluster in 3D space rather than distance-based sampling methods such as FPS-based sampling and SampleNet.



(a) (b) (c) (d)

Figure 4: Comparison of our method with WSM. We show the estimation results in a consecutive sequence showing a people turning around.(a) Depth images, (b) results of WSM, (c) our results, (d) ground truth. The left and right arms of WSM are swapped starting from the second row.

4.3 Comparison with the State-of-the-Art Methods

We compare the performance of our methods on the ITOP and NTU-RGBD datasets with other state-of-the-art methods, namely, V2V-PoseNet [Moon *et al.*, 2018](V2V for short), the weakly supervised adversarial learning methods (WSM for short) [Zhang *et al.*, 2020] and the purely RGB-based 3D pose estimation method VideoPose3D (V3D for short) [Pavllo *et al.*, 2019]. We conduct qualitative comparison and quantitative comparisons.

Table 2 shows quantitative comparison results. Among the compared methods, V2V first converts depth maps to volume pixels and then uses a neural network to regress 3D poses. WSM and our method both use point clouds as input. V3D is also a two-stage method (first, it calculates 2D joints; then, it uses them to estimate 3D joints). For the ITOP dataset, the mAP value for our method is 3.79 percentage points higher than that of WSM [Zhang *et al.*, 2020](see Table 2). The mean joint errors of our method are 3.56 cm and 2.72 cm lower than those of V2V-PoseNet and WSM, respectively. These results show that our method performs better than ex-

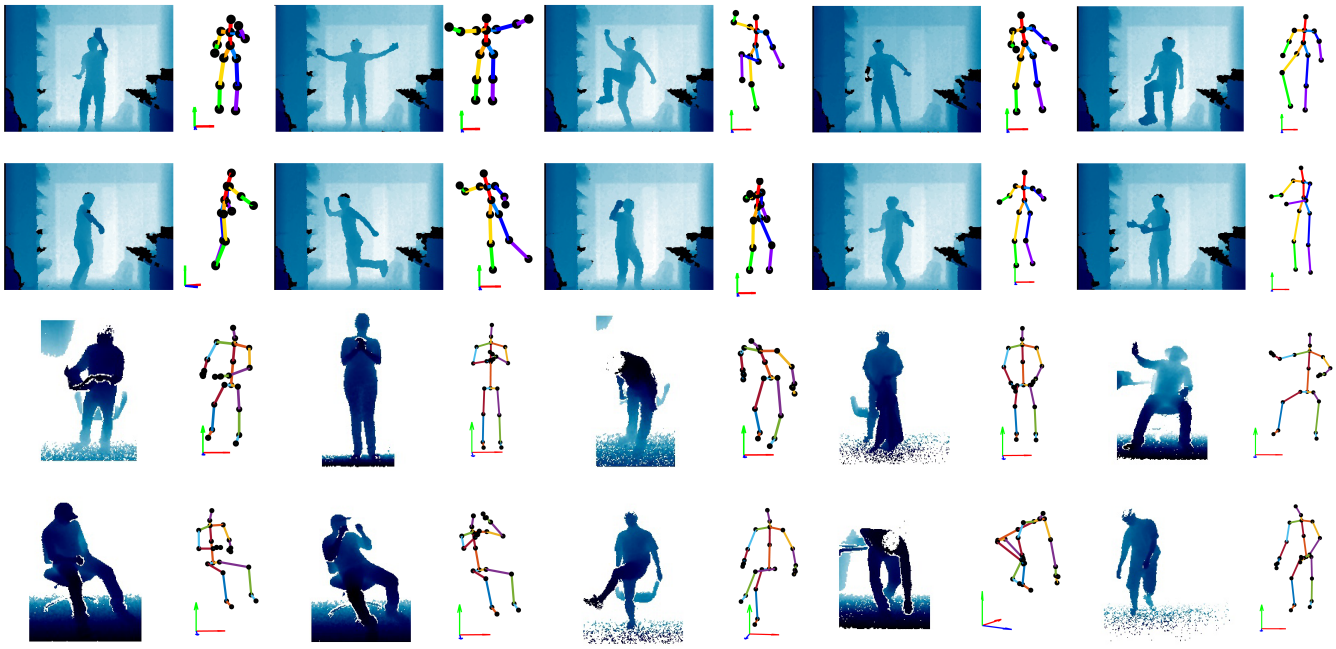


Figure 5: Qualitative results of our methods. The results on the ITOP dataset are shown in the first two rows, and the results on the NTU-RGBD dataset are shown in the last two rows.

isting depth-based methods. We also conduct comparisons on the NTU-RGBD dataset, and we compare our method with the state-of-the-art WSM [Zhang *et al.*, 2020]. As shown on the right side of Table 2, the mean average precision (mAP) of our method is 7.07 percentage points higher than that of WSM [Zhang *et al.*, 2020].

In Fig. 4, we show the qualitative results of a consecutive sequence showing a person turning around. We uniformly select some frames from the sequence and show the estimation results of WSM, our method and the ground truth. From the figure, we can determine that the poses generated by WSM have the error of swapping the left and right arms starting from the second row, but our method produces reasonable results due to the sequential information. Our methods outperform other state-of-the-art methods on the ITOP dataset.

In order to investigate whether depth is a key clue to achieve better human pose estimation results than the method of solely lifting 2D poses to 3D, we also compare our method with the RGB-based pose estimation method. In the comparison, we choose V3D for comparison because V3D is trained with sequential data in a weakly supervised manner, which is consistent with our approach. Since V3D is also a two-stage method, we can easily feed the ground truth 2D human pose to V3D and recover the 3D human pose. As shown in Table 2, our method outperforms V3D by a large margin (16.21 percentage points higher than V3D). We can observe that depth information is important in 3D human pose estimation.

Moreover, we conduct comparison experiments on the running time between our method and the other depth image based methods (V2V and WSM) on the ITOP dataset. The running time of our method, V2V and WSM is 50.0, 3.5 and 24.4 FPS on a single NVIDIA 2080Ti GPU. Our density-based point cloud sampling strategy is effective to reduce the input point clouds and improve efficiency.

We show several examples of our estimation results on the ITOP dataset and the NTU-RGBD dataset in Fig. 5.

5 Conclusion

In this work, we propose an effective approach that adopts sequential information and a novel point sampling method to achieve high-fidelity 3D human pose estimation. Our method also adopts the weakly supervised method on sequential data so that we can use more easy-to-access training data. Furthermore, our model is robust over different levels of training data annotations. Experiments demonstrate that our method can achieve state-of-the-art performance on two main benchmark datasets. Our method could inspire related studies such as those on differentiable point cloud sampling.

Although inspiring results were obtained from this research, this work can be further improved. First, we believe that the combination of depth information and color images can help us generate more reasonable results. Moreover, we only tested our pose estimation methods on dense point clouds, *i.e.,* the point clouds captured by a depth camera. However, human pose estimation methods from sparse point clouds captured by LiDAR are expected in the future.

6 Acknowledgments

This work was supported by National Key R&D Program "Science and Technology Winter Olympics" Key Special Project No.2020YFF0304701, the "Innovative Project" of the Institue of Computing Technology, CAS, and Distinguished Young Researcher Program, Institute of Software, CAS.

References

[Arnab *et al.*, 2019] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2019.

[Bertasius *et al.*, 2019] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely-labeled videos. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2019.

[Dabral *et al.*, 2017] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, and Arjun Jain. Structure-aware and temporally coherent 3d human pose estimation. *arXiv preprint arXiv:1711.09250*, 2017.

[Ding and Fan, 2014] Meng Ding and Guoliang Fan. Fast human pose tracking with a single depth sensor using sum of gaussians models. In *International Symposium on Visual Computing*, pages 599–608, 2014.

[Gao *et al.*, 2019] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[Haque *et al.*, 2016] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *European Conference on Computer Vision*, 2016.

[Kanazawa *et al.*, 2019] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Lang *et al.*, 2020] Itai Lang, Asaf Manor, and Shai Avidan. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020.

[Lin *et al.*, 2017] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3d pose sequence machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 810–819, 2017.

[Liu *et al.*, 2019a] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[Liu *et al.*, 2019b] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, pages 965–975, 2019.

[Liu *et al.*, 2021] Zhenguang Liu, Haoming Chen, Runyang Feng, Shuang Wu, Shouling Ji, Bailin Yang, and Xun Wang. Deep dual consecutive network for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[Moon *et al.*, 2018] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, 2018.

[Pavllo *et al.*, 2019] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Qi *et al.*, 2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[Qi *et al.*, 2017b] 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.

[Qi *et al.*, 2018] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[Qi *et al.*, 2019] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019.

[Rayat Imtiaz Hossain and Little, 2018] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 68–84, 2018.

[Shahroudy *et al.*, 2016] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[Zhang *et al.*, 2020] Zihao Zhang, Lei Hu, Xiaoming Deng, and Shihong Xia. Weakly supervised adversarial learning for 3d human pose estimation from point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[Zhou and Tuzel, 2018] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[Zimmermann *et al.*, 2018] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 3d human pose estimation in rgbd images for robotic task learning. In *IEEE International Conference on Robotics and Automation*, 2018.