

# LiCamPose: Combining Multi-View LiDAR and RGB Cameras for Robust Single-timestamp 3D Human Pose Estimation

## Supplementary Material

### 6. Different Scanning Patterns of Point Cloud

There are various methods to obtain or scan the point cloud: 1) randomly sampling the depth map; 2) sampling the depth map using multiple equidistant horizontal lines to mimic Velodyne LiDARs; and 3) sampling the depth map with the "Rose curve" sampling equation as discussed in our paper to replicate Livox LiDARs. Figure 9 illustrates that the "Rose curve" sampling equation yields minimal information due to its localized concentrated scan. However, Livox LiDARs are more cost-effective than Velodyne LiDARs and have been employed in numerous applications, including surveillance. Additionally, our Baseket-Ball dataset is captured using Livox LiDARs. Therefore, we adopt the Livox scanning pattern to simulate the point cloud scanning in our experiments.

### 7. BaseketBall

BasketBall is an outdoor dataset capturing a basketball match using four sensor nodes, each comprising one Livox LiDAR and one RGB camera, in a convergent acquisition setup. The dataset presents challenges due to its extensive coverage, occlusions, and the dynamic motions of the players (Figure 3). We have developed an annotation tool to label the players' 3D bounding boxes and IDs. In the future, we plan to integrate 3D human keypoint annotation into the tool with the assistance of LiCamPose.

### 8. The Detailed Structure of V2V-Net and Fusion-Net

Figure 8 shows the detailed structure of V2V-Net and Fusion-Net.  $i = 1$  for the one of point cloud information and  $i = K$  for the one of RGB information,  $K$  is the number of joints.  $X, Y, Z$  represents the setting of volumetric space, and  $F$  represents  $F_1$  and  $F_2$ . As indicated in the legend, the yellow arrow represents a standard 3D convolutional layer, while the blue arrow denotes a Residual Block consisting of two 3D convolutional layers. As indicated in the legend, the yellow arrow represents a standard 3D convolutional layer, while the blue arrow denotes a Residual Block consisting of two 3D convolutional layers.

### 9. Human Prior Loss

We designed the human prior loss to encourage the network to generate human-like 3D keypoints. The human prior loss comprises three components: 1) the predicted

bone lengths should be within a reasonable range; 2) the predicted lengths of symmetric bones should be similar; and 3) the predicted bone angles should be reasonable according to human kinematics.

We set a limited length range for all bones. In our case, we set  $l_{\min} = 0.05\text{m}$  and  $l_{\max} = 0.7\text{m}$ . So the  $\mathcal{L}_{\text{length}}$  can be defined as:

$$\mathcal{L}_{\text{length}} = \sum_{b=1}^N \mathcal{C}(B_i - l_{\max}, 0) + \mathcal{C}(l_{\min} - B_i, 0), \quad (7)$$

where  $\mathcal{C}(\cdot)$  is the clipping function that clip the value greater than 0,  $N$  is the number of bones. As to the symmetric bones, we set the symmetric bones as a pair, and set  $L_2$  loss among them. So the  $\mathcal{L}_{\text{symm}}$  can be defined as:

$$\mathcal{L}_{\text{symm}} = \sum_{b=1}^N \|B_i - B_{\text{symm}(i)}\|_2, \quad (8)$$

where  $B_{\text{symm}}$  is the symmetric bone of  $B_i$ . As to angle loss, we limit the nose-neck-midhip angle and hip-knee-ankle angle specifically to let nose be in front of the body and legs be bent forward. Figure 10 shows the definition of each joint and vectors. Specially, we do not directly calculate the angle of the bones, but calculate the dot product of corresponding vectors. First, we calculate the forward direction vector  $\vec{d}_{\text{forward}}$  of the body, which is the cross product of the unit vector from neck to midhip  $\vec{J}_0\vec{J}_2$  and unit vector from neck to left shoulder  $\vec{J}_0\vec{J}_3$ :

$$\vec{d}_{\text{forward}} = \vec{J}_0\vec{J}_2 \times \vec{J}_0\vec{J}_3, \quad (9)$$

Then, as to the nose-neck-midhip angle, we calculate the unit vector from neck to nose  $\vec{J}_1\vec{J}_0$  denoted by  $\vec{d}_{\text{nose}}$ , and we calculate the dot product of  $\vec{d}_{\text{nose}}$  and  $\vec{d}_{\text{forward}}$  and get the head angle loss:

$$\mathcal{L}_{\text{head.ang}} = \mathcal{C}(\vec{d}_{\text{forward}} \cdot \vec{d}_{\text{nose}}, 0, 1), \quad (10)$$

where  $\mathcal{C}(\cdot)$  is the clipping function that clip the value into 0 to 1. As to the hip-knee-ankle angle, we need to get the midpoint of the hip and ankle denoted by  $c_l$  and  $c_r$  for left leg and right leg respectively. Then, we calculate the unit vectors from knee point to the leg's midpoint as  $\vec{d}_{l,\text{leg}}$  and  $\vec{d}_{r,\text{leg}}$ . Therefore, we get the leg angle loss:

$$\mathcal{L}_{\text{leg.ang}} = \mathcal{C}(\vec{d}_{\text{forward}} \cdot \vec{d}_{l,\text{leg}}, 0, 1) + \mathcal{C}(\vec{d}_{\text{forward}} \cdot \vec{d}_{r,\text{leg}}, 0, 1), \quad (11)$$

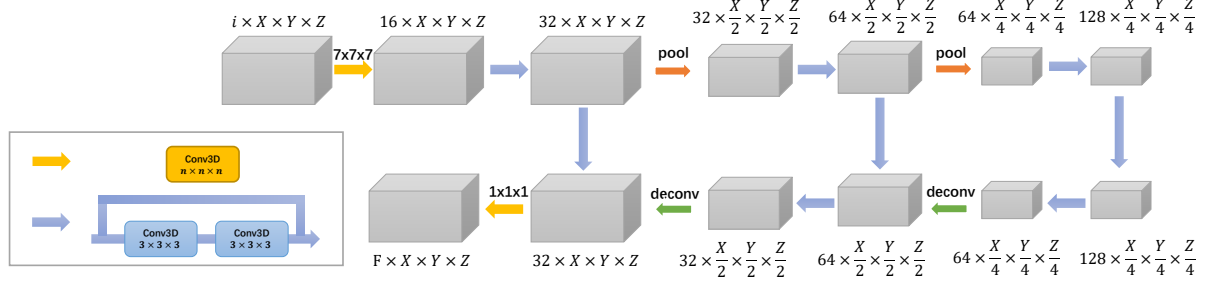


Figure 8. The structure and detailed setting of V2V-Net and Fusion-Net.

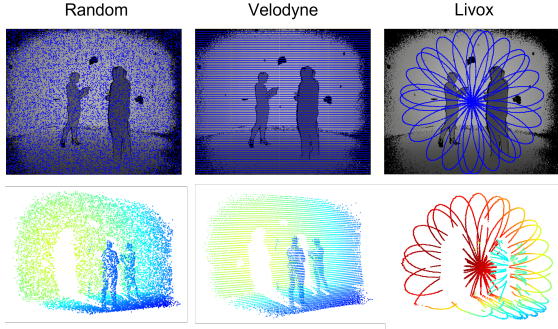


Figure 9. Different scanning patterns of point clouds. All samples shown in this figure are from the same scene, captured at the same time, and contain the same number of points.

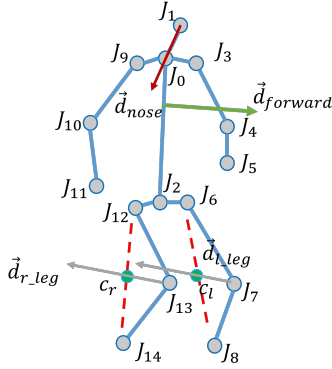


Figure 10. Definition of  $\mathcal{L}_{\text{angle}}$ .

where  $\mathcal{C}(\cdot)$  is the clipping function that clip the value into 0 to 1. Therefore, we can calculate the angle loss:

$$\mathcal{L}_{\text{angle}} = \mathcal{L}_{\text{head\_ang}} + \mathcal{L}_{\text{leg\_ang}}, \quad (12)$$

Finally, we combine the three losses together as the human prior loss:

$$\mathcal{L}_{\text{prior}} = \gamma_1 \mathcal{L}_{\text{length}} + \gamma_2 \mathcal{L}_{\text{symm}} + \gamma_3 \mathcal{L}_{\text{angle}}, \quad (13)$$

where  $\gamma_1$ ,  $\gamma_2$  and  $\gamma_3$  are the weights of each loss. In our case, we set all the weights as 1.

Table 6. Human detection results on different datasets. “#” means using synthetic datasets (BasketBallSync and PanopticSync) to pretrain and directly evaluate on corresponding real-world datasets.

Datasets	Methods	Metric	
		AP <sub>50</sub>	AP <sub>70</sub>
BasketBall	MVDet	69.41	37.66
	PointPillars#	88.17	44.26
	PointPillars	<b>89.77</b>	<b>69.96</b>
Panoptic	VoxelPose	21.17	0.19
	PointPillars#	40.17	6.25
	PointPillars	<b>73.83</b>	<b>13.97</b>

## 10. Extended Experiments

In this section, we conduct experiments to verify the advantages of using point cloud input for pedestrian detection. Additionally, we present more examples to explain the relationship between entropy value and pose rationality.

### 10.1. Human Detection

For evaluating human detection, we assess performance using the established average precision (AP) metric as described in KITTI [16]. We consider detections as true positives if they overlap by more than 70% (AP<sub>70</sub>) or 50% (AP<sub>50</sub>).

In our current experiment, we adopt PointPillars [32] to detect human bounding boxes. For comparison with multi-view RGB-based methods, we utilize VoxelPose’s CPN [52] and MVDet [23], which is more suitable for large scene applications. In the CMU Panoptic Studio setup, VoxelPose [52] achieves relatively accurate center localization. However, it sets the bounding box size to a constant value (we use  $0.8\text{m} \times 0.8\text{m} \times 1.9\text{m}$  for tighter results, compared to  $2\text{m} \times 2\text{m} \times 2\text{m}$  in [52]), which affects detection performance. In the Basketball dataset, we adopt MVDet to detect humans. Table 6 shows that the point cloud-based method outperforms the multi-view RGB-based method in

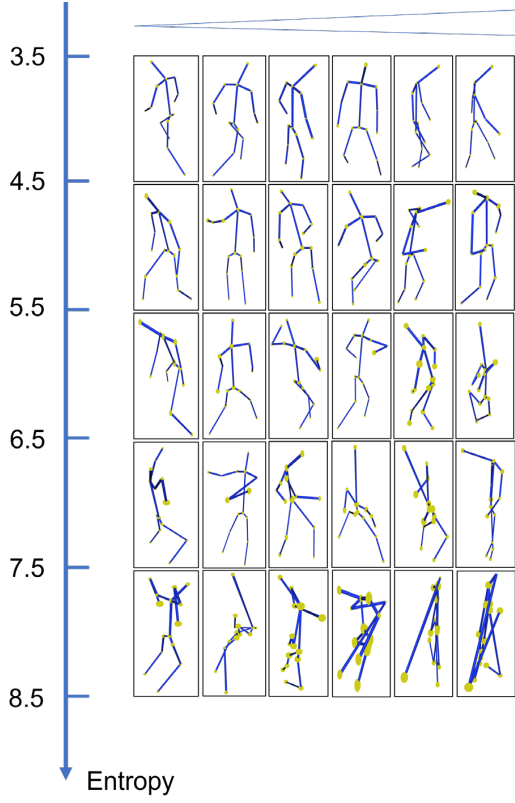


Figure 11. The entropy value and the specific poses.  $\leftarrow$  represents an increasing entropy value from left to right among row's samples. The size of joints' ball represents the magnitude of the joint's entropy value.

terms of  $AP_{50}$  and  $AP_{70}$ , benefiting from the 3D information of the original point cloud. Additionally, we verify the generalization ability of the point cloud-based method by pretraining it on our synthetic dataset, and it still produces acceptable results.

## 10.2. The analysis of unsupervised training losses.

Figure 12 qualitatively shows that these designed unsupervised training losses significantly enhance robustness to 2D pose estimation errors.

## 10.3. Entropy Analysis

Figure 11 shows the entropy value and the specific poses, and we can find that the 3D poses become more and more irrational while the entropy goes up.

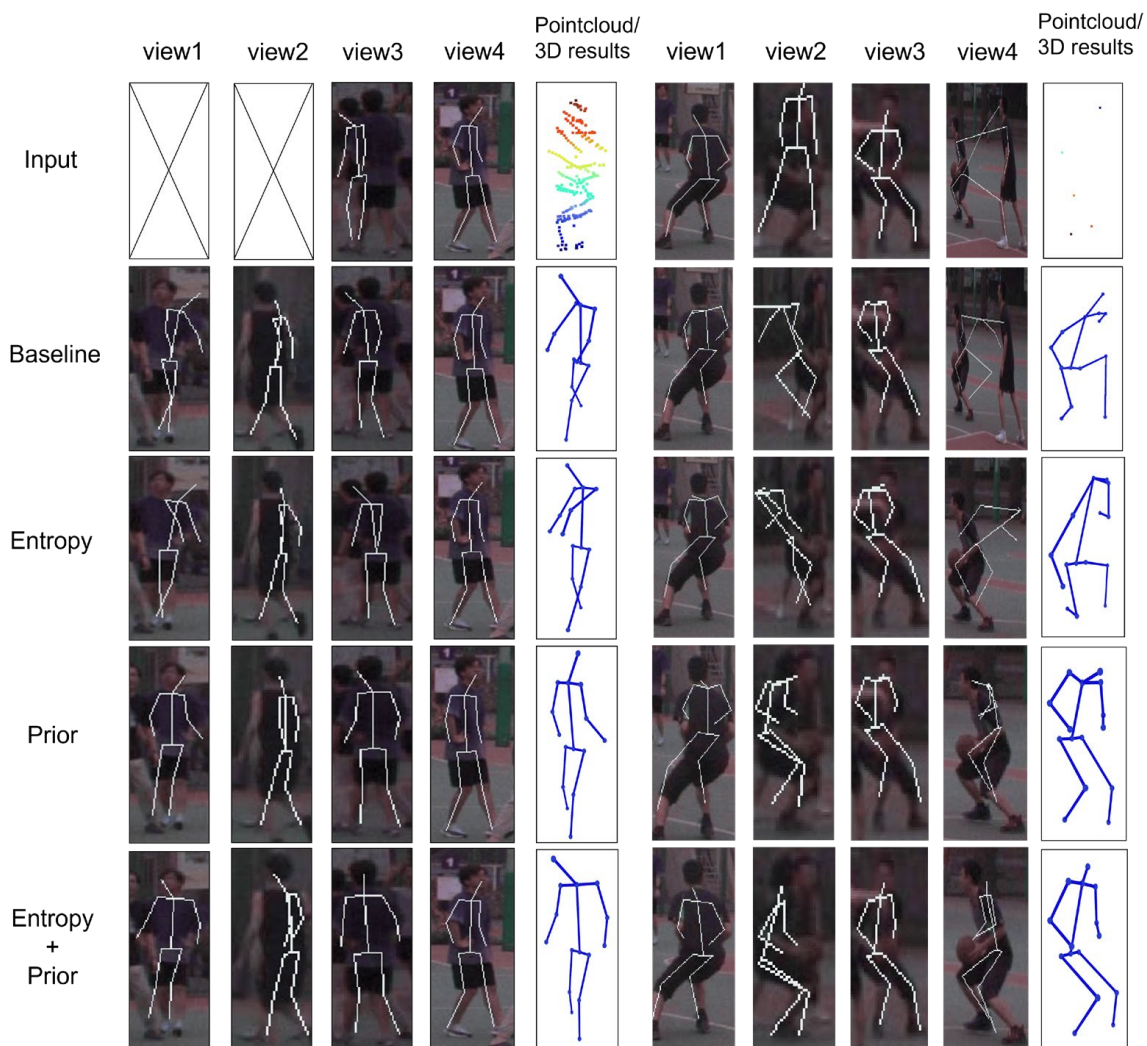


Figure 12. Qualitative visualization on Basketball about different unsupervised training losses. “Baseline” uses only pseudo 2D pose supervision. “Entropy” indicates the addition of entropy-selected pseudo 3D pose supervision. “Prior” denotes the incorporation of human prior loss.