# Supplementary Materials of "milliFlow: Scene Flow Estimation on mmWave Radar Point Cloud for Human Motion Sensing"

Fangqiang Ding[1], Zhen Luo[1], Peijun Zhao[2], and Chris Xiaoxuan Lu[3]

[1]University of Edinburgh [2]MIT [3]UCL

This supplementary document is organized as follows:

- Sec. A illustrates more details about the mmWave radar we used for experiments and two deep learning layers for point cloud processing.
- Sec. B presents more implementation details on sensor specifications, data preprocessing, evaluation metrics and network and application strategies.
- Sec. C provides sensitivity analysis of our scene flow network against aggregation mechanisms, test environment and the number of training subjects.
- Sec. D explains several limitations to be considered for future work.

## A Preliminary

### A.1 SFCW mmWave Radar

In this work, we employ the Vayyar radar [2], which is a stepped-frequency continuous-wave (SFCW) radar [9] bespoken designed for fine-grained human sensing. This subsection aims to provide a concise overview of the point cloud generation principles of SFCW radar to aid in comprehending the hardware components of the proposed system. In what follows we introduce the working principle of SFCW radars.

**Range Estimation.** With SFCW radar, we can detect range information with a single Tx-Rx pair. For a CW radar, the transmitting signal can be written as $s(t) = Asin(2\pi f_0 t)$, and the corresponding receiving signal from the reflection on an object at range $R$ would be $s_r(t) = A_r sin(2\pi f_0 t - \phi)$, where $\phi$ is the phase difference of the transmitting and receiving signals, and can be written as $\phi = 2\pi f_0 \frac{2R}{c}$. As a result, we are able to estimate the range as $R = \frac{c}{4\pi f_0}\phi$. Note that $\phi$ can only be in range $[0, 2\pi]$. As a result, for CW radar that works on a single frequency, the maximum unambiguous range is very limited. For example, if $f_0 = 2GHz$, $R_{max} = 7.5cm$. With SFCW radar, we are able to get the phase difference at different frequencies, which provides much richer information for range estimation. For two consecutive frequencies $f_1$ and $f_2$, the frequency difference $\Delta f = f_2 - f_1$, and the phase differences at each frequency step are $\phi_1 = \frac{4\pi R}{c} f_1$ and $\phi_2 = \frac{4\pi R}{c} f_2$. We have the following equation:

$$\Delta\phi = \phi_2 - \phi_1 = \frac{4\pi R}{c}(f_2 - f_1) = \frac{4\pi R}{c}\Delta f \tag{1}$$

and we get

$$R = \frac{c}{4\pi \Delta f} \Delta\phi \tag{2}$$

Also, we have $\Delta\phi \in [0, 2\pi]$, so the maximum unambiguous range is inversely proportional to $\Delta f$, which is the frequency step in SFCW radar. During each frame, the transmitter sequentially emits a series of waves of increasing frequency at equal intervals. The transmitted signal is reflected back at different surfaces in the scene and received by the receiving antenna. The ADC samples the change in amplitude and phase between the transmitted signal and the received signal and stores the values in IQ format. We can apply the Fourier Transform to get the phase variation, with which we can further derive the range of the objects. This is also known as 'Range-FFT'. The range resolution is inversely proportional to the bandwidth of the radar, written as $R_{res} = \frac{c}{2B}$, where $B = f_{max} - f_{min}$.

**Angle-of-arrival Estimation.** Angle-of-Arrival (AoA) estimation can be performed using a linear receiver array with equally-spaced elements, typically separated by a distance of $1/2\lambda$, where the number of elements in the array, denoted as $N$, is greater than or equal to two. When a reflected signal is received at each element of the array, a phase difference arises due to the slight differences in signal travel distance, as determined by the angle of arrival, $\theta$. This phase difference can be quantified using the formula $\Delta\phi = kdsin(\theta)$, where $k$ represents the wave number and $d$ is the distance between consecutive elements. The angle of arrival can then be estimated by analyzing the phase difference using a Fourier Transform technique, commonly referred to as 'Angle-FFT'. Vayyar uses Multiple-Input Multiple-Output (MIMO) array, and for each frame, we are able to get a 2D virtual antenna matrix, from which we can estimate azimuth and elevation AoA simultaneously.

**Point Cloud Generation.** For each frame, the raw data is stored as a complex matrix of size $N_a * N_e * M$, where $N_a$ and $N_e$ are the numbers of Tx-Rx antenna pairs (i.e., virtual antennas) in azimuth and elevation directions, respectively, and M is the number of frequency steps. First, the clutter removal is applied to focus on dynamic objects. We then apply Range-FFT for Range estimation for each virtual antenna and further perform 2D Angle-FFT to get the AoA information in both directions. Following the above processing, we are able to get a 3D heatmap for each frame. Strong peaks are then detected with methods like CFAR [8], and converted to 3D points, which is the input to our scene flow estimation.

To summarize, given the 3D data cube, mmWave radar conducts the range-FFT, Doppler-FFT and two 'Angle-FFT' to obtain a 4D radar heatmap. Strong peaks are then detected to generate a radar point cloud, which is the input to our scene flow estimation model.

### A.2 Deep Learning Layers for Point Cloud

**Set Abstraction Layer.** In our radar scene flow network, we adopt the set abstraction layer proposed in PointNet++ [7] as the basic learning layer to extract local point features. As an extension of PointNet [6], PointNet++ [7]

proposes a hierarchical structure composed of multiple set abstraction levels. At each level, the local features of a set of points are abstracted and taken as the input to the next level. The working process of each set abstraction layer can be summarized in three steps. First, the iterative farthest point sampling (FPS) is applied to select $N'$ points the point sets as the centroids for local regions. Second, the ball query identifies neighbour points within a radius $R$ of each centroid, resulting in $N'$ local point sets. Lastly, the local region features are encoded for each centroid using the PointNet layer [6], which is composed of an MLP to extract high-level representations and a max-pooling layer to aggregate the representations of all local points to the centroid. The output is the per-point local features for $N'$ selected points at a specific scale determined by the radius $R$. We believe that this learning layer can effectively gather information from neighbour points to each radar target.

**Cost Volume Layer.** To encode points motion between two radar frames, in our network design, we leverage the point cloud cost volume layer in [10] to correlate features. In the cost volume layer, the matching costs $Cost(p_i, q_j)$ of all point pairs $(p_i, q_j)$ between two point clouds $\mathcal{P}$ and $\mathcal{Q}$ are first calculated through an MLP. Then, the costs are aggregated in a patch-to-patch manner to produce robust and stable cost volumes. During aggregation, the neighbourhood is first found for each point $p_c$ in $\mathcal{P}$. Then, for each neighbour point $p_i$ of $p_c$, a neighbourhood is found around it in $\mathcal{Q}$. With these patches in $\mathcal{Q}$ and $\mathcal{P}$, the costs can be aggregated respectively through a weight-sum operation whose weights are leanrable parameters from MLPs. The output is two-dimension cost volumes with shape $N \times D$, where $N$ is the number of points of the point cloud $\mathcal{P}$ and $D$ is the dimension of the cost volume.

## B Implementation Details.

### B.1 Sensor Specifications

As said in the main paper, we use a commercial Vayyar vTrigB imaging mmWave radar [2] and a RealSense D455 depth camera [1] to capture mmWave radar point clouds and RGB-D images respectively. The mmWave radar device is portable with a size of 10.4 cm $\times$ 8.5cm and a weight of 110g in total, which is designed following the stepped frequency continuous wave (SFCW) principles (*cf*. Sec. A). With the default sensor setting, the range resolution of our mmWave radar is 9.35 cm, and the maximum range is 14 m, while the angular resolution is approximately 6.7 degrees. The frequency bandwidth is set as 1.6GHz (from 62 to 63.6 GHz) and the frequency step is 10.66 MHz, which results in 151 frequency samples for each frame. 16 RX and 20 Tx antennas on its PCB are employed for producing radar data, which equals 320 Rx/Tx virtual pairs. The internal data processing pipeline of this device follows the steps we illustrate in Sec. A.1. The final output is a set of 3D points with per-point intensity values. For the

depth camera, the RGB-D image size is set as 640×480 and its depth measurement range is from 0.6m to 6m. The average frame rate over all collected data is ∼13.2Hz.

### B.2 Data Preprocessing

Given sequences of mmWave radar point clouds, we first filter them by range to discard points outside our region of interest. The side, forward and height range in the radar coordinate frame is set as [(-3, 3), (0.5, 5), (-1.5, 1.5)] meters respectively. We then filter points by intensity to omit the background points. The intensity threshold is empirically set as 0.5. For all training and validation frames, we randomly sample 128 points from each radar point cloud to facilitate fast mini-batch-based training. We keep the number of points unchanged in all testing frames to make sure that the scene flow estimation for each point can be examined. In the end, we generate scene flow samples by combing pair-wise frames. Each 200-frame sequence can produce 199 samples, each of which consists of two consecutive mmWave radar point clouds and their corresponding RGB-D images.

### B.3 Evaluation Metrics

We use the following evaluation metrics to quantify the performance of scene flow estimation and downstream human sensing tasks.

- **EPE3D (m)**. It computes the average 3D endpoint error $||f_i - f_{gt}||_2$ over all points in a frame. Following [3, 4], we also report the errors separately for the *moving* and *static* points. Points with a ground truth flow vector larger than 0.01m are labelled as *moving*.
- **Acc3D**. It measures the ratio of points in one frame that meets the strict/relax accuracy requirements, i.e., either EPE3D $< 0.05/0.1$m or relative error $<5\%/10\%$. These two metrics (strict and relax) can represent the estimation accuracy well in autonomous driving scenarios [3, 10]. However, in our human sensing scenarios, the movement scale of points is often smaller than 0.05m per frame. As a result, almost all points can easily meet the 0.05/0.1m requirement. To better examine and compare different methods, we redefine these two metrics by reducing 0.05/0.1m to 0.025/0.05m while keeping the relative error 5%/10% requirement unchanged.
- **Overall accuracy (oA) (%)**. It measures the proportion of examples for which the predicted label matches the single target label, i.e., $\frac{TP+TN}{TP+FP+TN+FN}$. In our experiments, we use it to formulate HAR and HP problems as per-sequence and per-point classification problems.
- **Mean Intersection over Union (mIoU) (%)**. For a single body part class, its IoU is defined as $\frac{TP}{TP+FP+FN}$, measuring the overlap between the predicted subset of points and the ground truth one divided by their union. The mean IoU (mIoU) score is calculated by averaging scores on all body part classes. In our evaluation, we use this metric for our HP tasks to show the paring results.

– **Mean Joint Localization Error (mJE) (m)**. This metric is used for our HBPT task. We define it as the mean 3D Euclidean distance between the positions of endpoints of the tracked body parts and their ground truth positions.

### B.4 Network and Training Details

For the scene flow network, the grouping radii for four SA layers used for local or context feature extraction are [0.05, 0.1, 0.2, 0.4] meters. The numbers of local samples for them are set as [4, 8, 16, 32] and the dimension of the MLP used in and after each SA layer is [32, 32, 64], [64, 64, 64] respectively. In all global feature aggregation, the MLP used for attention mapping is two-layer with a hidden dimension of 128. In the CV layer, the number of neighbours for patch-to-patch aggregation is set as 8 and each MLP used to learn aggregation weights has three layers with hidden dimensions of [8, 8]. The local encoder used to generate flow embedding has the same hyperparameters as the former one except for its MLP which has the dimension of [512, 256, 64] in each SA layer. The dimension of the flow regressor MLP is [256, 128, 64, 3] and the threshold $\epsilon$ to constrain the output is set as $0.1m$. To implement our temporal propagation, we divide the long sequence of scene flow samples into many mini-clips with a length of 5 frames and train using mini-batches of them after shuffling. During inference, we re-initialize the hidden state of the GRU to zero vectors after 5 frames to fit the training pattern.

The threshold $\zeta$ used for our loss function is fixed as 0.1m for all experiments while the weight hyperparameters $\alpha_l$ and $\alpha_s$ are set as 2 and 1. For all training in our experiments, we consistently use the Adam optimizer with an initial learning rate of 1e-3, which decays by 0.9 after each epoch. The performance on the validation set is used to determine when to stop the training process and to select the best model for each training experiment.

### B.5 Downstream Network

**HAR Base Network.** Given a sequence of $T$ radar point clouds together with their point-level features as input, we follow the process in local feature abstraction and global feature aggregation to extract the local-global representations for each point cloud. Then we use another local encoder to extract higher-level features and aggregate the global feature vector again for each frame. We utilize the LSTM network [5] to track the temporal relationship across $T$ global vectors and then send the updated hidden state into a final MLP to regress the classification scores $S^a = \{s_i^a\}_{i=1}^K$ for $K$ classes. The HAR network can be supervised with a simple cross-entropy loss that measures the difference between $S^a$ and a one-hot ground truth activity label.

**HP Base Network.** Similar to HAR, HP network also takes sequential radar point clouds as input and utilizes local encoders as well as global aggregation to obtain the latent representation and a global feature vector for each frame. Differently, the human parsing task needs to estimate per-point classification scores for every point rather than the whole sequence. Therefore, after applying

**Table A:** Comparison of global aggregation mechanisms.

| | EPE3D (m) ↓ | | | Acc3D ↑ | |
|---|---|---|---|---|---|
| Mechanism | All | Moving | Static | Strict | Relax |
| Attention-based | **0.046** | **0.051** | **0.009** | **0.406** | **0.703** |
| Max-pooling | 0.050 | 0.059 | 0.011 | 0.394 | 0.679 |
| Average-pooling | 0.047 | 0.055 | 0.009 | 0.399 | 0.698 |

**Table B:** Results for different test environments.

| | EPE3D (m) ↓ | | | Acc3D ↑ | |
|---|---|---|---|---|---|
| Scene | All | Moving | Static | Strict | Relax |
| Hallway | 0.040 | 0.046 | 0.007 | 0.464 | 0.746 |
| Square | 0.042 | 0.049 | 0.009 | 0.435 | 0.722 |
| Parking lot | 0.051 | 0.058 | 0.011 | 0.374 | 0.671 |

the temporal information propagation, we use the point-level final features to regress the parsing scores $S^p = \{s_i^p\}_{i=1}^V$ for each point. Here $V$ denotes the number of human body segments. Our HP network is also supervised using a cross-entropy loss with per-point parsing labels. Specifically, we average the loss values over all classes to balance their impact on training.

### B.6 Scene Flow Application Strategies

As told in the main text, we propose two strategies to harness the scene flow network as a plug-and-play module to the downstream HAR and HP network. We believe both can effectively improve downstream performance.

**S1 - Point cloud decoration.** The first way is to directly take the estimated scene flow as point-level raw features and decorate each radar point with them. Such additional scene flow features can explicitly provide the full motion information of radar point clouds. We adopt a two-stage training manner to implement this strategy, which first end-to-end trains the scene flow network and then freezes it to provide raw scene flow features to the downstream network training.

**S2 - Latent feature recycling.** Another way is to leverage the latent representations encoded by scene flow networks to enhance the low-quality radar point clouds. By guiding the network to estimate accurate scene flow, the learned representation can provide high-level spatial-temporal information to downstream networks in a specific aspect. We follow a joint learning fashion to implement this strategy that takes the final features from the scene flow network as the input features to the downstream network and trains these two networks *jointly* with a combination of the scene flow and downstream loss functions.

## C Sensitivity Analysis

Here we investigate the sensitivity of our scene flow network against a) global aggregation mechanisms, b) test environments and c) the number of training subjects, and conduct separate analyses for them in the following.

### C.1 Impact of Global Aggregation Mechanism

Besides the attention-based mechanism, another two typical operations, i.e., max-pooling and average-pooling, are tested into our network for global feature aggregation. The comparison results are shown in Table A. From the table,

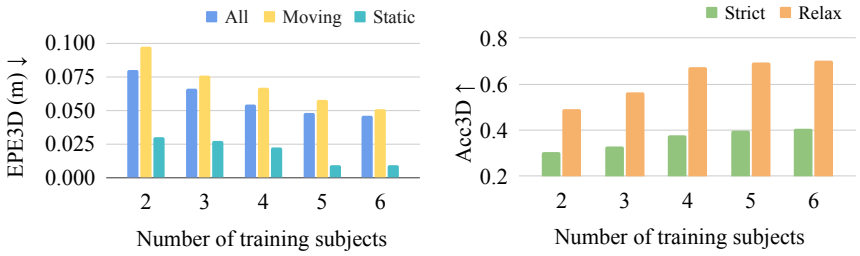


**Fig. A:** Impact of the number of training subjects.

the attention-based mechanism yields the best performance as it can learn to dynamically adjust the weights according to per-point local features. The average-pooling performs better than the max-pooling. We credit this to the ability of the average-pooling to retain the mean feature across all points, which is more robust to outliers.

### C.2 Impact of Test Environment

In Table B, we report the results of our trained model on each scene individually. As we can see, our model shows the best performance for the hallway scene while performing worst in the parking lot scene. We credit the difference in performance between test environments to two factors. First, the background clutter in the parking lot, is more complicated, which results in severe multi-path reflection during measurement. This decreases the SNR in radar data and further degrades the fidelity of generated radar point clouds. Second, our pseudo label generation is affected by the changeable outdoor illumination conditions. Not only does the 2D keypoint estimation on RGB images become more challenging, but also the strong sunlight can interfere with the depth measurements. Consequently, the quality of skeleton estimation in outdoor scenes is lower and the training is not as effective as in the indoor scene.

### C.3 Impact of Training Subjects Number

In our experiments, the data from 6 subjects are used for training. Here we utilize 2/3/4/5 people out of the training set respectively for training to analyze the impact of the number of training subjects. As seen in Figure A, the performance of our network increases in all metrics as more subjects are added for training. This proves that there is still room to improve our performance when more training data is available. We can also observe that, even with only 3 training subjects (9k data frames), our network can still generalize well to the testing set with an overall EPE3D of 0.066m and a relax Acc3D of 0.56. This result further demonstrates the feasibility of our automatic scene flow labelling scheme to generate reliable pseudo labels.

## D Limitation and Future Work

Despite the progress achieved, there are limitations to be considered in future work. First, we only experiment on a stationary radar platform where only the human subject is movable. This is akin to the scenarios of human motion sensing in smart spaces and sensors are receded into the background environment. In future work, we plan to evaluate our system on mobile platforms, such as drones and wheeled robots, where the scene flow can be induced by both ego-sensor and subject motion. Second, as a proof-of-concept work, the functioning of our method is restricted to the single subject with face-forward body motion at this stage. We plan to extend our approach to multi-person scenarios and introduce variations in the subjects' position and orientation relative to the sensor in the next study. Lastly, the number of activities included in our dataset is limited as we only design activities that can be separated into the rigid motion of multiple body skeletons. This fits our need to annotate scene flow labels based on body skeleton displacement. In our upcoming work, we will design more complex and smaller activities and exploit a novel approach to derive the scene flow.

## References

1. Intel® realsense™ depth camera d455 (2023), `https://www.intelrealsense.com/depth-camera-d455/`
2. Vayyar imaging - home - vayyar (2023), `https://vayyar.com/`
3. Baur, S.A., Emmerichs, D.J., Moosmann, F., Pinggera, P., Ommer, B., Geiger, A.: SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation. In: Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference. pp. 13126–13136 (2021)
4. Ding, F., Pan, Z., Deng, Y., Deng, J., Lu, C.X.: Self-Supervised Scene Flow Estimation With 4-D Automotive Radar. IEEE Robotics and Automation Letters pp. 1–8 (2022)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
6. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
7. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems **30** (2017)
8. Scharf, L.L., Demeure, C.: Statistical signal processing: detection, estimation, and time series analysis. Prentice Hall (1991)
9. Weiss, J.M.: Continuous-wave stepped-frequency radar for target ranging and motion detection. In: Proceedings of MICS symposium (2009)
10. Wu, W., Wang, Z.Y., Li, Z., Liu, W., Fuxin, L.: PointPWC-Net: Cost Volume on Point Clouds for (Self-) Supervised Scene Flow Estimation. In: Proceedings of the European Conference on Computer Vision. pp. 88–107 (2020)