DeepEgo

Deep Instantaneous Ego-Motion Estimation Using Automotive Radar

Zhu, Simin; Yarovoy, Alexander; Fioranelli, Francesco

DOI
10.1109/TRS.2023.3288241

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Radar Systems

DeepEgo: Deep Instantaneous Ego-Motion Estimation Using Automotive Radar

Simin Zhu, *Graduate Student Member, IEEE*, Alexander Yarovoy, *Fellow, IEEE*, and Francesco Fioranelli, *Senior Member, IEEE*

***Abstract*— The problem of instantaneous ego-motion estimation with mm-wave automotive radar is studied. *DeepEgo*, a deep learning-based method, is proposed for achieving robust and accurate ego-motion estimation. A hybrid approach that uses neural networks to extract complex features from input point clouds and applies weighted least squares (WLS) for motion estimation is utilized in *DeepEgo*. Additionally, a novel loss function, *Doppler loss*, is proposed to locate "inlier points" originating from detected stationary objects without human annotation. Finally, a challenging real-world automotive radar dataset is selected for extensive performance evaluation. Compared to other methods selected from the literature, significant improvements in estimation accuracy, long-term stability, and runtime performance of DeepEgo in comparison to other methods are demonstrated.**

***Index Terms*— Ego-motion estimation, radar odometry, automotive radar, radar point cloud, deep learning.**

I. Introduction

OVER the past few decades, advanced driving systems (ADSs) have attracted a lot of attention in industry and academia. Nowadays, self-driving cars use ADS to assist or even replace the driver in handling complex driving situations [1], [2]. It is considered to be a technology that could reduce energy consumption and potentially save many lives by reducing road accidents caused by human errors [3]. At the system level, ADS can be decomposed into a perception frontend and a decision-making backend. The backend performs tasks such as path planning and motion control, and the frontend is responsible for driving scene understanding [4].

Among many signal processing steps in the frontend, ego-motion estimation (i.e., self-localization) is one of the first signal processing components that directly processes the raw sensor data [5]. Therefore, it is extremely important to have a robust and accurate ego-motion estimator. In this way, the performance of downstream processing steps, such as multi-object tracking and environment mapping, will not be seriously affected by errors in such estimation.

Conventionally, the motion of the ego-vehicle can be determined using odometry sensors such as inertial measurement units (IMU), wheel encoders, and global positioning systems (GPS). However, the IMU drifts over time [6], the wheel encoder is subject to slippage of the wheels under acceleration and braking [7], and the GPS is susceptible to non-line-of-sight (NLOS) propagation and multipath reflections [8]. Therefore, these shortcomings make them insufficient on their own for reliable ego-motion estimation.

To address this issue, ego-motion estimation using other sensor technologies such as camera [9], LiDAR (Light Detection and Ranging) [10], SAS (Synthetic Aperture Sonar) [11], scanning radar [12], or automotive radar [13] has been introduced. Compared to cameras, automotive radar works in all weather and illumination conditions [14]. Also, it is less susceptible to line-of-sight (LoS) occlusions than LiDAR [15]. Furthermore, automotive radars are generally lightweight, low-cost, and compact, thanks to the millimeter-wave (mmWave) technology [16]. Due to its compactness, radar can be easily installed behind the bumper of the vehicle [17]. Moreover, unlike other sensor modalities, automotive radar can directly estimate the Doppler/velocity of detected objects [18]. The additional Doppler information has been discovered to be helpful for applications such as moving object tracking [19] and instantaneous ego-motion estimation [20].

Even if the use of automotive radar can potentially provide more robust ego-motion estimation performance than other sensors, it has several challenges. For example, radar data typically contains fewer geometrical characteristics of detected objects, because of its low resolution in range and angle-of-arrival (AoA) estimation [21]. Additionally, radar measurements are subject to various factors such as false positives, missed detections, radar cross section (RCS) fluctuations, multipath reflections, and mutual interference [22]. These challenges make it difficult for some well-established ego-motion estimation methods to be applied directly to radar data. To fill this gap, the objective of this work is to design a robust and accurate ego-motion estimation algorithm tailored for processing data from automotive radars.

In this work, we propose an end-to-end solution for instantaneous ego-motion estimation using neural networks. Compared to conventional scan-matching methods, the proposed method requires only one scan of radar data for estimation, avoiding to solve the challenging task of data association. Also, instead of forcing the neural network to regress ego-motion [23], the proposed method uses a hybrid approach. Specifically, the neural network learns how to weight each point in the radar point cloud, and the weighted least square

Manuscript received 13 March 2023; revised 10 May 2023; accepted 17 June 2023. Date of publication 21 June 2023; date of current version 28 June 2023. *(Corresponding author: Francesco Fioranelli.)*

The authors are with the Microwave Sensing, Signals and Systems (MS3) Group, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: s.zhu-2@tudelft.nl; a.yarovoy@tudelft.nl; f.fioranelli@tudelft.nl)

Digital Object Identifier 10.1109/TRS.2023.3288241

(WLS) approach is used to compute the final estimation result.

In addition to the above advantages, the proposed method uses neural networks to directly process multi-dimensional radar point clouds, requiring only necessary preprocessing steps such as data normalization and resampling. As shown in the results, utilizing other measured features of the detected object, such as range and return power, can further improve the accuracy in translational and rotational velocity estimation by **64.3%** and **20.9%**, respectively.

Compared to our preliminary results in [24], the method proposed in this work differs significantly in terms of network structure and its training, and presents performance improvements that can be summarized in the following aspects. Firstly, in addition to the conventional motion loss, which penalizes the difference between estimated and ground-truth motion, this work proposes a novel loss function, the Doppler loss, for training the network. The Doppler loss can help the network automatically locate 'inliers' in radar point clouds without manual annotation, which is extremely important for the success of WLS. Moreover, using the Doppler loss, the proposed neural network learns how to predict the likelihood (weight) that a detection point originates from a stationary object. This can then be further utilized beyond ego-motion estimation, in many other downstream processes such as map denoising and static object detection, outside the scope of this work. Secondly, inspired by [25], this work proposes a different network architecture that estimates and adds point-wise offsets to the weighted least squares method. The objective of the point-wise offsets is to move distant 'outliers' closer to the regression line, thereby mitigating their negative impact on the regression. It has been found in [25] that the point-wise offset can stabilize network training for surface normal estimation applications. Besides this effect on training stability, the system performance in ego-motion estimation can be further improved by adding the proposed point-wise offsets, as shown in the ablation study presented in this work. Thirdly, the performance of the proposed method is thoroughly evaluated and compared with the performance of six other methods available in the literature using a challenging real-world automotive radar dataset, *RadarScenes* [26]. In detail, the proposed method is tested on 64 radar data sequences with an equivalent drive length of more than **79km**. It is shown to improve the estimation accuracy by more than **50%** compared to the second-best method selected from the literature, as well as a general improvement across all relevant performance metrics. Furthermore, the proposed method is straightforward and requires no iterative optimization or random sampling. Quantitatively, it is about **129.6** times faster than the Normal Distribution Transform (NDT)-based method used for comparison.

The rest of the paper is structured as follows. An overview of published research on this topic is given in Section II. Next, design details of the proposed method are provided in Section III. The performance of the proposed method and other methods in the literature are compared in Section IV, using a challenging real-world automotive radar dataset. After that, an ablation study is presented in Section V, in order to illustrate the impact of the design choices made in the proposed method. Finally, conclusions and future studies are given in Section VI.

II. Related Works

This section reviews the recent advances in (A) Ego-motion estimation with automotive radar and (B) Deep learning on point cloud processing from which the proposed method of this paper is inspired.

A. Ego-Motion Estimation With Automotive Radar

While the idea of using automotive radar for ego-motion estimation began a decade ago, researchers have been working on the problem for some time [27].

The Iterative Closest Point (ICP) [28], [29], [30] is one of the first algorithms to locate robots based on continuous measurement scans. It is an iterative method that solves the problem of unknown scan-to-scan association by iteratively modifying the required transformation and aligning the closest points between scans. Due to its broad applicability, many ICP variants have been proposed, affecting all stages of the algorithm, from the data association strategy to the optimization process [31].

It is known that the success of ICP and its variants depends on reliable and stable data collection [32]. However, this is usually not a valid assumption, especially for mm-wave radar data where the point clouds of the same object can significantly change from frame to frame. To address this issue, NDT has been introduced as a compelling alternative [33]. Instead of requiring explicit data association like ICPs, NDT uses local normal distributions to represent sensor measurements and implicitly models the uncertainty of data associations. Compared to ICP, NDT is more robust, more consistent, and less sensitive to poor initialization [34].

However, conventional NDTs [35], [36], [37] only use spatial measurements (range and Angle of Arrival, AoA) and are not well adapted to automotive radars. To fully exploit radar data, a variant of NDT based on the spatial and Doppler information is proposed in [38] and [39]. Compared to the standard NDT, it uses the Doppler information to combat artefacts and errors caused by dynamic objects and false positives. Furthermore, this approach uses clustering techniques to provide a globally smooth mixture representation and overcome the discontinuities at grid cell crossings [40]. This approach is further detailed and developed in [41], [42], and [43].

Nevertheless, like ICP, NDT is still an iterative method. Also, it requires at least two valid scans to estimate ego-motion. Moreover, NDT suffers from local maxima due to the sum approximation of the score function [44]. These facts limit its application in demanding tasks such as autonomous driving. To solve the problem, Kellner et al. [20], [45] proposed an instantaneous ego-motion estimation method based on the measured radial velocity profile analysis. The method is specially designed for processing radar data and requires only one measurement scan for motion estimation. In contrast

to ICPs and NDTs, Kellner's method requires no data association. Additionally, the ego-motion estimation accuracy is less susceptible to outliers due to the use of Random Sample Consensus (RANSAC) algorithm [46].

However, the runtime performance and estimation accuracy of Kellner's method are significantly affected by multiple hyperparameters, which cannot be predetermined or adaptively tuned. Also, it is unclear how Kellner's method can make use of other information about detected objects such as range and return power, in addition to radial velocity and AoA.

B. Deep Learning on Point Clouds

Driven by breakthroughs in deep learning (DL) techniques, a large body of recent work has focused on utilizing various DL frameworks for radar perception [47]. However, most influential works have only explored representations of data collected by LiDAR [48], [49], [50] or scanning radar [51], [52], [53], with little consideration for automotive radar. As mentioned earlier, this is mainly because automotive radar data is usually sparser and noisier than other sensor modalities, making the extraction of stable object features from the data a difficult task for neural networks (NNs). Despite the difficulties, some pioneering works have managed to address the problem of ego-motion estimation using automotive radar. However, these works either exploit other odometry sensors to improve the overall performance [23], or are limited to idealized sensing environments [54]. Furthermore, even with scanning radar, most works rely heavily on image-based network architectures, such as using convolutional neural networks (CNNs) [55].

It is worth mentioning at this stage that the multi-dimensional radar point cloud has become the standard format for many commercial automotive radars [56]. Thus, prior to presenting the proposed method of this paper, it is necessary to revisit some conventional DL techniques used to process point clouds:

1) **Image-based methods** such as [57] and [58] propose the use of multi-view projections. They project 3D point clouds onto different 2D planes to form images at various azimuth values. After projection, many well-studied architectures, such as traditional 2D-CNNs, can be applied to extract and fuse features from different viewpoints. However, the performance of these image-based methods is affected by the choice of the projection plane. Additionally, the projection mechanism will inevitably destroy the inherent 3D structure of the point cloud and create occlusion.
2) **Voxel-based methods**, for example [59], attempt to transform raw point clouds into new representations of 3D voxels. In this way, conventional 3D-CNNs can be directly applied. Compared with image-based methods, the voxel representation preserves more geometric features of the object. However, it is computationally inefficient and can incur a large memory cost [60]. Furthermore, compared to LiDAR point clouds, radar point clouds are typically sparser and have higher feature dimensions ($\geq$ 3) [56], making it more challenging to apply voxel-based methods to radar data due to the resulting computational and memory costs. Not to mention that voxelization will inevitably reduce point cloud resolution and degrade localization performance.
3) **Point-based methods** focus on designing NNs that can directly process point clouds, rather than using projections or voxelization. PointNet [61] is one of the most influential works proposed to deal with point clouds. It learns global information of the point cloud through various pointwise operations such as multi-layer perceptron and max-pooling. Subsequent works like PointNet++ [62] and Point Transformer [63] showed further improvements and provided tools for learning local and temporal characteristics from point clouds.

In summary, it is clear that researchers have been trying to propose various solutions for point cloud processing. In this paper we take the approach that point-based methods are better suited for processing automotive radar data than image-based and voxel-based methods. This is because point-based methods can preserve accurate geometric information and directly process high-dimensional point clouds. Finally, due to limited space, readers who are interested in a comprehensive overview of point cloud processing techniques are recommended to refer to [64].

III. Methodology

This section provides the design details of the proposed method for ego-motion estimation.

A. Problem Statement

The proposed method considers the general problem of using a linear array in automotive radar to estimate the 2D motion of the ego-vehicle within a single measurement cycle. The radar data considered in this work is the multi-dimensional radar point cloud $P_{t,\mathbf{n}}^{J\times M}$, where t is the timestamp of the radar scan, $\mathbf{n}$ is the radar index,[1] J is the number of detected points, and M is the number of features of each point of the cloud.

Given a testing vehicle $\mathbf{c}$, it is common to assume that the 2D coordinate system has its origin at the center of gravity of the vehicle, the direction of the x-axis coincides with the down-range motion of the vehicle, and the direction of the y-axis coincides with the cross-range motion of the vehicle. Therefore, the vehicle's 2D motion state can be described as $e_{\mathbf{c}} = \{v_x^{\mathbf{c}}, v_y^{\mathbf{c}}, \omega^{\mathbf{c}}\}$, where $v_x^{\mathbf{c}}$ is the down-range velocity, $v_y^{\mathbf{c}}$ is the cross-range velocity, and $\omega^{\mathbf{c}}$ is the rotational velocity.

Assume that an automotive radar $\mathbf{n}$ is mounted at the position $\{x_{\mathbf{n}}^c, y_{\mathbf{n}}^c, \theta_{\mathbf{n}}^c\}$, where $x_{\mathbf{n}}^c$ is the distance to the x-axis, $y_{\mathbf{n}}^c$ is the distance to the y-axis, and $\theta_{\mathbf{n}}^c$ is the mounting angle relative to the vehicle x-axis. Since the radar measures the relative motion between itself and the detected object, it is reasonable to transform the ego-motion from the vehicle's coordinate system to the radar's coordinate system.

For the radar's coordinate system, it is usually assumed that the direction of the x-axis coincides with the boresight direction of the radar. Then, the 2D motion state of the radar

[1] Assuming a vehicle can carry multiple automotive radars, $\mathbf{n}$ indicates the n-th radar on the vehicle.

$\mathbf{n}$ can be expressed as $e_{\mathbf{n}} = \left\{v_x^{\mathbf{n}}, v_y^{\mathbf{n}}, \omega^{\mathbf{n}}\right\}$. It is important to note that $\omega^{\mathbf{n}} = \omega^{\mathbf{c}}$, since all points on a rigid body experience the same angular velocity. Finally, the transformation between the motion states $e_{\mathbf{c}}$ and $e_{\mathbf{n}}$ can be expressed as:

$$\begin{bmatrix} v_x^{\mathbf{n}} \\ v_y^{\mathbf{n}} \\ \omega^{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} cos(\theta_{\mathbf{n}}^c) & sin(\theta_{\mathbf{n}}^c) & 0 \\ -sin(\theta_{\mathbf{n}}^c) & cos(\theta_{\mathbf{n}}^c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -y_n^c \\ 0 & 1 & x_n^c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x^{\mathbf{c}} \\ v_y^{\mathbf{c}} \\ \omega^{\mathbf{c}} \end{bmatrix} \tag{1}$$

It can be seen from the above formula that if the motion state of the radar can be estimated, the ego-motion of the vehicle can be obtained. Fortunately, this can be achieved by exploiting the measured radial velocity and AoA information of detected stationary objects.

Given a radar point cloud $P_{t,\mathbf{n}}^{J\times M}$, assuming $M \geq 2$ containing at least radial velocity and AoA measurements, the radial velocity and AoA of the j-th detection point p_j in the radar point cloud can be expressed as $p_j = \left\{d_j^n, \alpha_j^n\right\}$, with unit $\{m/s, radians\}$. Since the radial velocity represents the radial component of the relative motion between the radar and the detected object, assuming all J detections are from stationary objects, the relationship between the radar motion state and the radial velocity measurements can be expressed as:

$$\begin{bmatrix} d_1^{\mathbf{n}} \\ d_2^{\mathbf{n}} \\ \vdots \\ d_J^{\mathbf{n}} \end{bmatrix} = - \begin{bmatrix} cos(\alpha_1^n) & sin(\alpha_1^n) \\ cos(\alpha_2^n) & sin(\alpha_2^n) \\ \vdots & \vdots \\ cos(\alpha_J^n) & sin(\alpha_J^n) \end{bmatrix} \begin{bmatrix} v_x^{\mathbf{n}} \\ v_y^{\mathbf{n}} \end{bmatrix} \tag{2}$$

For simplicity, the vector of all radial velocity measurements is denoted as D, the negative of the radial velocity projection matrix is denoted as A, and the vector of $v_x^{\mathbf{n}}$ and $v_y^{\mathbf{n}}$ is denoted as V. Then, the Equation 2 can be re-written as:

$$D = A \cdot V \tag{3}$$

Based on the above equation, two conclusions can be drawn. First, given at least two independent detection points ($J \geq 2$), it is possible to estimate $v_x^{\mathbf{n}}$ and $v_y^{\mathbf{n}}$, using standard regression approaches such as the least square method as:

$$V^{est} = (A^T A)^{-1} A^T D \tag{4}$$

However, the least square method is very sensitive to outliers. It is worth noting that outliers are extremely common in radar measurements, especially in automotive applications. Examples of outliers can be non-stationary objects such as moving cars and pedestrians, false positives, and multi-path reflections. Moreover, as explained in [65], the object height can also influence the measured radial velocity. In this work, outliers represent the measurement of radial velocities inconsistent with Equation 2.

The second conclusion is related to the observability of the vehicle's motion state. It is not difficult to see that the full 2D motion state of the vehicle cannot be estimated only by knowing $v_x^{\mathbf{n}}$ and $v_y^{\mathbf{n}}$. However, this issue can be solved by assuming that there is no lateral drifting, i.e., $v_y^{\mathbf{c}} = 0$. It is worth mentioning that this is a common assumption for automotive applications and has been widely accepted by many previous works and radar datasets [20], [41], [43], [56], [66].

Therefore, with the estimates of $v_x^{\mathbf{n}}$ and $v_y^{\mathbf{n}}$, and the radar position $\left\{x_{\mathbf{n}}^c, y_{\mathbf{n}}^c, \theta_{\mathbf{n}}^c\right\}$, the ego-motion of the vehicle in Equation 1 can be computed by:

$$v_x^{\mathbf{c}} = v_x^{\mathbf{n}} cos(\theta_{\mathbf{n}}^c) - v_y^{\mathbf{n}} sin(\theta_{\mathbf{n}}^c) + \omega^{\mathbf{c}} y_{\mathbf{n}}^c \tag{5}$$

$$\omega^{\mathbf{c}} = \frac{v_y^{\mathbf{n}} cos(\theta_{\mathbf{n}}^c) + v_x^{\mathbf{n}} sin(\theta_{\mathbf{n}}^c)}{x_{\mathbf{n}}^c} \tag{6}$$

However, the remaining question is how to estimate $v_x^{\mathbf{n}}$ and $v_y^{\mathbf{n}}$ as accurately as possible, taking into account outliers that will be present, if not prevalent, in radar point clouds.

B. System Architecture

The network architecture of the proposed ego-motion estimator (DeepEgo) is illustrated in Figure 1. In short, DeepEgo uses neural networks to extract the complex features from input radar point clouds for predicting pointwise weights and pointwise offsets, and the ego-motion is estimated based on the weighted least square (WLS) method.

As shown in the figure, the input data of DeepEgo is a multi-dimensional radar point cloud, with J points and M features. It is worth mentioning that DeepEgo is a general-purpose network and is not specially designed for automotive radars with given specifications. That said, DeepEgo will work as long as the radar is able to provide radial velocity and AoA measurements of the detected object (i.e., $M \geq 2$).

Given the input point cloud, the most challenging task for DeepEgo is to locate existing inliers. To achieve this, in the first step, a shared multilayer perception (shared-MLP) [61] is used to encode the input features and project them onto a high-dimensional feature space. The shared-MLP applies linear and nonlinear transformations individually and independently to all input detection points. Note that the shared-MLP performs a pointwise operation on the feature dimension of the input point cloud, which means that it is not sensitive to the order of the points in the input point cloud. This is important because radar point clouds are typically unordered.

After the feature encoding, the output point cloud is processed by an average pooling layer for global feature extraction. The average pooling layer is a symmetric function that aggregates information from all input points. In this way, the output feature vector can provide a global signature of the input data.

Next, the global feature vector is duplicated J times and concatenated with the input point cloud and the intermediate point cloud from the shared-MLP. As illustrated in Figure 1, the output is a $J \times F$ pointwise feature matrix, which contains not only global features of the point cloud, but also local point features from different layers. It is important to note that the mixture of local and global features is crucial to improve the accuracy of pointwise weight and offset prediction in the later stage.

Then, the feature matrix is further processed by another shared-MLP. It decodes the local-global pointwise features and transforms them into a low-dimensional feature space. After transformation, the output of the shared-MLP is sent

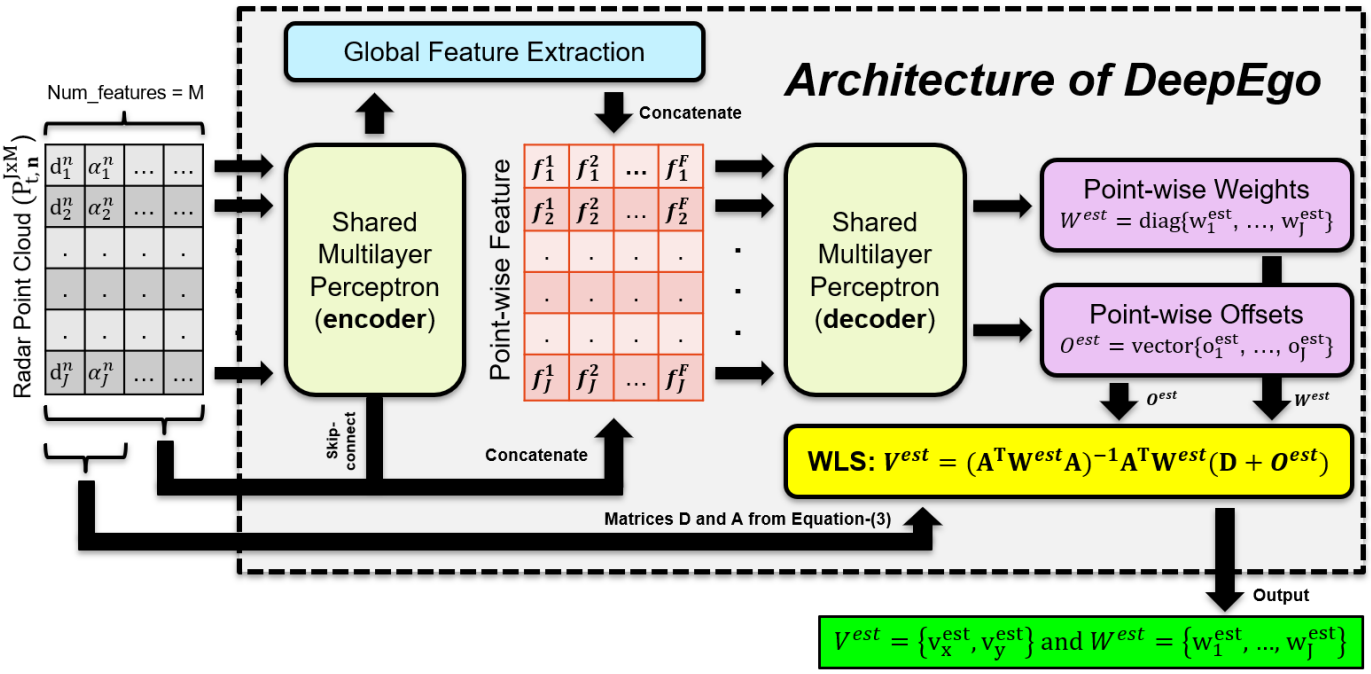


Fig. 1. The architecture of the proposed method (*DeepEgo*). The input data is the radar point cloud with J number of points and M number of features. Note that $M \geq 2$ is required in this work because radial velocity and AoA measurements are essential object features for instantaneous ego-motion estimation. The outputs of DeepEgo are the motion predictions V^{est} and pointwise weights W^{est}.

to two prediction heads for predicting pointwise weights and pointwise offsets.

Since DeepEgo relies on WLS for ego-motion estimation, the pointwise weight is the most important output as it directly scales down large fitting errors caused by outliers. The prediction of pointwise weights involves a fully connected layer with a single output neuron. The predicted weight ranges from 0 to 1 due to the sigmoid activation function used.

Inspired by [25], the pointwise offset is introduced in this work. Prior to WLS, the measured radial velocity are shifted by the predicted pointwise offset. In this way, errors caused by the 'distant' outliers, which have measured radial velocity far from expected, will be further reduced. Moreover, the network will be more robust and less sensitive to the weights of these distant outliers. Similar to pointwise weight prediction, offset prediction uses a single-neuron fully connected layer but with no activation function.

Finally, DeepEgo outputs the ego-motion of the automotive radar, and the translational and rotational velocities of the ego-vehicle can be quickly determined according to Equations 5 and 6.

C. Loss Function

This section presents a novel loss function proposed for training DeepEgo. It consists of three important components, namely motion loss, Doppler loss, and sample weights.

The *motion loss* uses the mean squared error (MSE) to measure how close the estimated motion V^{est} is to the ground truth V^{gt}. Specifically, it can be expressed as follows:

$$Loss_{Motion} = \frac{1}{B}\sum_{b=1}^{B}\left(V_b^{gt} - V_b^{est}\right)^2 \tag{7}$$

where B is the batch size. Although the motion loss is quite straightforward, it is an essential component of the proposed loss function simply because it reflects the main objective of this work, which is to achieve accurate ego-motion estimation.

While the motion loss drives the predictions as close as possible to the ground truth, it fails to explicitly instruct the network which points are outliers and should be assigned smaller weights during the WLS process. As illustrated later in Section V, a direct consequence of training using only the motion loss is that the model tends to overfit at a few points, while ignoring many inlier points. This makes the WLS process highly dependent on the accuracy of finding keypoints, and leads to performance degradation when it incorrectly flags an outlier as an inlier.

To address this issue, a novel *Doppler loss*is introduced in the proposed loss function. In a nutshell, the Doppler loss uses the ground-truth ego-motion to calculate the discrepancy between the expected and measured radial velocity, and uses that to guide the estimated pointwise weight (W^{est}).

In detail, based on Equation 3, given the ground-truth ego-motion V^{gt} and the radial velocity projection matrix A, the expected measurements of radial velocities can be written as:

$$D^{exp} = A \cdot V^{gt} \tag{8}$$

Then, the Doppler error (D^{error}) between expected and measured radial velocities is calculated as follows:

$$D^{error} = D^{exp} - D \tag{9}$$

Ideally, D^{error} should be a zero vector, if all J points are originated from stationary objects and the radial velocity and AoA measurements and ground-truth ego-motion are noise-free. However, this is not true in real-life scenarios. In order

to locate the outliers, it is common to assume that the Doppler error follows a Gaussian distribution with a mean of zero [41]. Therefore, the pointwise likelihood (W^{gt}) that detection points are inliers can be expressed as:

$$W^{gt} = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp(-\frac{(D^{error} - 0)^2}{2 * \sigma^2}) \tag{10}$$

where σ is the standard deviation for the given Gaussian distribution. In the proposed method, this is an empirical parameter to be tuned. However, it can also be approximated using the angular standard deviation and Doppler standard deviation as in [38] and [41]. Given W^{est} and W^{gt}, the Doppler loss can be computed using the MSE metric:

$$Loss_{Doppler} = \frac{1}{B}\sum_{b=1}^{B}\left(W_b^{gt} - W_b^{est}\right)^2 \tag{11}$$

It is important to point out that the success of the proposed loss function has an underlying assumption that all radial velocity measurements of detected stationary objects obey Equation 2. However, as mentioned earlier, this is not always the case. Therefore, in order to mitigate the effect of 'bad' training examples, the *sample weight* is proposed to weight each training data individually. The sample weight S is the sum of the pointwise likelihood W^{gt}, and it can be expressed as follows:

$$S = \{s_1, \ldots, s_B\} = \left\{\sum_{j=1}^{J}(W_{1,j}^{gt}), \ldots, \sum_{j=1}^{J}(W_{B,j}^{gt})\right\} \tag{12}$$

It is clear that the proposed sample weight tends to assign higher importance to training examples with more points that are consistent with expected radial velocity measurements (or ground truth). This can significantly mitigate negative effects caused by, for example, radar point clouds with only outliers, inaccurate ground-truth, or non-zero lateral velocities and non-zero object heights.

Finally, combining all components together, the proposed loss function for the whole network can be written as:

$$Loss_{all} = (Loss_{Motion} + \mu \cdot Loss_{Doppler}) \cdot S \tag{13}$$

where μ is a weighting factor that should be adjusted empirically.

D. Implementation Details

To reproduce the proposed method, the following implementation details are provided:

1) **Shared-MLP**: This work uses two shared-MLPs for point cloud feature encoding and decoding. The shared-MLP consists of three layers of fully connected neural networks (FCNNs) with an encoder of size (128, 256, 512) and a decoder of size (512, 256, 128). Each FCNN is followed by a Batch Normalization (BN) [67] layer and a non-linear activation layer. Unless stated otherwise the rectified linear unit (ReLU) [68] is used as the activation function in this work.
2) **Data Resampling**: Due to fluctuations in RCS and mm-wave target scattering behavior, the radar point cloud has different sizes, i.e., J varies with time. Therefore, the collected radar point cloud is randomly up- or down-sampled to a fixed number of points of 256. Moreover, to eliminate point clouds with too few points to learn, a threshold is applied before resampling, which removes point clouds with less than 30 points.
3) **Data Normalization**: For the multi-dimensional radar point cloud, $M \geq 2$ is required. This is because the radial velocity and AoA are necessary object features for instantaneous ego-motion estimation. In this work, the data units for radial velocity and AoA are m/s and radians, respectively, and no data normalization is applied to these two features. However, other features such as range and return power are recommended to be rescaled using approaches like min-max normalization.
4) **Network Training**: For optimizing the loss function, the mini-batch gradient descent method is used, with a batch size of 512 ($B = 512$). The learning rate starts at 1e-3 and the descent direction is adaptively adjusted by the root mean squared propagation (RMSProp). For validation, 20% of the training data is separated out, and the training is stopped when the validation loss stops decreasing over 50 epochs. Finally, it is worth mentioning that the proposed network considers the general case of a moving vehicle equipped with one radar. If multiple radars are present, a separate copy of the proposed network needs to be trained for each radar.

IV. Results and Discussion

This section presents the evaluation results of the proposed ego-motion estimation method.

A. Radar Dataset

To measure the performance of the proposed method in various driving scenarios, RadarScenes [26], a real-world radar point cloud dataset, is used. As shown in Figure 2, the dataset contains recordings from four automotive radar sensors mounted on the testing vehicle. The dataset covers large variations in ego-vehicle motion, road conditions, illumination, and dynamic objects. In total, each radar captures 158 separate sequences based on different road scenarios and times.

Besides various driving road scenes, the RadarScenes dataset also focuses on different road users. However, in order to record road users such as pedestrians and cyclists, the ego vehicle was set to be stationary during some data sequences. Since the main objective in this work is to estimate the ego-motion of the vehicle, data sequences with static ego-vehicle will preclude a comprehensive evaluation. Therefore, this work uses data sequences with driving distances over 500 meters. This yielded 64 data series in total, with an equivalent drive length of over 79 kilometers.

Lastly, the number of input features is equal to four (i.e., $M = 4$ as illustrated in Figure 1), since point clouds in the selected dataset are four-dimensional, including AoA,

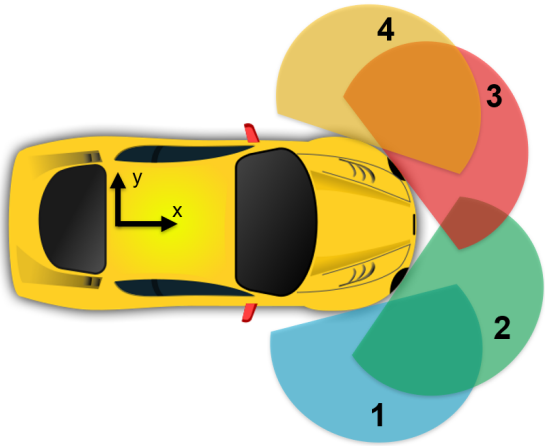


Fig. 2. An illustration of the testing vehicle and four automotive radars used for collecting the RadarScenes dataset [26]. As described in the dataset, the four radars have a maximum detection range of 100m, a field of view of about ±60deg, a range resolution of 0.15m, an angular resolution of 0.5deg at the boresight direction, and a radial velocity resolution of 0.1km/h. Regarding to the pointing direction, radar #1 and #4 are side-looking tilted 85deg outwards, while radar #2 and #3 are front-facing radars tilted 25deg outwards.

Doppler/velocity, range, and returned power measurements of detected objects. For more details on the dataset, the reader is referred to [26].

B. Evaluation Metrics

Benchmarking different ego-motion estimation methods requires ground-truth poses of the ego-vehicle and suitable evaluation metrics. Conventionally, metrics for evaluating pose estimation accuracy are widely used [20], [38], [39]. For automotive applications, it is also important to understand the long-term estimation performance of the ego-motion estimation method. Therefore, metrics such as Absolute Trajectory Error (ATE) [23] are used to quantify the cumulative error. However, such metrics are usually computed over the entire trajectory and are more sensitive to early errors [41]. Therefore, in order to conduct a comprehensive performance analysis, this work introduces two evaluation metrics:

1) **Absolute Pose Error (APE)** measures the pose difference between the estimation and the true motion. Specifically, it calculates the root-mean-square-errors (RMSE) between the estimated poses and the ground-truth poses. Thus, given an estimate, it can be used to indicate the estimation accuracy of the method under test. In this work, the final outcome of the APE is the translational error and the rotational error denoted by 'APE_Trans' and 'APE_Rot', respectively.
2) **Relative Trajectory Error (RTE)** is similar to ATE, which also measures the long-term stability of the method under test. However, instead of processing the entire trajectory, RTE first divides the estimated trajectory and ground-truth trajectory into multiple segments according to a predefined time interval (or drive length). Then, between each paired segment, the starting positions in the 2D space are aligned and the euclidean distance between the ending points is calculated. Finally, RTE is equal to the mean square of the computed euclidean distances of all pairs of trajectory segments. In this work, a variety range of driving distances are used for evaluation. For simplicity, the RTE result for a drive length of X meters is denoted in the rest of the paper by 'RTE_X', where X indicates a given drive length.

In summary, APE can directly indicate the estimation accuracy of the method under test. However, it cannot reflect the drifting effect, and large estimation errors can be hidden by the averaging process. Therefore, this work employs an additional evaluation metric, RTE. As discussed above, this provides a means to analyze the drifting stability of the testing method over a predetermined drive length.

C. Selected Methods for Comparison

Ego-motion estimation is an important topic in robotics and autonomous driving. However, due to the challenges mentioned in the Introduction, there is not much work on ego-motion estimation using only automotive radars. The following works selected from the literature cover a broad range in terms of methodology and are considered as the state-of-the-art (SOTA) for performance comparison:

1) **Biber's Method [33]** is the first NDT-based method. It uses local normal distributions to model uncertainty in measurements without explicitly establishing associations between scans. It is an iterative method, the score function is piecewise continuous and differentiable, and can be optimized using, for example, Newton's method.
2) **Kellner's Method [20]** is an instantaneous method that requires only one scan for ego-motion estimation. However, the estimation process is still iterative as it relies on the RANSAC algorithm [46] to mitigate the effects of outliers. Unlike NDT-based methods, Kellner's method solves the problem of parametric curve fitting, and the input data are the AoA and radial velocity measurements.
3) **Rapp's Method [41]** is an improved clustering-based NDT method with a spatial registration component and a likelihood model for radial velocity measurements. The method is an enhancement of the previous works [38], [39]. It is reported that Rapp's method outperforms the previous work [38] when using a single radar sensor, but the estimation accuracy is lower than Kellner's method.
4) **Lu's Method [23]** is the first deep model-based method for ego-motion estimation using automotive radar and IMU. It proposed a CNN-based feature extractor subnet to process mmWave radar images and extract ego-motion features. The mmWave subnet has shown better performance than two conventional ICP-based methods [28] and [69].
5) **Heller's Method [42]** extends traditional 2D-NDT methods to 3D geometries in polar coordinates. The extra dimension is used to incorporate the radial velocity measurements. Moreover, the returned power of detected objects has been used for constructing the normal distribution and calculating the score function. This method reported better performance than conventional NDT methods.

6) **Kung's Method [43]** presents a 2D-NDT based signal processing pipeline that can be used to process data from scanning and automotive radar. The method uses multiple scans to construct a probabilistic radar submap and returned power for measurement thresholding. It reported better performance than Kellner's method [45], Rapp's method [41], and conventional ICPs [13], [70].

As shown above, there is one instantaneous method based on AoA-velocity profile analysis (Kellner's), one scan-matching method based on deep neural network (Lu's), and four scan-matching methods based on NDT. Due to the redundancy of NDT-based methods, it is reasonable to compare them first and select the most competitive NDT method for further comparison with the proposed method and others from the SOTA.

Since the main difference between the NDT-based methods resides in their proposed score function, Figure 3 shows the values of the four score functions generated by grid search. First, it is clear that the distance between the global maximum and the ground-truth pose in Figure 3-(d) are closer than those in Figure 3-(a). However, neither Biber's method nor Kung's method can determine the translational velocity of the ego vehicle. In contrast, Rapp's and Heller's method can indicate the translational and rotational velocity of the vehicle more accurately, thanks to the additional information of radial velocity measurement. Furthermore, it is apparent that with proper initialization, Rapp's method is more likely to converge to the global maximum than Heller's method. Therefore, among the four NDT-based methods, Rapp's method is selected for further comparison with the proposed method.

Although this is the first work comparing the four NDT-based methods using the same dataset, a detailed failure analysis is beyond the scope of this paper due to space limitations. Nevertheless, the parameterizations of the compared works are presented in Appendix-A, and readers interested in these methods are also advised to read the original references.

D. Results of Comparison With SOTA Methods

Table I shows the five scenes selected for the SOTA comparison. These scenes cover a variety of weather, traffic, and road type conditions. Each scene contains data sequences from the four automotive radars mounted on the vehicle (i.e., 20 data sequences in total for testing from four radars and five scenes). For deep learning-based methods, these scenes are used for testing after model training according to the leave-one-out (L1O) method. In this way, they are 'unseen' data for the models, which is crucial for measuring their ability to generalize.

Table II presents the APE of translational velocity estimation. In this case, Lu's method scores the worst; a potential reason is that Lu's method does not specifically mitigate the influence of dynamic objects in the scene. For Rapp's method, when the radar '#2' or '#3' is used, the error in translational velocity estimation is small. This is because these two radars are front-facing, hence, they are more sensitive to vehicle speed than the radar '#1' or '#4', which are side-looking. Clearly, Kellner's method has the lowest APE in translational velocity estimation among the three methods chosen for comparison. However, the proposed approach still outperforms Kellner's method, regardless of the position of the radar on the vehicle. On average, the proposed solution is about **37.9%** better than Kellner's method.

TABLE I

THE 5 SELECTED SCENES USED FOR PERFORMANCE EVALUATION WITH SOTA METHODS. EACH SCENE HAS DATA SEQUENCES FROM 4 AUTOMOTIVE RADARS. FOR DEEP LEARNING-BASED METHODS, THESE DATA SEQUENCES ARE NOT USED FOR MODEL TRAINING AND VALIDATION

Index	Surface	Weather	Traffic	Length	Road Types
#89	Bumpy	Rain	No	≈752m	Collector Road
#67	Smooth	Clear	Little	≈668m	Local Street
#108	Smooth	Clear	No	≈2281m	Arterial Road
#10	Smooth	Cloud	High	≈1443m	Collector Road
#138	Smooth	Clear	Medium	≈700m	Collector Road

TABLE II

THE ABSOLUTE POSE ERROR OF TRANSLATIONAL VELOCITY ESTIMATION. THE UNIT IS METERS PER SECOND. THE RESULTS FOR EACH RADAR ARE AVERAGED OVER THE FIVE TEST SEQUENCES

Methods	Radar #1	Radar #2	Radar #3	Radar #4	Mean
Kellner's	0.135	0.0904	0.145	0.390	0.190
Rapp's	1.268	0.474	0.692	1.296	0.933
Lu's	3.207	2.508	2.807	2.918	2.860
Proposed	**0.0955**	**0.0877**	**0.0876**	**0.203**	**0.118**

TABLE III

THE ABSOLUTE POSE ERROR OF ROTATIONAL VELOCITY ESTIMATION. THE UNIT IS DEGREES PER SECOND. THE RESULTS FOR EACH RADAR ARE AVERAGED OVER THE FIVE TEST SEQUENCES

Methods	Radar #1	Radar #2	Radar #3	Radar #4	Mean
Kellner's	0.785	1.014	2.315	0.894	1.255
Rapp's	7.563	13.866	14.668	8.079	11.058
Lu's	5.031	5.546	4.973	4.641	5.048
Proposed	**0.642**	**0.859**	**0.911**	**0.653**	**0.768**

Table III shows the APE of rotational velocity estimation. First, it can be noted that all tested methods perform relatively poorly on both front-facing radars. This is because the side-looking radar is more sensitive to the angular motion of the ego-vehicle. Nevertheless, the proposed method shows the lowest APE compared to the SOTA methods. On average, the estimation accuracy increased by **38.8%** compared to the second-best method.

The previous results demonstrate the accuracy of the proposed method in translational and rotational velocity estimation. However, as discussed earlier, the APE metric cannot quantitatively measure the effect of drifting. Therefore, an additional evaluation metric, RTE, is introduced in this work. Figure 4 shows the mean-squared RTE for the four tested methods. To better understand the long-term stability of the tested method, three different driving distances were chosen for the RTE metric, namely 10m, 20m, and 30m. Based on the figure, it is evident that RTE is more affected by the error in rotational velocity estimation than in translational velocity estimation, although it should be a combined effect. Still, the proposed method scores the best among the tested methods, regardless of the radar position and the chosen challenging scenes. Finally, it is worth mentioning that although

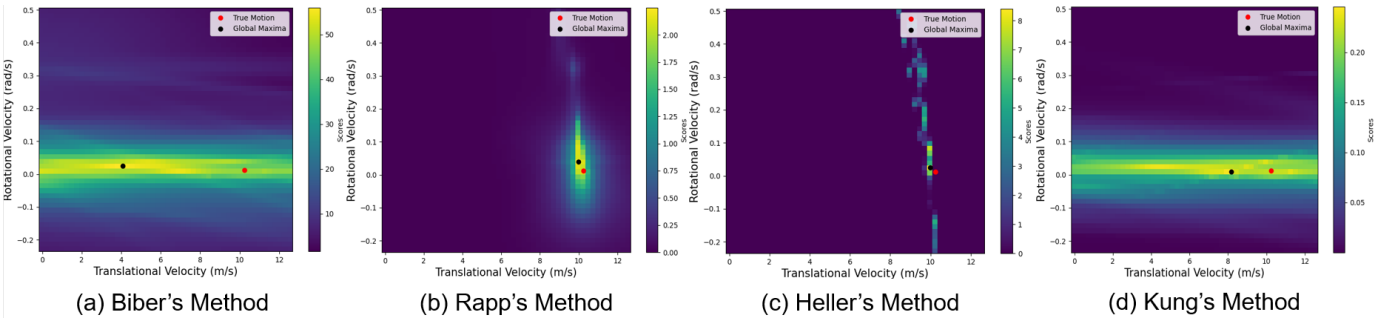


Fig. 3. Results of the score functions of the four selected NDT-based methods. The scores are calculated using the exhaustive grid search in the translational velocity and rotational velocity space. In the above plots, the location of the ground-truth pose is represented by a red dot, and the global maximum of the score function is denoted by a black dot.

the proposed method outperforms Kellner's method on the APE metric by a large margin, Kellner's method closely follows the proposed method on the RTE metric.

For real-time applications such as autonomous driving, ego-motion estimation methods need to have good runtime performance. To illustrate this, Figure 5 presents the update rate of the tested methods. As shown in the figure, Rapp's method is the slowest with 5 frames per second (FPS) only, although 17 FPS was recorded in [41] with a better CPU. This is a reasonable update rate because Rapp's method has to perform an iterative optimization process for every measurement scan. On the other hand, Kellner's method and Lu's method have similar runtime performance, with 215 FPS and 210 FPS, respectively. For Kellner's method, there is no optimization process, but each scan requires a certain number of iterations due to the random sampling mechanism [46]. For Lu's method, although the ego-motion estimation is direct and non-iterative, it requires a lot of computation due to its large network size. Finally, it is evident that the proposed method is the fastest. Compared with Rapp's method and Kellner's method, the proposed method is straightforward and has no iterative process. Furthermore, the proposed neural network is lightweight with about 800K trainable parameters compared to Lu's method with nearly 15M parameters. Lastly, it is important to note that this experiment assumes that the data collection has already been done. For real-world situations, the runtime performance of the tested method can be limited by factors such as the radar update rate.

Finally, in order to visualize the long-term stability of the tested method, Figure 6 shows the trajectories computed by integrating the estimated ego-motion based on the sampling timestamp. For the comparison, all test methods use the same global reference point, and the ground-truth trajectory is also plotted. As shown in the plot, Lu's method can hardly follow the true trajectory of the ego-vehicle. This can be attributed to the discretization process, as it inevitably destroys the accurate position of the measurements in the radar point cloud. However, in order to use CNNs, the discretization process is unavoidable. Rapp's method performs slightly better than Lu's method, but the drift is still very prominent. The reason behind this can be the 'tunnel' effect caused by the ubiquitous road curb, which leads to the spatial likelihood [41] assigning a large weight to zero ego-motion. On the other hand, Kellner's method shows better long-term stability compared to other SOTA methods. It is able to mitigate the effects of non-stationary objects due to the random sampling mechanism [46]. However, in each iteration, it randomly samples a few measurement points and assigns equal weights for estimation. Compared to the proposed method, as also shown in Table II and III, this can lead to large estimation errors if the selected measurements are not noise-free. Finally, it is visually evident that our proposed method has less drift effects compared to the other SOTA methods, especially when using the side-looking radar.

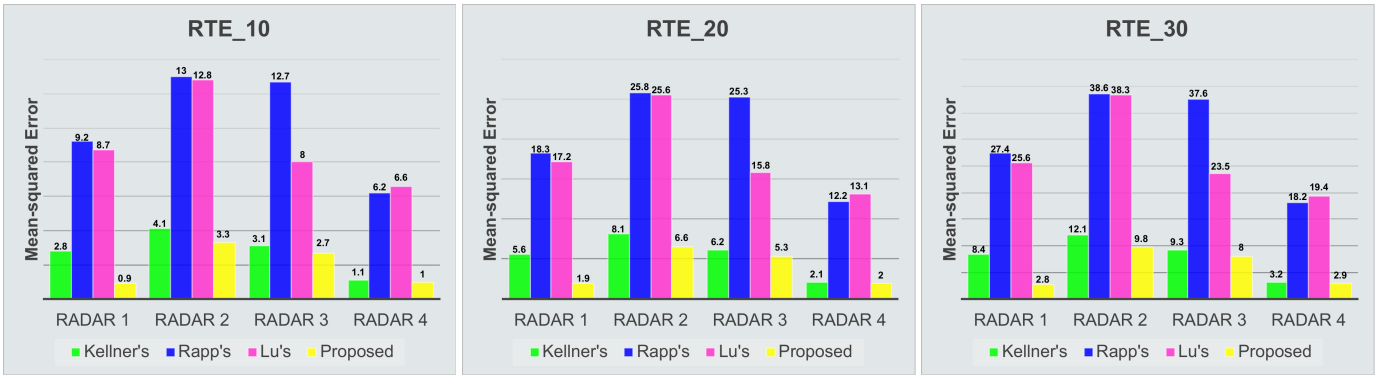
E. Further Results on All Data

The previous sections evaluated seven ego-motion estimation methods using the five selected scenes collected by four automotive radars. The results have shown that the proposed method provides the best estimation accuracy, long-term stability, and runtime performance. To achieve a comprehensive analysis, in this section further comparisons are conducted using all 64 data sequences, with an equivalent drive length of over 79km. However, only Kellner's method is used for comparison since it is the most competitive and scored the second-best performance among the other approaches.

Table IV presents a full-scale performance evaluation measured using the APE_Trans, APE_Rot, and RTE_50 metrics; the results are averaged over 64 testing sequences for each radar. For the proposed method, the leave-one-out (L1O) approach is used for training and testing across all the sequences. Based on the results, it is evident that the proposed method significantly outperforms Kellner's method. On average, the proposed method has **51.2%** improvement on APE_Trans, **49.8%** improvement on APE_Rot, and **23.2%** improvement on RTE_50.

While these evaluation metrics qualitatively indicate the performance of the tested method, it is difficult to understand where the errors come from. Therefore, Figure 7 presents the 2D histogram of estimation errors accumulated over 64 data sequences for each radar. The first row shows the histograms of Kellner's method, while the second row gives the histograms of our proposed method.

At first glance, it is clear that a large amount of errors occurs in the region where the rotational velocity is close to zero. This is caused by an imbalanced dataset, as the ego vehicle goes



(a) RTE with 10m drive length. (b) RTE with 20m drive length. (c) RTE with 30m drive length.

Fig. 4. The mean-squared RTE of the methods under comparison. In this experiment, three driving distances, 10m, 20m, and 30m, were selected for the RTE metric. For each radar, the result is the average of the five test data sequences.

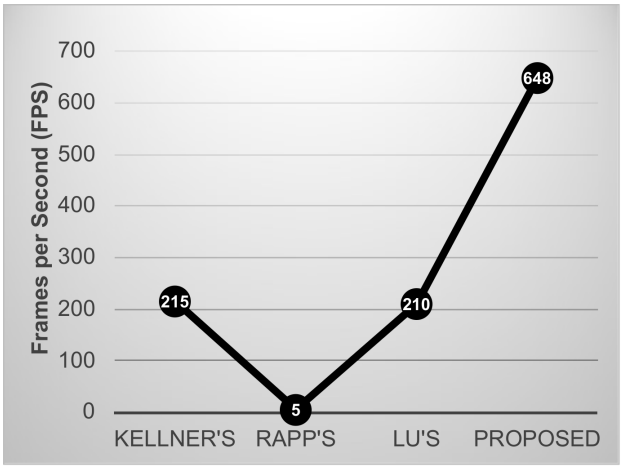


Fig. 5. Update rate of the methods under test. All methods are tested under the same system environment using the Delft High Performance Computing Center (DHPC) *DelftBlue* [71]. In this experiment, neither GPU nor parallel computing is applied. Instead, a single Intel Xeon compute node is used.

TABLE IV

A FULL-SCALE PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND KELLNER'S. FOR EACH RADAR, RESULTS ARE AVERAGED OVER 64 DATA SEQUENCES. FOR THE PROPOSED METHOD, THE LEAVE-ONE-OUT APPROACH IS USED DURING MODEL TRAINING, IN ORDER TO MEASURE THE ESTIMATION ERROR FOR THE TESTING SEQUENCE THAT IS UNSEEN TO THE TRAINED MODEL

APE_Trans (m/s)					
Methods	Radar #1	Radar #2	Radar #3	Radar #4	Mean
Kellner's	0.193	0.132	0.253	0.363	0.235
Proposed	0.108	0.0976	0.112	0.142	0.115
Improve.	**44.0%**	**26.1%**	**55.7%**	**60.9%**	**51.2%**
APE_Rot (deg/s)					
Methods	Radar #1	Radar #2	Radar #3	Radar #4	Mean
Kellner's	0.802	2.172	2.934	0.882	1.697
Proposed	0.607	1.071	1.049	0.682	0.852
Improve.	**24.3%**	**50.7%**	**64.3%**	**22.7%**	**49.8%**
RTE_50 (m^2)					
Methods	Radar #1	Radar #2	Radar #3	Radar #4	Mean
Kellner's	11.7	18.8	21.2	6.1	14.5
Proposed	6.5	15.1	17.4	5.5	11.1
Improve.	**44.7%**	**19.7%**	**18.0%**	**10.4%**	**23.2%**

straight more often than making turns. Nevertheless, compared to Kellner's method, the proposed method has appreciably smaller errors when the vehicle is traveling at high speeds or cornering quickly. In addition, from the perspective of road safety, reducing the estimation errors in high-speed areas is more crucial and meaningful than in low-speed areas.

TABLE V

THE IMPACT OF DIFFERENT SELECTED INPUT POINT CLOUD FEATURES. A TOTAL OF 4 EXPERIMENTS ARE PERFORMED AND COMPARED WITH THE PROPOSED METHOD WITHOUT ANY ABLATION. AMONG THEM, 'WO_RANGE_WO_POW' MEANS THE INPUT RADAR POINT CLOUD HAS NO RANGE AND POWER/INTENSITY INFORMATION OF THE DETECTED OBJECT, AND 'WO_SAMPLE_WEIGHT' MEANS THAT THE PROPOSED SAMPLE WEIGHT IS NOT USED DURING MODEL TRAINING

Condition	Train_loss	Val_loss	APE_Trans	APE_Rot
wo_range_wo_pow	3.31E-3	7.56E-3	0.118	0.557
w_range_wo_pow	1.04E-3	5.44E-3	0.101	0.513
wo_range_w_pow	1.05E-3	3.71E-3	0.0817	0.479
wo_sample_weight	9.02E-4	2.14E-3	0.0551	0.533
Proposed	**6.18E-4**	**1.19E-3**	**0.0421**	**0.441**

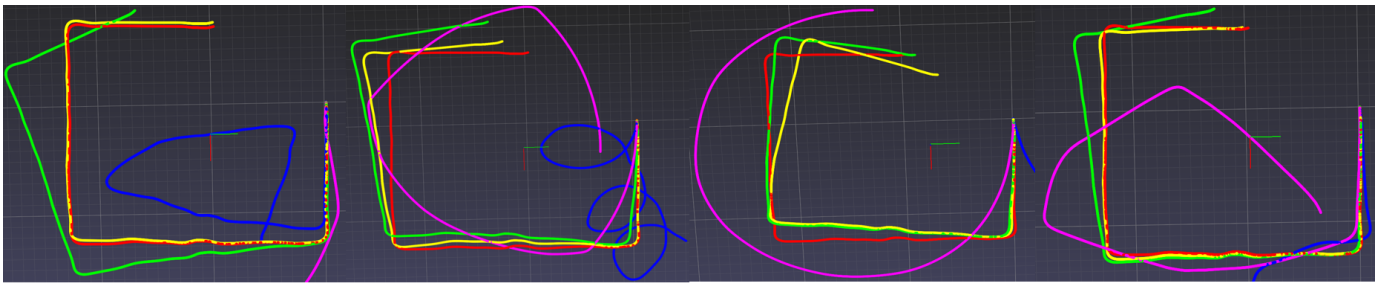
V. ABLATION STUDY

In this section, an ablation study is presented in order to illustrate the impact of the design choices made in the proposed method.

A. Ablation Study on Input Features

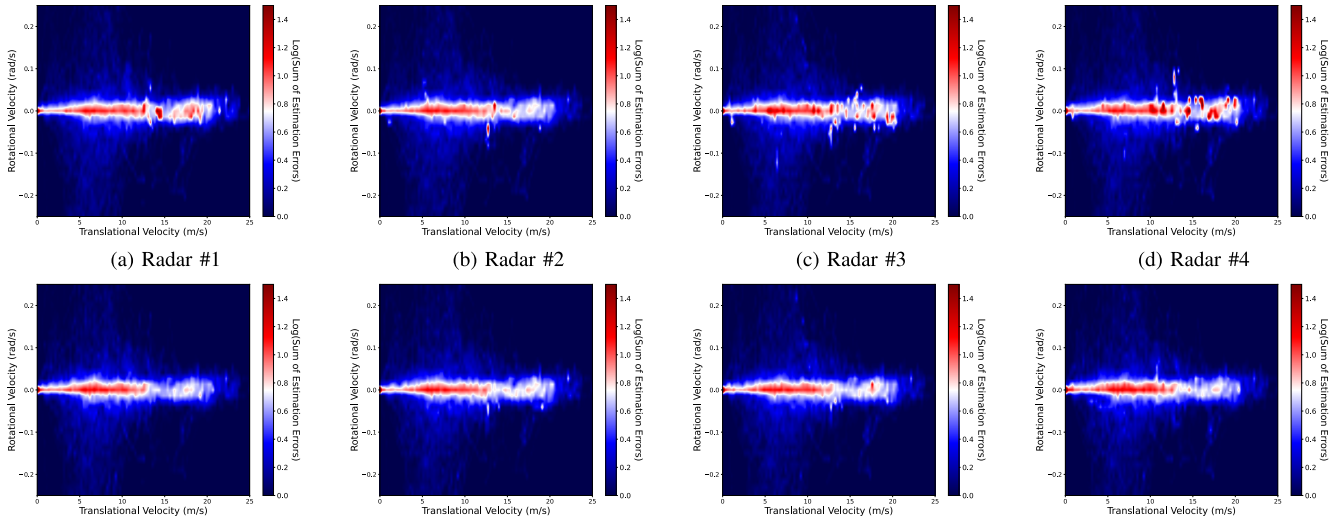
Conventionally, ego-motion estimation methods based on scan-matching only use the range and AoA measurements, and methods based on parametric curve fitting use AoA and radial velocity measurements. However, radar point cloud often contains multi-dimensional features of the detected object. For instance, point clouds in the RadarScenes dataset [56] are four-dimensional and include AoA, radial velocity, range, and returned power/intensity measurements of objects. Although the proposed method provides a straightforward approach to deal with multi-dimensional radar point clouds, it is also important to argue about the usefulness of incorporating different features of the detected object for ego-motion estimation.

Table V presents the ablation study on the input features. In detail, the proposed model is trained multiple times with different input features being discarded. However, since the AoA and radial velocity measurements are indispensable



(a) Radar #1 (b) Radar #2 (c) Radar #3 (d) Radar #4

Fig. 6. The trajectory of the ego-vehicle in Scene #67. The trajectory is calculated by integrating the estimated ego-motion given the timestamps. In the plot, red line represents the **ground-truth** trajectory, yellow line represents our **proposed** method, green line represents **Kellner's** Method, pink line represents **Lu's** Method, and blue line represents **Rapp's** Method.



(a) Radar #1 (b) Radar #2 (c) Radar #3 (d) Radar #4

(e) Radar #1 (f) Radar #2 (g) Radar #3 (h) Radar #4

Fig. 7. The histogram of the estimation error accumulated over 64 data sequences (an equivalent drive length of over 79km). The plots in the first row show the results of Kellner's method, and the second row shows the results of the proposed method. Due to the imbalanced ego-motion distribution in the dataset, the natural logarithm is applied to the sum of the estimation error for better visualization. For the proposed method, the leave-one-out approach is used during model training, in order to measure the estimation error for the testing sequence that is unseen to the trained model.

information for the proposed method, they are not ablated. Based on the results, it can be concluded that both range and power/intensity measurements help improve estimation accuracy. Moreover, the performance can be further improved when all measured features are used, as shown by the results labelled as 'Proposed'.

Nevertheless, the quality of the input feature can be affected by many other factors. For example, as discussed in Section III, the measured Doppler/velocity of a detected object can be influenced not only by the motion relative to the ego-vehicle, but also by measurement error, object height, and lateral velocity. Therefore, the expected radial velocity measurement, computed based on the ground-truth ego-motion, may not match with the measured radial velocity perfectly. Consequently, this can confuse the optimization process as the neural network tries to learn features that are invariant between different inputs.

To address this issue, an extra penalty term is added to the proposed loss function to weight the training data. As shown in Table V, the proposed model achieves more accurate pose estimation and scores better in both training and validation losses compared with training without such sample weight.

B. Effect of Pointwise Offset

In contrast to previous works that considered using simulated radar data or data collected under controlled environments, this work uses a challenging real-world radar dataset containing a variety of driving scenarios and road conditions. Therefore, the dataset contains not only stationary objects, but also a significant number of moving objects, false positives, and ghost objects. In order to address these real world non-idealities, the proposed neural network learns to regress a set of pointwise weights and pointwise offsets. In the weighted least square procedure, the pointwise weight is used to scale down errors caused by outliers, and the pointwise offset shifts radial velocity measurements that are far from expected values. In another word, the pointwise offset aims to further reduce large errors caused by 'distant' outliers, e.g., measurements from a car that is moving fast toward the ego-vehicle.

Table VI shows the impact of the predicted pointwise offset. In this experiment, the proposed method is tested on the five selected data sequences, first using the predicted pointwise offset. Then, as a comparison, the predicted pointwise offset is removed and the model is rerun using the test data. Based

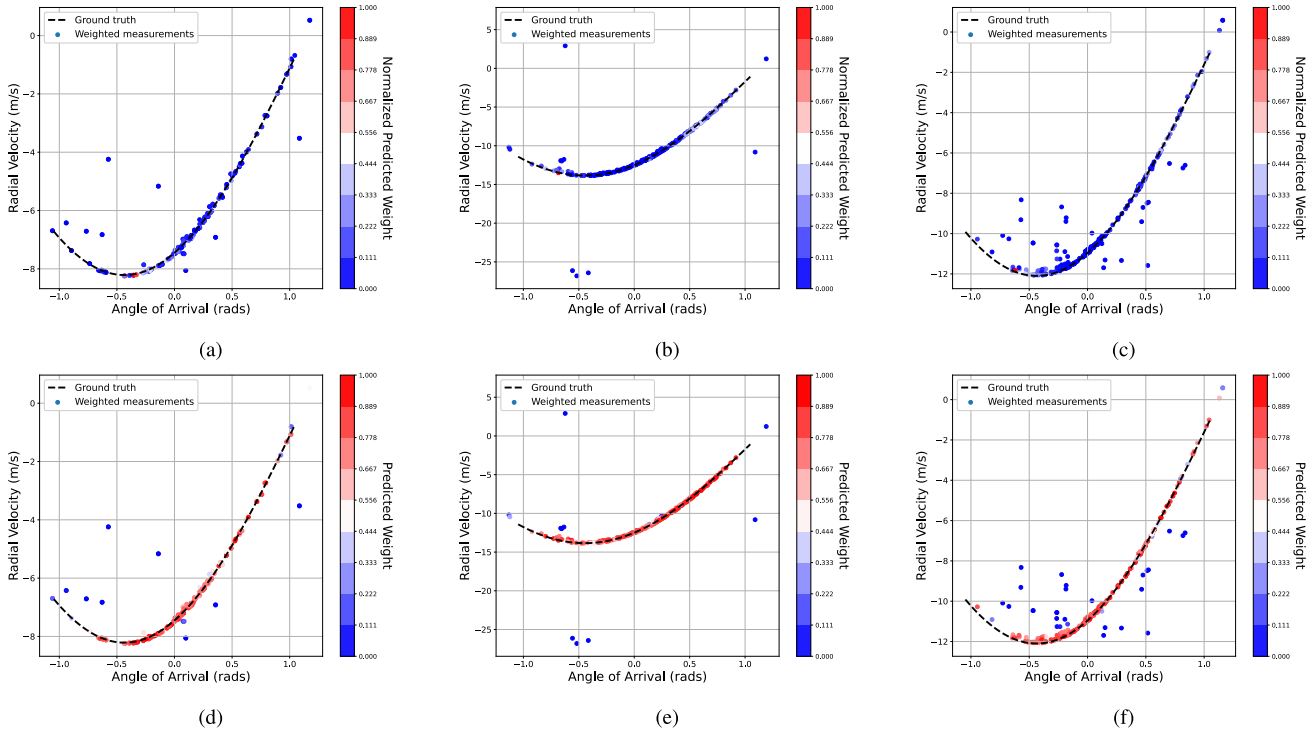


Fig. 8. The effect of the proposed Doppler loss. For illustration purposes, three frames of radar data are randomly selected and presented as AoA-Radial Velocity plots. The figures in the first row are the results of the model trained *without* the Doppler loss, while the results in the second row are trained *with* the Doppler loss. The black dashed line represents the expected radial velocity, given the ground-truth ego-motion and the AoA of the detected object. Measured radial velocities are represented by dots, and colors indicate the magnitude of predicted weights. The weights in the first row are normalized by the maximum weight value, while the weights in the second row are direct outputs of the proposed method.

TABLE VI

THE EFFECT OF POINTWISE OFFSET. THIS EXPERIMENT IS CONDUCTED AFTER MODEL TRAINING, AND TESTING USING THE FIVE TEST SEQUENCES. SPECIFICALLY, THE 'WO_DOPPLER_OFFSET' RE-RUNS THE TEST SEQUENCES AND USES ONLY THE PREDICTED WEIGHT FOR EGO-MOTION ESTIMATION, WHILE THE 'PROPOSED' USES THE PREDICTED WEIGHT AND POINTWISE OFFSET

Condition	APE_Trans	APE_Rot	RTE_50
wo_pointwise_offset	0.0903	0.991	12.589
Proposed	0.0884	0.928	10.118
Improvement (%)	**2.1%**	**6.4%**	**19.6%**

on the results, it is clear that the pointwise offset not only improves the estimation accuracy, but also improves long-term stability. However, the performance improvement is not as significant as in the previous ablation study (i.e., Table V). This is because the pointwise weight plays the most influential role in the linear regression process.

C. Effect of Doppler Loss

The main objective of this work is to estimate the motion of the ego-vehicle. To achieve it, a hybrid model is proposed. In detail, a neural network is used to exploit the multi-dimensional radar point cloud and provide pointwise weights for each measurement, and the weighted least square approach is applied to estimate the ego-motion. Therefore, it is obvious that knowing which measurements are more important than others is the key capability that the proposed model should learn during training.

One simple solution to make the model learn where to 'focus' is by using the motion loss as mentioned in Section III. However, if not explicitly instructed, the model will learn to overfit at a few key points. As shown in the first row of Figure 8, although many measurements should be considered valid and have high weights, almost all of them have actually small weights if the proposed Doppler loss is not used.

As a comparison, the results from the model trained with the proposed Doppler loss are shown in the second row of Figure 8. It is clear that the model has learned to assign high weights to measurements that are close to the ground-truth curve (denoted by the black dashed line). Compared to relying on a few points, using more measurements for estimation increases the robustness of the estimator. This can also be seen from the RTE metric, which is improved by about **19.5%** compared to the model trained without the Doppler loss. Furthermore, with the Doppler loss, the proposed method not only can provide the instantaneous ego-motion of the vehicle, but also can directly indicate the likelihood of measurements originating from stationary objects. Therefore, it is important to note that the predicted weight, as a byproduct of ego-motion estimation, can help other downstream applications such as environment mapping, multiple object tracking, instance segmentation, and others yet to be explored.

VI. CONCLUSION

A novel deep learning-based method, *DeepEgo*, for instantaneous ego-motion estimation using automotive radar is proposed. Unlike previous works, DeepEgo can directly process multi-dimensional radar point clouds with minimal

pre-processing steps required. In detail, DeepEgo consists of a hybrid architecture that uses neural networks (NN) for feature extraction and weighted least squares (WLS) for ego-motion estimation. In addition, DeepEgo is fully differentiable and can be trained end-to-end without manual annotation. Moreover, DeepEgo can provide pointwise weights for each point in the radar point cloud. Thanks to the newly proposed Doppler loss function, the predicted weights are associated with the likelihood that the detection point comes from a stationary object. DeepEgo was tested on a challenging real-world radar dataset, demonstrating its superior performance in terms of estimation accuracy, long-term stability, and computation time, compared to other methods selected from the literature.

For future research, it is important to acknowledge that this work focuses on ego-motion estimation based on a single type of sensor. Although radar sensor fusion is currently a well-researched topic [72], [73], [74], the capabilities and potential of automotive radar for ego-motion estimation remain to be fully explored. Therefore, various fusion approaches (e.g., homogeneous and heterogeneous sensor fusion) are not discussed in this work and are reserved for future research. Other interesting future work directions could be to explore other regression methods such as orthogonal distance regression (ODR) [75], or to improve the robustness of the current work to different radar installation locations.

APPENDIX A
PARAMETERIZATION OF THE COMPARED WORKS

Kellner's Method [20]: Due to its fast iteration time, the grid search method is implemented based on five selected scenes (Table I) to find the best parameter combinations. As a result, the number of anchor points is set to 5, the inlier ratio is set to 0.6, the inlier probability is set to 0.99, and the value of the corridor threshold is set to 0.1.

Rapp's Method [41]: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is used to represent the reference scan as Gaussian mixtures. The clustering radius is 1.4m as suggested by [41], and the minimum number of samples is set to 3. To optimize the proposed objective function, the gradient descent algorithm is used. The maximum eigenvalue ratio is set to 1e3 to prevent non-invertible covariance matrices.

Lu's Method [23]: Since Lu's method considers heterogeneous sensor fusion, only the mmWave sub-network is used to process the given radar data. The architecture of the sub-network is the same as suggested in their original paper with nine convolutional layers. Also, the FC Pose Regressor is used to process the output of the subnet and computes the pose estimate. Since the used radar point clouds have no elevation information, they are converted to 2D bird's eye view images instead of 2D "depth" images.

REFERENCES

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[2] SAE On-Road Automated Vehicle Standards Committee, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," *SAE Standard J.*, vol. 3016, pp. 1–16, Jan. 2014.

[3] S. Ayyasamy, "A comprehensive review on advanced driver assistance system," *J. Soft Comput. Paradigm*, vol. 4, no. 2, pp. 69–81, Jul. 2022.

[4] S. Behere and M. Törngren, "A functional architecture for autonomous driving," in *Proc. 1st Int. Workshop Automot. Softw. Archit. (WASA)*, May 2015, pp. 3–10.

[5] S. Liu, J. Tang, Z. Zhang, and J. Gaudiot, "Computer architectures for autonomous driving," *Computer*, vol. 50, no. 8, pp. 18–25, 2017.

[6] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 869–880, Dec. 1996.

[7] J. Yi, J. Zhang, D. Song, and S. Jayasuriya, "IMU-based localization and slip estimation for skid-steered mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 2845–2850.

[8] Y. Gu, Y. Wada, L. Hsu, and S. Kamijo, "Vehicle self-localization in urban canyon using 3D map based GPS positioning and vehicle sensors," in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, Nov. 2014, pp. 792–798.

[9] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun./Jul. 2004, p. I.

[10] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. X*, vol. 2, no. 9, Berkeley, CA, USA, Jul. 2014, pp. 1–9.

[11] A. Xenaki, B. Gips, and Y. Pailhas, "Unsupervised learning of platform motion in synthetic aperture sonar," *J. Acoust. Soc. Amer.*, vol. 151, no. 2, pp. 1104–1114, Feb. 2022.

[12] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6045–6052.

[13] E. Ward and J. Folkesson, "Vehicle localization with low cost radar sensors," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 864–870.

[14] Z. Hong, Y. Petillot, and S. Wang, "RadarSLAM: Radar based large-scale SLAM in all weathers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5164–5170.

[15] E. Jose and M. D. Adams, "An augmented state SLAM formulation for multiple line-of-sight features with millimetre wave RADAR," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3087–3092.

[16] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-wave technology for automotive radar sensors in the 77 GHz frequency band," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 3, pp. 845–860, Mar. 2012.

[17] C. Waldschmidt, J. Hasch, and W. Menzel, "Automotive radar—From first efforts to future systems," *IEEE J. Microw.*, vol. 1, no. 1, pp. 135–148, Jan. 2021.

[18] G. Hakobyan and B. Yang, "High-performance automotive radar: A review of signal processing algorithms and modulation schemes," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 32–44, Sep. 2019.

[19] S. Zollo and B. Ristic, "On polar and versus Cartesian coordinates for target tracking," in *Proc. 5th Int. Symp. Signal Process. Appl. (ISSPA)*, vol. 2, Aug. 1999, pp. 499–502.

[20] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using Doppler radar," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 869–874.

[21] I. Roldan, F. Fioranelli, and A. Yarovoy, "Self-supervised learning for enhancing angular resolution in automotive MIMO radars," *IEEE Trans. Veh. Technol.*, early access, Apr. 21, 2023, doi: 10.1109/TVT.2023.3269199.

[22] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[23] C. X. Lu et al., "MilliEgo: Single-chip mmWave radar aided egomotion estimation via deep sensor fusion," in *Proc. 18th Conf. Embedded Netw. Sensor Syst.*, Nov. 2020, pp. 109–122.

[24] S. Zhu, F. Fioranelli, and A. Yarovoy, "Radar-only instantaneous ego-motion estimation using neural networks," in *Proc. 20th Eur. Radar Conf. (EuRAD)*, 2023.

[25] R. Zhu et al., "AdaFit: Rethinking learning-based normal estimation on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6098–6107.

[26] O. Schumann et al., "RadarScenes: A real-world radar point cloud data set for automotive applications," Mar. 2021. [Online]. Available: https://zenodo.org/record/4559821, doi: 10.5281/zenodo.4559821.

[27] M. Adams, M. D. Adams, and E. Jose, *Robotic Navigation and Mapping With Radar*. Norwood, MA, USA: Artech House, 2012.

[28] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.

[29] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, Apr. 1992.

[30] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, Oct. 1994.

[31] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Modeling*, 2001, pp. 145–152.

[32] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro Universitet, Örebro, Sweden, 2009.

[33] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Oct. 2003, pp. 2743–2748.

[34] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, "Beyond points: Evaluating recent 3D scan-matching algorithms," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 3631–3637.

[35] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.

[36] T. Stoyanov, M. Magnusson, and A. J. Lilienthal, "Point set registration through minimization of the L_2 distance between 3D-NDT models," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 5196–5201.

[37] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1377–1393, Oct. 2012.

[38] M. Barjenbruch, D. Kellner, J. Klappstein, J. Dickmann, and K. Dietmayer, "Joint spatial- and Doppler-based ego-motion estimation for automotive radars," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 839–844.

[39] M. Rapp, M. Barjenbruch, K. Dietmayer, M. Hahn, and J. Dickmann, "A fast probabilistic ego-motion estimation framework for radar," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2015, pp. 1–6.

[40] A. Das and S. L. Waslander, "Scan registration with multi-scale k-means normal distributions transform," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2705–2710.

[41] M. Rapp, M. Barjenbruch, M. Hahn, J. Dickmann, and K. Dietmayer, "Probabilistic ego-motion estimation using multiple automotive radar sensors," *Robot. Auto. Syst.*, vol. 89, pp. 136–146, Mar. 2017.

[42] M. Heller, N. Petrov, and A. Yarovoy, "A novel approach to vehicle pose estimation using automotive radar," 2021, *arXiv:2107.09607*.

[43] P.-C. Kung, C.-C. Wang, and W.-C. Lin, "A normal distribution transform-based radar odometry designed for scanning and automotive radars," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 14417–14423.

[44] K. Haggag, S. Lange, T. Pfeifer, and P. Protzel, "A credible and robust approach to ego-motion estimation using an automotive radar," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6020–6027, Jul. 2022.

[45] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using multiple Doppler radars," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1592–1597.

[46] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[47] F. J. Abdu, Y. Zhang, M. Fu, Y. Li, and Z. Deng, "Application of deep learning on millimeter-wave radar signals: A review," *Sensors*, vol. 21, no. 6, p. 1951, Mar. 2021.

[48] Q. Li et al., "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8465–8474.

[49] Y. Cho, G. Kim, and A. Kim, "Unsupervised geometry-aware deep LiDAR odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2145–2152.

[50] W. Wang et al., "PointLoc: Deep pose regressor for LiDAR point cloud localization," *IEEE Sensors J.*, vol. 22, no. 1, pp. 959–968, Jan. 2022.

[51] R. Aldera, D. D. Martini, M. Gadd, and P. Newman, "Fast radar motion estimation with a learnt focus of attention using weak supervision," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 1190–1196.

[52] D. Barnes, R. Weston, and I. Posner, "Masking by moving: Learning distraction-free radar odometry from pose information," 2019, *arXiv:1909.03752*.

[53] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9484–9490.

[54] S. Lim, J. Jung, B. Lee, J. Choi, and S.-C. Kim, "Radar sensor-based estimation of vehicle orientation for autonomous driving," *IEEE Sensors J.*, vol. 22, no. 22, pp. 21924–21932, Nov. 2022.

[55] R. Weston, M. Gadd, D. De Martini, P. Newman, and I. Posner, "FastMbyM: Leveraging translational invariance of the Fourier transform for efficient and accurate radar odometry," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2186–2192.

[56] O. Schumann et al., "RadarScenes: A real-world radar point cloud data set for automotive applications," in *Proc. IEEE 24th Int. Conf. Inf. Fusion (FUSION)*, Nov. 2021, pp. 1–8.

[57] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[58] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.

[59] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[60] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.

[61] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[62] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[63] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.

[64] E. Camuffo, D. Mari, and S. Milani, "Recent advancements in learning algorithms for point clouds: An updated overview," *Sensors*, vol. 22, no. 4, p. 1357, Feb. 2022.

[65] A. Laribi, M. Hahn, J. Dickmann, and C. Waldschmidt, "A new height-estimation method using FMCW radar Doppler beam sharpening," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, p. 1932.

[66] C. D. Monaco and S. N. Brennan, "RADARODO: Ego-motion estimation from Doppler and spatial data in RADAR images," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 3, pp. 475–484, Sep. 2020.

[67] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[68] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.

[69] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *J. Field Robot.*, vol. 27, no. 5, pp. 609–631, Sep. 2010.

[70] M. Holder, S. Hellwig, and H. Winner, "Real-time pose graph SLAM based on radar," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1145–1151.

[71] Delft High Performance Computing Centre (DHPC). (2022). *Delft-Blue Supercomputer (Phase 1)*. [Online]. Available: https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1

[72] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A. Agha-Mohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5739–5746.

[73] C. Doer and G. F. Trommer, "X-RIO: Radar inertial odometry with multiple radar sensors and yaw aiding," *Gyroscopy Navigat.*, vol. 12, no. 4, pp. 329–339, Dec. 2021.

[74] Y. Z. Ng, B. Choi, R. Tan, and L. Heng, "Continuous-time radar-inertial odometry for automotive radars," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 323–330.

[75] M. Steiner, O. Hammouda, and C. Waldschmidt, "Ego-motion estimation using distributed single-channel radar sensors," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Apr. 2018, pp. 1–4.

Simin Zhu (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering and automation from the Central South University in 2016. Afterward, he worked for 1.5 years as a hardware engineer at Huawei Technology Co. Ltd. In 2019, he started his master's study at Delft University of Technology (TU Delft). During his master's program, he specialized in radar signal processing and machine learning. In November 2021, he completed his master's thesis and graduated from the Microwave Sensing, Signals and Systems (MS3) group at TU Delft. In December 2021, he continued his research in the MS3 group as a Ph.D. candidate.

Alexander Yarovoy (Fellow, IEEE) received the Diploma degree (Hons.) in radiophysics and electronics and the Candidate Physics and Mathematical Sciences and the Doctor Physics and Mathematical Sciences degrees in radiophysics from Kharkov State University, Kharkiv, Ukraine, in 1984, 1987, and 1994, respectively. In 1987, he joined the Department of Radiophysics, Kharkov State University, as a Researcher and became a Full Professor in 1997. From September 1994 to 1996, he was with the Technical University of Ilmenau, Germany, as a Visiting Researcher. Since 1999, he has been with the Delft University of Technology, The Netherlands, where he has been leading as the Chair of the Microwave Sensing, Systems and Signals Group since 2009. He has authored and coauthored more than 500 scientific or technical articles, seven patents, and 14 book chapters. His main research interests are in high-resolution radar, microwave imaging, and applied electromagnetics (in particular, UWB antennas). He was a recipient of the European Microwave Week Radar Award for the paper that best advances the state-of-the-art in radar technology in 2001 (together with L. P. Ligthart and P. van Genderen) and in 2012 (together with T. Savelyev). In 2010 together with D. Caratelli, he got the Best Paper Award of the Applied Computational Electromagnetic Society (ACES). He served as the General TPC Chair for the 2020 European Microwave Week (EuMW'20), as the Chair and the TPC Chair for the 5th European Radar Conference (EuRAD'08), and a Secretary for the 1st European Radar Conference (EuRAD'04). He also served as the Co-Chair and the TPC Chair for the Xth International Conference on GPR (GPR2004). He serves as an Associate Editor for the IEEE TRANSACTION ON RADAR SYSTEMS. From 2011 to 2018, he served as an Associate Editor for the *International Journal of Microwave and Wireless Technologies*. From 2008 to 2017, he served as the Director of the European Microwave Association (EuMA).

Francesco Fioranelli (Senior Member, IEEE) received the Laurea (B.Eng., cum laude) and Laurea Specialistica (M.Eng., cum laude) degrees in telecommunication engineering from Università Politecnica delle Marche, Ancona, Italy, in 2007 and 2010, respectively, and the Ph.D. degree from Durham University, U.K., in 2014.

He was an Assistant Professor with the University of Glasgow from 2016 to 2019 and a Research Associate with University College London from 2014 to 2016. He is currently an Associate Professor with the Delft University of Technology (TU Delft), The Netherlands. He has authored over 140 publications between book chapters, journal and conference papers, edited the books on "micro-doppler radar and its applications" and "radar countermeasures for unmanned aerial vehicles" published by IET-Scitech in 2020. His research interests include the development of radar systems and automatic classification for human signatures analysis in healthcare and security, drones and UAVs detection and classification, automotive radar, wind farm, and sea clutter. He received three best paper awards.