

Advancements in Radar Odometry

Matteo Frosi, Mirko Uscelli, and Matteo Matteucci
Department of Electronics Information and Bioengineering
Politecnico di Milano
Milan, Italy
{name.surname}@polimi.it

Abstract—Radar odometry estimation has emerged as a critical technique in the field of autonomous navigation, providing robust and reliable motion estimation under various environmental conditions. Despite its potential, the complex nature of radar signals and the inherent challenges associated with processing these signals have limited the widespread adoption of this technology. This paper aims to address these challenges and simultaneously present an understanding about the current advancements in radar odometry estimation. First, we propose novel improvements to an existing state-of-the-art method, which are designed to enhance accuracy and reliability in diverse scenarios. Our pipeline consists of filtering, motion compensation, oriented surface points computation, smoothing, one-to-many radar scan registration, and pose refinement. In particular, we enforce local understanding of a scene by including additional information through smoothing (Gaussian kernels) and alignment (ICP), introduced by us in the existing pipeline. Then, we present an in-depth investigation of the contribution of each improvement to the localization accuracy. Lastly, we benchmark our system and state-of-the-art methods on all sequences of well-known datasets for radar understanding, i.e., the Oxford Radar RobotCar, MulRan, and Boreas datasets. In particular, Boreas includes scenarios with challenging weather conditions, such as snow or overcast, and, to our knowledge, it has never been used for evaluation or benchmarking in the literature. The effectiveness of the proposed improvements is proven by an increased translation and rotation accuracy on the majority of scenarios considered.

I. INTRODUCTION

Robust localization is necessary for numerous tasks in the real world, from autonomous driving and transportation, where continuous autonomy is required, to industrial robots, which usually operate in harsh environmental settings within underground mines, construction sites, or even in adverse weather conditions. Indeed, accurately estimating the odometry of a robot in large-scale scenarios remains a considerable challenge for autonomous navigation and all its applications.

Nowadays, state-of-the-art systems exploit exteroceptive sensors like cameras or LiDARs to perform accurate localization [1]. Although the available systems reached a mature state, these sensors are sensitive to environmental artifacts, such as dust, rain, fog, and snow, but also to illumination and viewpoint changes [2], [3], [4]. Due to their longer wavelength, radar sensors prove to be largely resilient to

these conditions [5]. Moreover, they can penetrate certain materials, extending their detection area and going further than the line of sight of LiDARs. Lastly, radars have been receiving increasing attention for various perception tasks, such as place recognition and loop closure detection [6].

A widely used type of radar sensor to achieve localization is the spinning radar, as it is able to provide accurate and dense data, in the form of polar images, representing the power returns at different range and azimuth values. One challenge of spinning radars is that the obtained measurements are inherently hard to interpret with respect to data acquired from other sensors (e.g., LiDARs or cameras), due to the noise characteristics of the sensor [7]. Most localization systems extract hand-crafted features from intensity peaks in polar images, perform robust data association, and compensate for motion distortion [8], [9], [10], [11]. Although state-of-the-art methods integrate local geometry into cost functions and exploit multiple radar sweeps for online incremental odometry, limited attention has been put into possible refinement and smoothing techniques, which further boost localization by using additional spatial information.

This manuscript builds upon the *Conservative Filtering for Efficient and Accurate Radar odometry* work, also known as CFEAR-3 Radar odometry [11]. The authors proposed (i) an efficient method for computing a sparse set of oriented surface points from the intensity peaks of a radar image, by applying a conservative filter and then analyzing local geometries. Then, they included (ii) an incremental scan matching approach that registers the latest scan to multiple non-consecutive scans jointly, by performing weighted minimization of point-to-point correspondence metric. With our work, we include several improvements and a new version of CFEAR that increases its accuracy, as can be seen in Fig. 1, representing the map obtained on the partial sequence 10-14-35 of the Oxford Radar RobotCar dataset [12].

We additionally include an extensive evaluation of the proposed method, along with multiple state-of-the-art systems for radar odometry estimation, on the most well-known datasets for radar data, i.e., the Oxford Radar RobotCar [12], MulRan [13], and the recent Boreas [14] datasets. To our knowledge, we represent the first work on radar odometry in which Boreas has been used for benchmarking.

In summary, we present multiple novel contributions.

- A full radar odometry estimation pipeline consisting of

This paper was supported by “Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile – CNMS)” project funded by the European Union NextGenerationEU program within the PNRR, Mission 4 Component 2 Investment 1.4.



Fig. 1. Map obtained by testing the proposed method on partial sequence 10-14-35 of the Oxford Radar RobotCar dataset [12].

filtering, motion compensation, oriented surface points computation, smoothing, scan-to-multi-keyframe registration, and pose refinement.

- An improved approach for computing the set of oriented surface points, as in (i), which consists of two optional and mutually exclusive methods for kernel smoothing.
- A pose refinement strategy, following (ii), that exploits the previously computed oriented surface points to perform direct point cloud alignment between temporally consecutive radar scans (not keyframes).
- An extensive evaluation of the proposed system and state-of-the-art methods on all sequences of the Oxford Radar RobotCar [12], MulRan [13], and Boreas [14] datasets, considering that the last one has not been yet tested in the literature despite the variety of weather conditions recorded, as we show in the results.
- Finally, we make our C++ implementation available to the community, completely ROS-independent: <https://github.com/AIRLab-POLIMI/CFEAR4>.

II. RELATED WORKS

Due to the advantages of radars over other sensors, many odometry estimation and mapping methods relying on radars have been proposed. A common practice is to use multiple automotive radar sensors [15], which currently offer range and azimuth resolutions similar to laser rangefinders, to estimate numerous relative target velocities (Doppler effect) that can assist in determining the robot's motion [16]. However, this method may not be the most effective for both odometry estimation and mapping due to the relatively low accuracy of velocity measurements compared to other sensors.

Attention then turned to the potential of harnessing the fundamental signal data from radars. From this data, prominent features can be extracted in a manner akin to vision-based systems. However, the process of extracting keypoints from radar data and using them for direct data association has proven to be a complex task due to challenging modeling.

Dissanayake et al. [17] were among the firsts to research the application of radars in localization and mapping. After thresholding the radar's amplitude signal, the obtained ranges and bearings were used to identify landmarks, to be estimated

by the proposed filter-based localization algorithm. Jose and Adams [18] later proposed a feature detector that estimates the probability of target presence while augmenting their formulation to include radar cross-section as a discriminating feature. The works in [19] directly found the transformation between pairs of dense scans, using 3D cross-correlation.

Cen and Newmann [8] then presented a method to extract stable keypoints in radar images, which are then used to perform scan matching and accurately estimate the motion of the robot. The same authors later proposed an update [9] to their radar odometry pipeline, which improved keypoint detection, and proposed a novel graph-matching strategy.

Recently, studies utilizing only radars have focused primarily on improving aspects of odometry [20], [21], [22]. Notable research on effective methods for odometry estimation is presented in the works of Burnett et al. [23] and Park et al. [22]. In the first approach, also referred to as YETI, the researchers measured the significance of motion distortion and demonstrated that Doppler effects need to be eliminated during both localization and mapping. In the second approach, known as PhaRaO, a direct method was suggested as an alternative to using feature-based methods.

Adolfsson et al. proposed CFEAR-1 [24] as an accurate radar odometry estimation method. First, a radar filter keeps only the strongest reflections per-azimuth that exceeds a threshold noise level. Then, the filtered radar data is used to incrementally estimate odometry by registering the current scan with a nearby keyframe. Registration is achieved through the modeling of local surfaces in the data and the minimization of a point-to-line metric. CFEAR-2 [25] improved over the previous method in two ways. Drift is reduced by jointly registering the latest radar scan to a history of keyframes, and the estimated velocity of the robot/sensor is also used to compensate for motion distortion point clouds extracted from radar data. Lastly, CFEAR-3 [11] proposed a combination of weighting, keyframe history, filtering, and a cost function that overcomes the previous challenges with sparsity, bias, and overly conservative filtering, making the method the current state-of-the-art in radar odometry.

Alhashimi et al. [26] presented a method built on the CFEAR framework, using a feature extraction algorithm called Bounded False Alarm Rate (BFAR) to add a constant offset to the usual Constant False Alarm Rate (CFAR) threshold. The resulting radar point clouds are registered to a sliding window of keyframes using an ICP-like optimizer.

Over the years, a few radar-based SLAM systems have been proposed. The work in [27] proposed an ICP-based graph-based SLAM system consisting of preprocessing, odometry estimation via data fusion, scan matching, loop closure, graph construction, and periodic optimization.

Recently, the same authors of CFEAR also presented TBV (Trust But Verify) Radar SLAM [28], a method for radar SLAM that verifies loop closure candidates and uses CFEAR-3 as odometry estimation method. The authors built on the place recognition method Scan Context by Kim et al. [29], [13], which detects loops and relative orientation by matching descriptors associated with multiple point clouds.

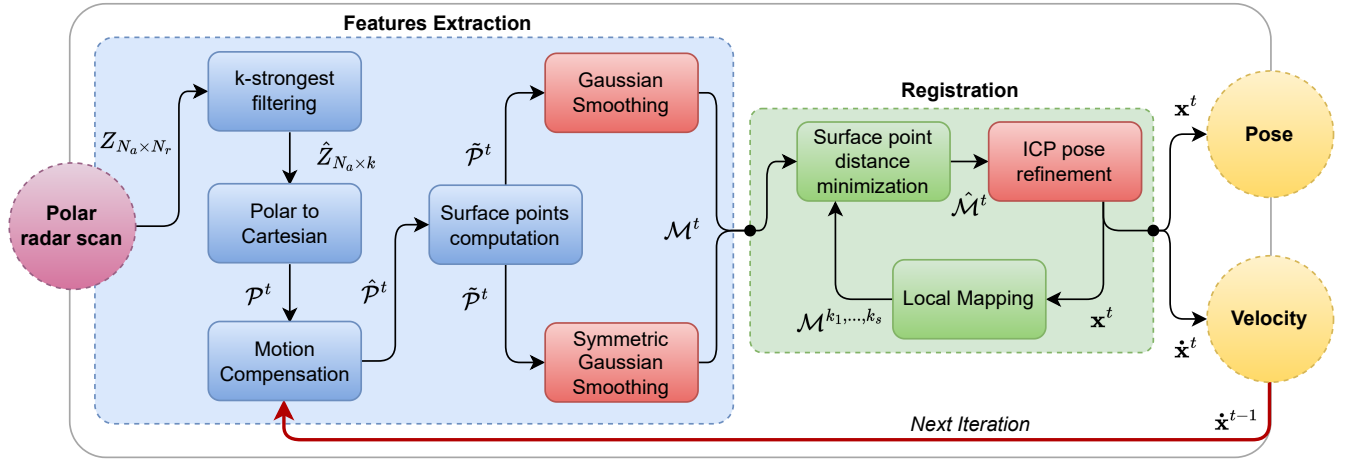


Fig. 2. Our pipeline enhances radar odometry by incorporating multiple steps: filtering polar radar scan, motion compensation, calculating oriented surface points, smoothing, registering multiple radar scans, and refining the pose through ICP. The process leverages smoothing techniques to supplement additional information and aligns successive scans for further refinement after the bulk registration. The red blocks represent the proposed improvements.

III. RADAR ODOMETRY ESTIMATION

Our incremental odometry estimation method, represented in Fig. 2, builds upon CFEAR-3 [11], operating on data gathered by a spinning radar, represented in polar form. Sensor data is first processed to keep only the k -strongest readings per azimuth, to filter out noise points. The filtered point cloud is then compensated for motion distortion using the estimated velocity from the previous iteration. Sparse-oriented surface points are then computed using a grid-based approach that optionally smooths the point distribution around each grid cell. The pose and velocity of the radar sensor are computed by registering the current scan to a history of previous, non-consecutive, scans (i.e., keyframes, forming a local map) and the pose is optionally refined through an ICP-based scan matching solution.

A. Radar pre-filtering

Given a polar radar scan of size $N_a \times N_r$, with N_a azimuth and N_r range bins, radar pre-filtering iterates over all azimuth bins separately. For each azimuth, we choose the k range bins that have the highest power returns, meaning they have intensities in the polar image that surpass the expected noise level, to reduce common data artifacts.

After filtering, each selected range and azimuth pair (d, a) , with $d \in [1, \dots, N_r]$ and $a \in [1, \dots, N_a]$, is transformed to a point in the Cartesian space, for subsequent algorithms:

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} d\gamma \cos(\theta) \\ d\gamma \sin(\theta) \end{bmatrix}, \quad (1)$$

where $\theta = 2\pi a/N_a$ and γ is the sensor range resolution. The output point cloud is the set of all Cartesian points.

B. Motion compensation

Considering the most recent linear and angular estimated velocities $\dot{\mathbf{x}}_{t-1} = [v_x \ v_y \ \omega]$, the filtered point cloud is compensated for motion by projecting each point into the time t of the center of the radar sweep, such that $t_i = t + \delta t$,

where $\delta t = [-\Delta T/2, \Delta T/2]$ and ΔT is the duration of a full sweep. The time offset δt of each point is then computed from the index $a_{\Delta t}$ of the corresponding azimuth bin and the number of bins per sweep: $\delta t = (a_{\Delta t} - N_a/2) \Delta T/2$. Lastly, each point $p_{\delta t}$ can be corrected accordingly to the inverse of the distortion terms, compensating for its motion.

C. Oriented surface points computation

Instead of considering the filtered and motion-compensated point cloud as presented, we conceptualize it as an assembly of oriented surface points $\mathcal{M}_t = \{m_i\}$, to sparsely model geometries and data distribution within the scene. Each oriented surface point consists of mean and normal values, i.e., $m_i = (\mu_i, \mathbf{n}_i)$. The corrected point cloud is downsampled using a regular grid with cell size r .

For each cell in the grid that has at least one point, all l points within a radius r from the cell centroid, represented as matrix $\mathbf{P}_{2 \times l}$, are used to compute the weighted sample mean μ_i and weighted sample covariance:

$$\Sigma_i = (\mathbf{P} - [\mu_i]_{\times l}) \mathbf{W} (\mathbf{P} - [\mu_i]_{\times l})^T, \quad (2)$$

where \mathbf{W} is a normalized diagonal matrix whose elements are proportional to the reflected intensity, as in [20]:

$$\mathbf{W}_{j,j} = z_j - z_{min}, \quad (3)$$

The search radius r determines the final representation's target density of the environment, which is the set of oriented surface points. Considering that this parameter shares a value with the grid cell size, it allows a single point to contribute to the calculation of multiple surface points.

To improve the computation of these points, we incorporated two additional and mutually exclusive enhancements, which rely on smoothing techniques, described as follows.

Gaussian kernel Smoothing. Rather than directly utilizing the mean and covariance of each oriented surface point, a smoothing kernel can be applied to the grid outlined in the preceding sections. This allows both values to be

considered representative of each grid cell. Consequently, the distribution maintained by a single cell not only is associated with the distribution of points within a radius r from its center, but it also represents the weighted distributions of points belonging to the surrounding cells.

Given a typical 3×3 Gaussian kernel, represented as

$$W_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

where each element can be seen as a distribution importance weight w_i , left to right, top to bottom, the new mean and covariance for each surface point become:

$$\begin{aligned} \hat{\mu}_i &= \sum_{i \in \{1, \dots, 9\}} w_i \mu_i, \\ \hat{\Sigma}_i &= \sum_{i \in \{1, \dots, 9\}} w_i (\Sigma_i + \mu_i \mu_i^T) - \hat{\mu}_i \hat{\mu}_i^T. \end{aligned} \quad (4)$$

It is essential to note that each weight w_i is also multiplied by the number of points l that were used to compute the original mean and covariance of each cell before normalizing the sum of weights to 1. This smoothing method has negligible computation and storage requirements, and it allows to reduce sensor noise (both from measurements and inherent).

Symmetric Gaussian kernel Smoothing. A potential drawback of Gaussian kernel smoothing is that in non-uniform environments, such as urban settings, the distribution of points may be inconsistent across neighboring cells. Consequently, the smoothing process could alter the mean and covariance, skewing these values towards cells with higher density. This could lead to issues during the registration process, such as incorrect or misleading correspondences.

The solution we propose, as an alternative to a simple Gaussian kernel, is to use *symmetric smoothing*. Let W_k be the weights matrix of size $k \times k$, already normalized considering the number of points used to compute the various mean and covariance values μ_i and Σ_i . Before we proceed with the smoothing process, we ensure that the weights are *symmetric* with respect to the central element. If there exists at least one pair where one element is positive and the other is zero, then we abstain from performing smoothing.

Specifically, smoothing is only carried out if all weights in the kernel satisfy either of the given conditions:

$$\begin{aligned} (w_{i,j} > 0) \wedge (w_{k-i,k-j} > 0), \\ w_{i,j} = w_{k-i,k-j} = 0. \end{aligned} \quad (5)$$

As a last step, eigen-decomposition on the sample covariance Σ_i allows to estimate the surface normal \mathbf{n}_i , using the eigenvector that corresponds to the smallest eigenvalue. Where mean and normal have been successfully computed, the oriented surface point is then obtained as $m_i = (\mu_i, \mathbf{n}_i)$.

D. Local mapping

Keyframes, which are composed of oriented surface points known as reference scans and the corresponding robot poses, are stored in a sliding window of size s at regular intervals.

This only occurs when the pose, estimated through registration, surpasses a certain threshold distance in either rotation or translation from the previous keyframe. This method prevents unnecessary computation and memory usage when the robot's movement is slow or stationary, such as when waiting at an intersection, forming efficient local submaps.

E. Registration

The pose of the robot is estimated by finding optimization parameters $\mathbf{x}_t = [x \ y \ \theta]^T$ that best align the latest scan \mathcal{M}_t with a window of the most recent keyframes $\mathcal{M}_i, i \in \mathcal{K} = \{1, \dots, K\}$, with K being a fixed window size, according to the function to minimize:

$$f(\mathcal{M}_{\mathcal{K}}, \mathcal{M}_t, \mathbf{x}_t) = \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{C}} w_{i,j} \mathcal{L}_\delta(g(m_{k_j}, m_{t_i}, \mathbf{x}_t)). \quad (6)$$

The term \mathcal{C} is the set of correspondences between nearby oriented surface points associated with two keyframes, found by radius search and surface normal similarity.

As the cost function, we consider the point-to-point (P2P) approach [30], which proves to be the most reliable:

$$g_{P2P}(m_{k_j}, m_{t_i}, \mathbf{x}_t) = \|\mathbf{e}\|^2, \quad (7)$$

with $\mathbf{e} = \mu_{k_j} - (\mathbf{R}_\theta \mu_{t_i} + \mathbf{t}_{x,y})$ being the error term w.r.t. the translation and rotation (matrix form) components of \mathbf{x}_t .

The symbol \mathcal{L}_δ denotes the *Huber Loss*, which is designed to make the cost function less influenced by outliers.

Finally, all residuals are adjusted based on the similarity of surface points, taking into account factors such as point planarity, the number of detections used to calculate the points, and the direction of normals:

$$\begin{aligned} w_{i,j} &= w_{i,j}^{plan} + w_{i,j}^{dim} + w_{i,j}^{dir} \\ &= f_{sim}(p_i, p_j) + f_{sim}(n_i, n_j) + \max(\mathbf{n}_i \cdot \mathbf{n}_j, 0), \end{aligned} \quad (8)$$

where $f_{sim}(a, b) = 2 \min(a, b) / (a + b)$ represents the similarity between two values, and $p = \log(1 + |\lambda_{max} / \lambda_{min}|)$ is the planarity of a surface point, with λ_{max} and λ_{min} being the eigenvalues of the corresponding covariance matrix.

Intuitively, this procedure enforces spatial and temporal consistency between radar scans. Nevertheless, the representation and registration based on oriented surface points may not use some point-based information, which can instead be exploited to make odometry estimates more accurate.

ICP-based pose refinement. In a similar manner to our approach for calculating oriented surface points, where we implemented enhancements (including Gaussian kernel and symmetric Gaussian kernel smoothing) over CFEAR-3, we have also expanded the registration process with an optional pose refinement feature, following the previous step.

First, we compute the mean values of each set of oriented surface points, obtaining their corresponding positions in the Cartesian space. Given all the derived points, we form a 2D point cloud representing the various power returns of each radar scan. The point cloud corresponding to the current radar image is then aligned to the point cloud derived from the previous radar scan, not to be mistaken with the previous keyframe, which may correspond to a few images before

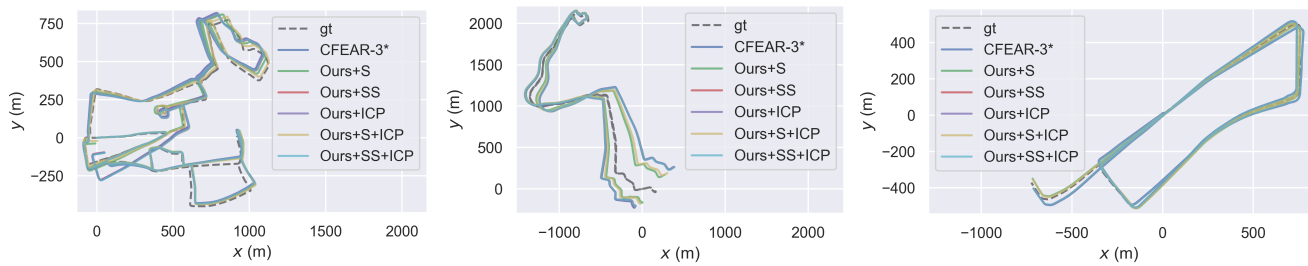


Fig. 3. Odometry trajectory comparison among our pipeline settings with respect to CFEAR-3* and the ground truth. Starting from left to right, the first image is Oxford 18-15-20; the second image is Boreas 01-26-11; and the last image is DCC01.

the current one (depending on the traveled distance). To perform this alignment, we use the robot’s pose estimated at the previous step of the pipeline, meaning the scan-to-multi keyframe registration, as initial guess, due to its overall accuracy. After this ICP-based alignment, we compute the residuals of the various point-to-point correspondences. If their sum is below a threshold (i.e., a mean squared distance between all correspondence valued below 1.0 meters), the pose is updated with the new estimate from the alignment.

This method offers the benefit of further refining consecutive robot steps by ensuring both temporal and spatial consistencies in small windows (due to the consecutive nature of the alignment). In this way, we are able to consider the oriented surface points as a whole point cloud, allowing for a more precise pose estimation.

IV. EVALUATION

A. Setup

We conducted an extensive experimental analysis on the major three datasets within the radar literature:

- Oxford Radar RobotCar [12] (Navtech CTS350-X);
- MulRan [13] (Navtech CIR204-H);
- Boreas [14] (Navtech CIR304-H).

Both Oxford and MulRan had been fully tested on all sequences, while not all trajectories of Boreas had a ground truth to use for comparison. As far as we know, this is the first paper in which Boreas has been used as dataset for evaluating multiple algorithms for radar odometry.

Each radar sensor model has been set according to the Navtech datasheet with respect to its range resolution, while hyper-parameters of the pipeline were chosen according to the CFEAR-3 values reported by Adolfsson et al. [11]. It is also important to note that the Boreas sequences were tested using two different range and resolution values, as the authors state in [14], due to a sudden driver update.

We compare our baseline, which is our own C++ re-implementation of CFEAR (referred to as CFEAR-1*, CFEAR-2*, and CFEAR-3*, respectively, depending on the configuration of parameters), with all possible combinations of the improvements detailed in Section III, and other methods in the literature. These improvements include Gaussian kernel smoothing (S), Symmetric Gaussian kernel smoothing (SS), and ICP-based pose refinement (ICP). We also consider

combinations of these improvements, such as Gaussian kernel smoothing with ICP (S + ICP) and Symmetric Gaussian kernel smoothing with ICP (SS + ICP). Fig. 3 shows a visual comparison of the proposed pipeline with the various improvements, over multiple trajectories.

Moreover, we distinguish between paper and GitHub versions of CFEAR-1* and CFEAR-2*, as the used parameters slightly differ. This distinction is not made for CFEAR-3*, as the set of parameters is unique (refer to [11]).

The comparison is achieved through multiple metrics, i.e., translation error percentage, rotation error expressed in degrees over 100 meters, relative pose error (RPE), in centimeters, and absolute trajectory error (ATE), in meters. The metrics are computed using a well-known and existing benchmarking toolbox¹. We do not include the translation error percentage and rotation error for the Boreas dataset, as the obtained values are considerably high and not as meaningful as the other metrics (i.e., RPE and ATE).

Lastly, in the tables expressing the results achieved in our experiments, we highlight with bold font the most accurate values among the ones we obtained, not including in the comparison the values reported from other papers.

B. Results and discussion

1) *Oxford Radar RobotCar*: Table I shows that the proposed improvements to the pipeline do not lead to better results when considering the sequences that are always used for odometry evaluation in the literature. Nevertheless, when studying the translation and rotation error over all sequences, it becomes clear that the developed methods achieve increased accuracy, having ICP-based solutions performing slightly better than all other alternatives, in translation.

We can also assert that the proposed improvements outperform CFEAR-3* on average across the recorded scenes, with significant improvements in ATE, expressed in Table II, observed in specific local scenes such as 18-14-46 or 10-11-46, among others. Of particular interest is the RPE metric, which always achieves minimum values when derived from the CFEAR-2* method using the parameters as described in the corresponding GitHub repository.

Lastly, it can be noticed that the Gaussian kernel smoothing leads to worse results. This is most likely related to the type of environment represented in the dataset: unbalanced

¹https://github.com/dan11003/radar_kitti_benchmark/tree/master

% translation error / deg/100 [m]	Sequence								Mean	Mean [†]
Method	10-12-32	16-13-09	17-13-26	18-14-14	18-15-20	10-11-46	16-11-53	18-14-46		
Cen2018 [8]	-	-	-	-	-	-	-	-	3.72/0.95	-
MC-RANSAC [23]	-	-	-	-	-	-	-	-	3.31/1.09	-
CC-MEANS [31]	-	-	-	-	-	-	-	-	2.53/0.82	-
RadarSLAM (odom) [10]	2.32/0.7	2.62/0.7	2.27/0.6	2.29/0.7	2.25/0.7	2.16/0.6	2.49/0.7	2.12/0.6	2.32/0.7	-
Kung [20]	-	-	-	-	2.20/0.77	-	-	-	1.96/0.6	-
CFEAR-1 [24]	1.59/0.57	1.84/0.64	1.84/0.64	1.83/0.60	1.71/0.59	1.74/0.57	2.11/0.63	1.69/0.54	1.79/0.60	-
CFEAR-2 [25]	1.35/0.49	1.50/0.51	1.52/0.54	1.52/0.52	1.41/0.50	1.33/0.48	1.61/0.53	1.48/0.50	1.46/0.51	-
CFEAR-3 [11]	1.23/0.36	1.25/0.39	1.25/0.40	1.34/0.41	1.26/0.41	1.26/0.39	1.42/0.39	1.42/0.44	1.31/0.40	-
CFEAR-1* _{paper}	1.83/0.62	2.28/0.75	2.24/0.80	2.14/0.72	2.19/0.76	1.99/0.73	2.62/0.75	2.11/0.73	2.18/0.73	2.20/0.74
CFEAR-1* _{github}	1.81/0.62	1.82/0.64	2.01/0.71	2.14/0.70	1.93/0.70	1.90/0.67	1.97/0.70	2.15/0.74	1.94/0.69	1.93/0.67
CFEAR-2* _{paper}	1.76/0.57	1.88/0.63	1.89/0.68	1.92/0.64	1.64/0.58	1.67/0.60	1.78/0.60	1.80/0.62	1.79/0.62	1.77/0.61
CFEAR-2* _{github}	1.63/0.57	1.69/0.60	1.78/0.64	1.89/0.65	1.77/0.64	1.60/0.57	1.75/0.62	1.88/0.66	1.75/0.62	1.71/0.60
CFEAR-3*	1.29/0.36	1.35/0.39	1.30/0.39	1.43/0.42	1.40/0.43	1.33/0.40	1.49/0.41	1.46/0.42	1.38/0.40	1.37/0.41
CFEAR-3* _S	2.21/0.47	2.45/0.49	2.05/0.56	1.72/0.41	1.70/0.41	1.52/0.40	2.18/0.43	1.75/0.43	1.95/0.45	1.95/0.44
CFEAR-3* _{SS}	1.30/0.37	1.33/0.40	1.31/0.40	1.43/0.42	1.36/0.42	1.34/0.39	1.48/0.41	1.47/0.43	1.39/0.41	1.36/0.40
CFEAR-3* _{ICP}	1.29/0.36	1.35/0.40	1.30/0.39	1.43/0.42	1.38/0.43	1.33/0.40	1.50/0.41	1.46/0.43	1.38/0.41	1.37/0.40
CFEAR-3* _{S+ICP}	1.73/0.42	1.79/0.44	1.66/0.45	1.72/0.41	1.64/0.42	1.54/0.41	2.11/0.43	1.76/0.43	1.73/0.43	1.84/0.43
CFEAR-3* _{SS+ICP}	1.30/0.36	1.33/0.40	1.31/0.40	1.43/0.42	1.36/0.42	1.33/0.39	1.48/0.41	1.46/0.43	1.39/0.40	1.36/0.40

TABLE I

COMPARISON IN % TRANSLATION ERROR AND DEG/100M ON THE OXFORD RADAR ROBOTCAR DATASET [12]. (*) IS OUR RE-IMPLEMENTATION; (S) IS SMOOTHING; (SS) IS SYMMETRIC SMOOTHING; (ICP) IS SCAN-MATCHING; (*paper*) REFERS TO THE PARAMETERS AS DESCRIBED IN [11]; (*github*) REFERS TO THE PARAMETERS AS DESCRIBED IN THE GITHUB REPOSITORY OF CFEAR-3; AND (†) IS THE MEAN OVER ALL SEQUENCES. VALUES REPORTED FROM OTHER MANUSCRIPTS (ABOVE THE DOUBLE LINE) ARE PRESENTED BUT NOT CONSIDERED IN THE COMPARISON.

RPE [cm] / ATE[m]	Sequence								Mean	Mean [†]
Method	10-12-32	16-13-09	17-13-26	18-14-14	18-15-20	10-11-46	16-11-53	18-14-46		
CFEAR-1* _{paper}	5.57/49.38	6.09/45.58	5.44/104.26	5.82/71.41	5.51/63.89	5.83/82.35	6.57/53.11	5.75/60.38	5.82/66.42	6.08/65.86
CFEAR-1* _{github}	4.78/56.86	5.28/60.42	4.40/63.00	5.10/62.44	4.74/72.77	5.08/59.92	5.59/69.83	5.06/79.57	4.88/65.60	5.25/58.72
CFEAR-2* _{paper}	5.34/32.37	5.86/55.15	5.06/70.21	5.63/57.85	5.28/52.25	5.55/44.33	6.04/40.58	5.61/56.71	5.42/51.05	5.77/46.96
CFEAR-2* _{github}	4.66/42.00	5.17/52.16	4.26/56.03	5.05/62.69	4.67/57.47	5.01/36.96	5.47/50.81	4.94/73.35	4.78/53.81	5.15/47.31
CFEAR-3*	5.41/5.29	5.82/20.43	5.10/11.92	5.55/15.26	5.14/25.55	5.64/13.71	5.80/15.44	5.46/23.59	5.37/16.40	5.71/17.76
CFEAR-3* _S	10.24/50.04	10.54/66.35	9.75/89.55	8.60/12.07	8.26/9.57	9.13/20.84	9.31/19.70	8.20/11.30	9.26/34.90	9.40/27.56
CFEAR-3* _{SS}	5.54/6.03	5.94/23.43	5.21/9.95	5.66/17.25	5.23/25.10	5.78/9.58	5.94/15.40	5.53/25.32	5.60/16.51	5.83/17.61
CFEAR-3* _{ICP}	5.96/5.36	6.04/20.49	5.36/11.67	5.89/15.90	5.78/24.62	6.28/12.21	6.27/14.11	5.92/23.53	5.94/16.36	6.13/17.65
CFEAR-3* _{S+ICP}	10.08/17.94	9.91/28.16	9.66/23.88	8.90/10.36	8.70/9.06	9.78/23.48	9.77/19.56	8.55/10.16	9.54/17.95	9.62/21.11
CFEAR-3* _{SS+ICP}	6.11/5.98	6.14/23.73	5.46/10.03	6.01/18.30	5.88/26.12	6.41/9.73	6.37/15.37	5.95/24.83	6.04/16.64	6.23/17.71

TABLE II

RPE [CM] AND ATE [M] METRICS, RESPECTIVELY, ON THE OXFORD RADAR ROBOTCAR DATASET [12]. (*) IS OUR RE-IMPLEMENTATION; (S) IS SMOOTHING; (SS) IS SYMMETRIC SMOOTHING; (ICP) IS SCAN-MATCHING; (*paper*) REFERS TO THE PARAMETERS AS DESCRIBED IN [11]; (*github*) REFERS TO THE PARAMETERS AS DESCRIBED IN THE GITHUB REPOSITORY OF CFEAR-3; AND (†) IS THE MEAN OVER ALL SEQUENCES.

scenes strongly bias the (non-symmetric) smoothing kernel, decreasing the overall odometry estimation accuracy.

2) *MulRan* : The most notable finding of this analysis, represented in Table III and Table IV, is the overall increased accuracy of ICP-based alternatives (e.g., ICP-only or ICP and kernel smoothing), emphasizing the importance of pose refinement in registration. This difference is particularly significant in the Sejong scene (which, to our knowledge, has never been used for evaluation in the literature), which features a large circular road with a diameter of 7 km. Here, the error accumulation is drastically reduced by the improvements we introduced in radar odometry compared.

An interesting thing to notice is that the lowest RPE is always achieved by the CFEAR-1* method, which uses the parameters from the corresponding GitHub repository (whereas for Oxford it was the CFEAR-2* algorithm).

3) *Boreas*: This dataset highlighted an interesting behavior of the existing algorithms (not studied before, because Boreas has never been used to evaluate methods for radar odometry estimation). CFEAR-3* has worse accuracy than CFEAR-1* and CFEAR-2*, as can be seen in Table V. After an in-depth analysis, we discovered that the point-to-point metric used for registration and optimization leads to worse estimates w.r.t. the point-to-line metric used in CFEAR-1* and CFEAR-2* (other parameters do not seem to particularly influence the RPE/ATE). For this reason, we wanted to further stress the benefit of the proposed improvements by applying them to the CFEAR-2* algorithm that uses the parameters from the corresponding GitHub repository.

In a similar way to what was described for *MulRan*, also here the lowest RPE is always achieved by the CFEAR-1* method, with parameters from the GitHub repository.

Lastly, we decided to not include the translation percent-

Method	Sequence												Mean [†]	Mean
	DCC			KAIST			Riverside			Sejong				
	01	02	03	01	02	03	01	02	03	01	02	03		
RadarSLAM (odom) [10]	2.70/0.5	1.90/0.4	1.64/0.4	2.13/0.7	2.07/0.6	1.99/0.5	2.04/0.5	1.51/0.5	1.71/0.5	-	-	-	1.97/0.5	-
CFEAR-1 [24]	2.73/0.73	1.82/0.60	1.77/0.62	2.62/0.97	2.45/0.90	2.85/1.08	2.55/0.90	2.71/0.82	3.56/0.82	-	-	-	2.56/0.83	-
CFEAR-2 [25]	2.44/0.63	1.65/0.54	1.41/0.50	2.12/0.81	1.93/0.74	2.08/0.87	2.30/0.80	2.07/0.66	2.60/0.59	-	-	-	2.07/0.68	-
CFEAR-3 [11]	2.28/0.54	1.49/0.46	1.47/0.48	1.59/0.66	1.62/0.66	1.73/0.78	1.59/0.63	1.39/0.51	1.41/0.40	-	-	-	1.62/0.57	-
CFEAR-1* _{paper}	2.89/0.68	2.18/0.56	1.78/0.50	3.03/0.95	2.02/0.66	2.50/0.79	2.52/0.72	2.56/0.73	4.00/0.79	2.98/0.63	3.55/0.85	3.63/0.83	2.80/0.72	2.61/0.71
CFEAR-1* _{github}	2.55/0.55	1.65/0.41	1.51/0.43	2.16/0.68	1.71/0.52	1.65/0.52	2.43/0.69	2.21/0.65	2.51/0.52	2.38/0.62	2.98/0.70	3.12/0.77	2.24/0.59	2.04/0.55
CFEAR-2* _{paper}	2.41/0.50	1.58/0.39	1.16/0.31	1.75/0.53	1.64/0.50	1.70/0.52	2.18/0.60	1.96/0.54	2.64/0.52	3.11/0.53	2.79/0.65	2.73/0.68	2.14/0.52	1.89/0.49
CFEAR-2* _{github}	2.41/0.49	1.54/0.38	1.21/0.31	1.63/0.51	1.54/0.46	1.62/0.50	2.12/0.58	1.90/0.53	2.53/0.49	3.80/0.53	2.82/0.66	2.77/0.69	2.16/0.51	1.83/0.47
CFEAR-3*	2.12/0.29	1.29/0.22	1.29/0.26	1.13/0.32	1.23/0.37	1.18/0.34	1.21/0.35	1.18/0.31	1.36/0.23	1.10/0.24	1.58/0.32	1.65/0.42	1.36/0.30	1.33/0.30
CFEAR-3* _S	2.37/0.33	1.64/0.26	1.77/0.31	1.34/0.35	1.39/0.40	1.31/0.36	1.57/0.38	1.78/0.37	1.64/0.21	1.48/0.25	1.91/0.34	1.93/0.43	1.68/0.33	1.64/0.33
CFEAR-3* _{SS}	2.13/0.29	1.31/0.22	1.32/0.27	1.14/0.32	1.25/0.37	1.16/0.33	1.24/0.35	1.15/0.31	1.37/0.23	1.10/0.24	1.61/0.33	1.64/0.42	1.37/0.31	1.34/0.30
CFEAR-3* _{ICP}	2.12/0.29	1.29/0.22	1.29/0.26	1.13/0.32	1.23/0.37	1.18/0.34	1.21/0.35	1.18/0.31	1.36/0.23	1.11/0.24	1.59/0.32	1.65/0.42	1.36/0.31	1.33/0.30
CFEAR-3* _{S+ICP}	2.37/0.33	1.64/0.26	1.77/0.31	1.34/0.32	1.39/0.40	1.31/0.36	1.56/0.38	1.78/0.37	1.65/0.21	1.48/0.24	1.90/0.34	1.93/0.43	1.68/0.33	1.65/0.33
CFEAR-3* _{SS+ICP}	2.13/0.29	1.31/0.22	1.31/0.27	1.14/0.32	1.25/0.37	1.16/0.33	1.24/0.35	1.15/0.31	1.37/0.23	1.10/0.24	1.61/0.32	1.64/0.42	1.37/0.31	1.34/0.30

TABLE III

COMPARISON IN % TRANSLATION ERROR AND DEG/100M ON THE MULRAN DATASET [13]. (*) IS OUR RE-IMPLEMENTATION OF CFEAR; (S) IS SMOOTHING; (SS) IS SYMMETRIC SMOOTHING; (ICP) IS SCAN-MATCHING; (paper) REFERS TO THE PARAMETERS AS DESCRIBED IN [11]; (github) AND (g) REFER TO THE PARAMETERS AS IN THE GITHUB REPOSITORY OF CFEAR-3; AND (†) IS THE MEAN OVER ALL SEQUENCES EXCEPT SEJONG.

VALUES REPORTED FROM OTHER MANUSCRIPTS (ABOVE THE DOUBLE LINE) ARE PRESENTED BUT NOT CONSIDERED IN THE COMPARISON.

Method	Sequence												Mean [†]	Mean
	DCC			KAIST			Riverside			Sejong				
	01	02	03	01	02	03	01	02	03	01	02	03		
PhaRaO (full) [22]	-/13.26	-/13.26	-/13.26	-/12.8	-/12.8	-/12.8	-/31.8	-/31.8	-/31.8	-	-	-	-/19.3	-
CFEAR-1 [24]	8.22/21.64	5.00/15.83	5.77/20.77	7.53/37.06	7.22/23.28	7.47/38.79	7.98/30.81	7.36/49.25	6.39/129.23	-	-	-	6.99/40.74	-
CFEAR-2 [25]	7.99/17.32	4.70/7.95	5.46/6.81	7.18/17.26	6.70/16.58	6.96/23.80	7.39/22.06	6.86/47.28	5.83/80.52	-	-	-	6.56/26.62	-
CFEAR-3 [11]	7.83/6.82	4.54/5.13	5.16/4.88	6.37/8.72	6.01/9.89	6.21/13.44	5.90/10.98	5.38/3.26	4.50/17.83	-	-	-	5.77/8.99	-
CFEAR-1* _{paper}	9.04/37.50	5.92/22.93	6.88/20.9	8.43/40.66	7.94/23.83	8.31/32.56	10.13/38.48	8.46/6.77	7.76/148.51	7.76/2741.56	6.6/2230.38	8.03/1863.47	8.09/48.02	7.93/605.63
CFEAR-1* _{github}	8.09/28.07	5.16/14.89	6.12/18.8	7.33/34.99	7.0/22.44	7.34/22.93	8.86/53.34	7.3/89.08	6.5/82.88	6.33/3064.77	6.1/2233.92	6.71/1723.52	7.08/40.83	6.9/615.8
CFEAR-2* _{paper}	8.77/24.40	5.7/12.05	6.61/10.37	8.0/18.1	7.43/23.26	7.85/19.67	9.58/20.97	8.09/45.55	7.16/76.84	8.37/2370.73	6.32/1769.7	6.99/1392.32	7.69/27.91	7.57/482.0
CFEAR-2* _{github}	8.56/24.36	5.5/11.15	6.42/11.22	7.79/16.11	7.18/18.11	7.58/22.04	9.35/22.62	7.75/57.26	6.91/80.77	9.62/2285.38	6.19/2028.48	6.81/1470.76	7.45/29.29	7.47/504.02
CFEAR-3*	8.73/6.56	5.53/5.09	6.58/5.25	7.57/9.21	7.03/11.5	7.32/14.64	8.68/6.86	7.25/9.07	6.24/19.54	6.38/854.37	6.18/737.08	6.77/129.12	7.21/9.75	7.02/150.69
CFEAR-3* _S	11.29/10.46	7.83/7.27	9.38/7.68	9.71/12.29	9.37/16.2	9.6/15.76	11.39/10.83	10.03/18.26	9.6/9.82	7.94/727.12	7.75/602.56	8.27/99.72	9.8/12.06	9.35/128.16
CFEAR-3* _{SS}	8.79/6.75	5.68/5.11	6.65/5.34	7.65/10.41	7.15/11.77	7.39/12.31	8.75/6.93	7.38/12.01	6.32/18.2	6.45/813.21	6.32/724.53	6.82/120.41	7.31/9.87	7.1/145.58
CFEAR-3* _{ICP}	8.74/6.55	5.97/5.12	6.62/5.25	7.58/9.21	7.03/11.5	7.32/14.64	8.69/6.86	7.26/9.06	6.38/19.0	6.48/856.67	6.23/737.03	6.82/129.38	7.29/9.69	7.09/150.86
CFEAR-3* _{S+ICP}	11.30/10.46	8.17/7.29	9.4/7.68	9.71/12.35	9.38/16.2	9.6/15.76	11.43/10.83	10.02/19.6	9.63/8.77	8.03/717.21	7.82/602.32	8.31/101.36	9.85/12.11	9.4/127.49
CFEAR-3* _{SS+ICP}	8.80/6.75	6.06/5.11	6.69/5.34	7.66/10.39	7.15/11.77	7.39/12.31	8.76/6.93	7.42/12.02	6.46/18.02	6.54/811.14	6.28/724.43	6.86/120.45	7.38/9.85	7.17/145.39

TABLE IV

RPE [CM] AND ATE [M] METRICS, RESPECTIVELY, ON THE MULRAN DATASET [13]. (*) IS OUR RE-IMPLEMENTATION OF CFEAR; (S) IS SMOOTHING; (SS) IS SYMMETRIC SMOOTHING; (ICP) IS SCAN-MATCHING; (paper) REFERS TO THE PARAMETERS AS DESCRIBED IN [11]; (github) AND (g) REFER TO THE PARAMETERS AS DESCRIBED IN THE GITHUB REPOSITORY OF CFEAR-3; AND (†) IS THE MEAN OVER ALL SEQUENCES EXCEPT SEJONG. VALUES REPORTED FROM OTHER MANUSCRIPTS (ABOVE THE DOUBLE LINE) ARE PRESENTED BUT NOT CONSIDERED IN THE COMPARISON.

RPE [cm] / ATE [m]	Sequence										Mean	Mean [†]
	Overcast, Snow	Overcast, Snow, Snowing	Overcast, Snow, Snowing	Sun	Sun, Clouds, Construction	Sun, Clouds	Night	Clouds				
Method	01-26-13	12-01-13	01-26-11	04-08-12	04-13-14	06-03-16	09-14-20	11-16-14				
CFEAR-1* _{paper}	12.50/96.74	13.45/166.64	10.61/166.83	14.42/136.21	13.21/91.11	12.94/82.48	13.58/53.51	13.81/48.81	13.06/105.28	13.09/93.55		
CFEAR-1* _{github}	11.98/131.85	12.89/132.99	9.78/187.25	13.88/158.26	12.51/92.76	12.40/48.59	13.06/48.25	13.48/26.84	12.37/103.35	12.53/80.21		
CFEAR-2* _{paper}	12.41/55.24	13.32/53.34	10.29/94.23	14.35/36.94	12.91/47.80	12.86/56.75	13.56/110.99	13.74/73.41	12.93/66.15	12.95/64.91		
CFEAR-2* _{github}	12.10/49.28	12.94/97.41	9.89/113.25	13.98/75.04	12.58/43.17	12.49/44.76	13.11/39.50	13.44/60.44	12.44/65.48	12.57/64.16		
CFEAR-2* _{g,S}	13.82/48.47	14.56/32.35	11.84/32.11	15.36/24.35	14.01/46.04	13.86/57.32	14.54/130.11	14.28/113.09	13.78/60.48	14.01/82.11		
CFEAR-2* _{g,SS}	12.17/49.66	12.97/97.61	10.02/87.05	14.07/67.00	12.68/29.20	12.52/41.12	13.19/34.13	13.47/63.03	12.64/58.72	12.62/60.72		
CFEAR-2* _{g,ICP}	12.16/49.28	12.98/97.41	10.31/113.36	14.05/75.04	12.69/43.45	12.52/44.76	13.15/39.50	13.47/60.44	12.67/65.53	12.64/64.11		
CFEAR-2* _{g,S+ICP}	13.88/48.44	14.61/32.35	12.11/31.41	15.45/24.29	14.11/46.13	13.90/57.32	14.60/130.11	14.34/113.09	14.13/60.27	14.08/82.09		
CFEAR-2* _{g,SS+ICP}	12.22/49.66	13.01/97.62	10.46/89.39	14.13/67.00	12.80/29.21	12.55/41.12	13.23/34.13	13.51/63.03	12.74/58.77	12.70/60.78		
CFEAR-3*	14.79/107.52	15.80/114.38	13.34/91.32	16.82/79.05	15.68/94.98	14.77/135.09	15.60/106.15	13.48/99.98	15.16/103.44	14.93/110.86		
CFEAR-3* _S	16.06/119.07	17.07/120.68	15.07/92.58	17.89/98.07	16.85/82.45	15.94/125.92	16.76/53.89	14.16/94.10	16.35/98.35	16.07/103.62		
CFEAR-3* _{SS}	14.80/102.25	15.87/108.09	13.56/89.72	16.86/78.50	15.75/90.70	14.82/128.71	15.62/107.41	13.51/98.72	15.10/100.64	14.97/109.04		
CFEAR-3* _{ICP}	14.79/107.52	15.84/113.80	13.55/87.18	16.82/79.67	15.71/94.98	14.78/135.09	15.61/106.48	13.51/98.80	15.08/103.05	14.96/110.56		
CFEAR-3* _{S+ICP}	16.07/114.99	17.09/120.68	14.94/76.61	17.88/92.63	16.92/79.70	15.97/128.87	16.77/53.89	14.19/95.37	16.23/95.34	16.09/102.45		
CFEAR-3* _{SS+ICP}	14.81/102.25	15.90/108.09	13.71/97.93	16.88/78.44	15.77/90.70	14.83/128.71	15.65/107.60	13.53/97.68	15.13/101.43	15.00/109.28		

TABLE V

RPE [CM] AND ATE [M] METRICS, RESPECTIVELY, ON THE BOREAS DATASET [14]. (*) IS OUR RE-IMPLEMENTATION OF CFEAR; (S) IS SMOOTHING; (SS) IS SYMMETRIC SMOOTHING; (ICP) IS SCAN-MATCHING; (paper) REFERS TO THE USAGE OF PARAMETERS AS DESCRIBED IN [11]; (github) AND (g) REFER TO THE PARAMETERS AS IN THE GITHUB REPOSITORY OF CFEAR-3; AND (†) IS THE MEAN OVER ALL SEQUENCES HAVING GROUND TRUTH.

age and orientation over distance errors, as they correspond to particularly high values (an average of 66.87 for the translation and 23.35 for the rotation) for all sequences and algorithms employed. These values require a more in-

depth analysis of the used benchmarking tool, to confirm whether they are representative. Nevertheless, the best overall accuracy is achieved by the (SS) configuration of CFEAR-2* and by CFEAR-1* (both GitHub parameters), respectively.

V. CONCLUSION

In this paper, we presented a complete radar-only odometry estimation pipeline, built on CFEAR-3, and consisting of multiple steps: filtering, motion compensation, oriented surface points computation, smoothing, scan-to-multi-keyframe registration, and pose refinement. We developed an improved method for computing the set of oriented surface points, exploiting kernel-based smoothing techniques. We also proposed a pose refinement strategy that performs direct point cloud alignment between temporally consecutive radar scans. We evaluated our pipeline on all sequences of the Oxford Radar RobotCar and MulRan datasets, and on the majority of trajectories of the recent Boreas dataset, showing that we are able to achieve better results in terms of accuracy in the majority of scenarios considered. We also performed an extensive quantitative analysis, describing the impact and performance of our improvements. As a natural progression of this research, we plan to integrate the proposed pipeline into a SLAM framework in the near future and further improve its localization capabilities in complex environments.

REFERENCES

- [1] S. Nahavandi, R. Alizadehsani, D. Nahavandi, S. Mohamed, N. Mohajer, M. Rokonzaman, and I. Hossain, "A comprehensive review on autonomous navigation," *arXiv preprint arXiv:2212.12808*, 2022.
- [2] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. 1–1.
- [3] N. Charron, S. Phillips, and S. L. Waslander, "De-noising of lidar point clouds corrupted by snowfall," in *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE, 2018, pp. 254–261.
- [4] C. Zhang, M. H. Ang, and D. Rus, "Robust lidar localization for autonomous driving in rain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3409–3415.
- [5] K. Burnett, Y. Wu, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot, "Are we ready for radar to replace lidar in all-weather mapping and localization?" *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 328–10 335, 2022.
- [6] M. Usulli, M. Frosi, P. Cudrano, S. Mentasti, and M. Matteucci, "RadarLCD: Learnable radar-based loop closure detection pipeline," 2023.
- [7] A. Venon, Y. Dupuis, P. Vasseur, and P. Merriault, "Millimeter wave fmcw radars for perception, recognition and localization in automotive applications: A survey," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 533–555, 2022.
- [8] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6045–6052.
- [9] —, "Radar-only ego-motion estimation in difficult settings via graph matching," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 298–304.
- [10] Z. Hong, Y. Petillot, A. Wallace, and S. Wang, "Radarslam: A robust simultaneous localization and mapping system for all weather conditions," *The international journal of robotics research*, vol. 41, no. 5, pp. 519–542, 2022.
- [11] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "Lidar-level localization with radar? the cfeare approach to accurate, fast, and robust large-scale radar odometry in diverse environments," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1476–1495, 2023.
- [12] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6433–6438.
- [13] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "Mulran: Multimodal range dataset for urban place recognition," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6246–6253.
- [14] K. Burnett, D. J. Yoon, Y. Wu, A. Z. Li, H. Zhang, S. Lu, J. Qian, W.-K. Tseng, A. Lambert, K. Y. Leung, A. P. Schoellig, and T. D. Barfoot, "Boreas: A multi-season autonomous driving dataset," *The International Journal of Robotics Research*, vol. 42, no. 1-2, pp. 33–42, 2023.
- [15] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [16] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using doppler radar," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 869–874.
- [17] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [18] E. Jose and M. D. Adams, "Relative radar cross section based feature identification with millimeter wave radar for outdoor slam," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 1. IEEE, 2004, pp. 425–430.
- [19] R. Rouveure, M. Monod, and P. Faure, "High resolution mapping of the environment with a ground-based radar imager," in *2009 International Radar Conference "Surveillance for a Safer World" (RADAR 2009)*. IEEE, 2009, pp. 1–6.
- [20] P.-C. Kung, C.-C. Wang, and W.-C. Lin, "A normal distribution transform-based radar odometry designed for scanning and automotive radars," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 417–14 423.
- [21] R. Aldera, D. De Martini, M. Gadd, and P. Newman, "What could go wrong? introspective radar odometry in challenging environments," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2835–2842.
- [22] Y. S. Park, Y.-S. Shin, and A. Kim, "Pharao: Direct radar odometry using phase correlation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2617–2623.
- [23] K. Burnett, A. P. Schoellig, and T. D. Barfoot, "Do we need to compensate for motion distortion and doppler effects in spinning radar navigation?" *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 771–778, 2021.
- [24] D. Adolfsson, M. Magnusson, A. W. Alhashimi, A. J. Lilienthal, and H. Andreasson, "Oriented surface points for efficient and accurate radar odometry," *CoRR*, vol. abs/2109.09994, 2021. [Online]. Available: <https://arxiv.org/abs/2109.09994>
- [25] D. Adolfsson, M. Magnusson, A. Alhashimi, A. J. Lilienthal, and H. Andreasson, "Cfeare radarodometry - conservative filtering for efficient and accurate radar odometry," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5462–5469.
- [26] A. Alhashimi, D. Adolfsson, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Bfar-bounded false alarm rate detector for improved radar odometry estimation," *arXiv preprint arXiv:2109.09669*, 2021.
- [27] M. Holder, S. Hellwig, and H. Winner, "Real-time pose graph slam based on radar," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1145–1151.
- [28] D. Adolfsson, M. Karlsson, V. Kubelka, M. Magnusson, and H. Andreasson, "Tbv radar slam – trust but verify loop candidates," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3613–3620, 2023.
- [29] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.
- [30] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.
- [31] R. Aldera, M. Gadd, D. De Martini, and P. Newman, "What goes around: Leveraging a constant-curvature motion constraint in radar odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7865–7872, 2022.