EGO-MOTION ESTIMATION WITH A LOW POWER MILLIMETER WAVE RADAR ON A UAV

***Pieter Meiresone*[1]*, David Van Hamme*[1]*, Wilfried Philips*[1]*, Tim Verbelen*[2]**

[1]*Department of Telecommunications and information processing, Ghent University, Ghent, Belgium - imec*
[2]*Department of Information Technology, Ghent University, Ghent, Belgium - imec*
**E-mail: Pieter.Meiresone@UGent.be*

Keywords: MILLIMETER-WAVE RADAR, EGO-MOTION ESTIMATION, LOOP CLOSURES

Abstract

Radar sensors have been shown to be capable of performing simultaneous localization and mapping (SLAM) tasks. However, single-chip mmWave radar sensors have received little attention because of their limited resolution. In this paper, we present a novel approach to obtain a robust ego-motion estimation of a UAV using a low power single-chip millimeter wave (mmWave) FMCW radar sensor. By using a novel method to match local radar signal descriptors, we are able to achieve a robust trajectory estimation. We then propose to optimise the trajectory by extracting loop closures from low-dimensional latent space descriptors. We validate our solution in an industrial IoT lab with a drone, but it can be applied more broadly in power contrained platforms.

1 Introduction

Recent years have seen an increase in the development of radar sensors and algorithms. With the limitations of LiDAR in harsh weather conditions and visual sensors at night, interests have grown in sensors such as radar that can be used in all kinds of environments and conditions. Recent developments in low cost mmWave radar sensors have opened a new field of applications thanks to its easy integration and low power consumption.

However, current state of the art in radar SLAM mainly focuses on mmWave radar sensors with longer range, high azimuth resolution and a 360° field of view. A lot of this research takes its ideas from visual SLAM that is carried over to the radar domain. In [1] the authors show that with a modified ICP algorithm two radar point clouds can be mapped. The authors in [2] propose to use SIFT feature descriptors for the radar targets. Similary in [3] a RANSAC algorithm is used in combination with a binary feature representation. We can also apply a direct method such as the Fourier-Mellin Transform to match radar frames [4]. In other work motion estimation is done from the maximizing the quality of the map as a function of the motion [5]. With the release of the Oxford Radar dataset [6], we have seen a big research effort into developing SLAM algorithms with radar as their only input. More advanced feature matching is also being done by novel keypoint descriptors and pairwise compatibility scores [7][8]. Recently neural networks are leveraged to generate good keypoints on which odometry estimation can be done [9][10]. This approach has shown to be very powerful to provide keypoints that are robust to the typical radar clutter.

Low power mmWave radar has been increasingly used the last few years in other research domains. In [11], the authors use mmWave radar to track and identify people using the radar's point cloud. The authors of [12] present a comprehensive dataset consisting of synchronized FMCW radar, depth, IMU and RGB data. More recently, in [13, 14] the entire radar cube is used with a deep neural network to detect vulnerable road users. Less research is done in solving the SLAM problem with mmWave radar. In [15], a mmWave radar is used to execute indoor SLAM by iteratively applying the normal distribution transform scan matching technique. However, an inertial measurement unit (IMU) is required. LatentSLAM [16] uses a biological approach for SLAM that produces good results with radar data and/or when it is fused with vision or depth data. Due to its low-dimensional state descriptors, only an experience map is obtained.

None of these methods listed above have been successfully demonstrated for drones, due to the power constraints, increased number of degrees of freedom and rapid motions inherent to drone flying.

In this paper our main contributions are:

- We present a robust ego-motion estimation technique consisting of a novel keypoint matching and loop closure detection technique.
- We show that our system can be used on a drone in an indoor environment using a low power mmWave radar sensor.

In the next section, we provide the outline of our method. In Section 3 the dataset is discussed on which we tested our method. Experimental results are discussed in Section 4. Finally, in Section 5 we present our conclusions and future work.

2 Methodology

In this section we outline the different steps in our ego-motion estimation algorithm. Our method is divided in the following steps:

1. Radar preprocessing
2. Keypoint extraction

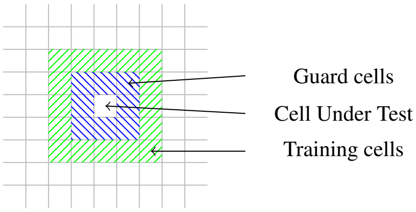


Fig. 1: Principle behind CFAR detections. A sliding window is used over the range-doppler map that calculates the adaptive noise level over the training cells. Guard cells are used to prevent leaks of the target energy in the noise level calculation.

3. Relative motion estimation
4. Loop closure

2.1 Radar preprocessing

In a typical radar processing pipeline the radar chip performs all the necessary tasks on chip and outputs a sparse point set of radar targets. Since this is not suited for in depth analysis, our platform records the raw ADC samples, similarly as in [12]. The subsequent range and doppler FFT are done in software. The angle of arrival (AOA) can be estimated by the FFT over the different antennas. However, this method delivers poor performance in practice. It is necessary to refine the azimuth resolution with a more sophisticated beamforming algorithm like CAPON [17] or MUSIC. We opted to use CAPON because of its good performance that can be achieved with relative low computational requirements. Finally, we obtained the radar cube. From this radar cube we can extract range-azimuth or range-doppler images.

2.2 Keypoint extraction

Due to the low azimuth resolution of the radar sensor, we cannot use typical ORB or SIFT feature detectors to extract keypoints from the range-azimuth plane. Instead, to extract radar keypoints we use the constant false alarm rate (CFAR) in the range-doppler map as shown in Fig. 1. We opted for OS-CFAR for the advantages mentioned in [18]. The downside of using CFAR features with a low azimuth resolution radar is that they can be sparse and non-uniformly distributed throughout the scene and inconsistent over time due to their statistical nature. However, our experiments show that this is not a problem for our algorithm. With the given keypoints, we extract a region of interest (ROI) around it in the range-azimuth plane. This is shown in Fig. 2

2.3 Relative motion estimation

Given the detected keypoints we perform relative motion estimation between two radar frames. Like in [3], we could use a random sample consensus (RANSAC) algorithm to find the correct correspondences between keypoints. However, due to the statistical nature CFAR, the amount of outliers is too high and we found that the RANSAC algorithm is not always able to select a good set of inliers. Instead, we will use a global minimization technique over all the extracted keypoints.

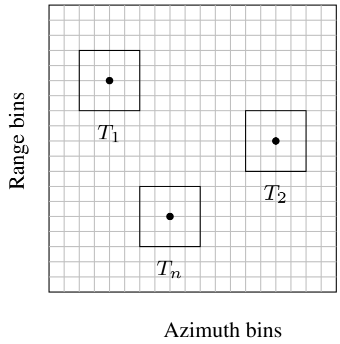


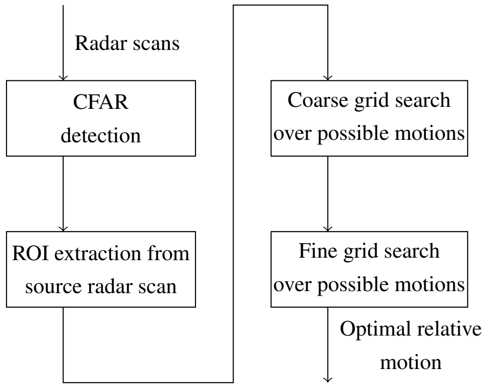
Fig. 2: Feature extraction from the range-azimuth plane. A region of interest (ROI) around the CFAR target is extracted. These descriptors are then later used to calculate a correlation coefficient between two different radar scans.

Our proposed method performs a grid search over possible motions. For each proposed motion, we evaluate the match between the range-azimuth radar scans. For this, each keypoint is transformed with the proposed motion and we extract the ROI in the destination frame. We choose to work as much as possible in polar coordinates because this is the natural sensor domain of the radar. Introducing an extra radar scan conversion from polar to cartesian would only introduce extra artifacts. Furthermore the azimuth bins are not uniformly sized. All the extracted ROI are then correlated between the radar scans. We choose to work with the Pearson correlation coefficient because it produces good results. The motion with the best correlation coefficient is selected. In our algorithm, the grid search is repeated twice. First a search is executed over a coarse grid. With the resulting motion, a new search over a smaller but finer grid is performed. The result is the most optimal motion. The entire procedure is summarized in Fig. 3.

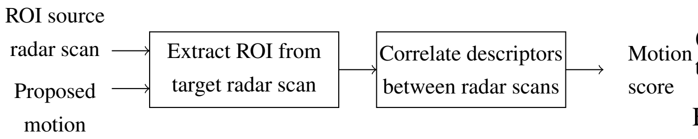
The advantage of our method is that it is very robust to the varying CFAR keypoints. The correlation of all ROI combined has the advantage that we don't require a one-to-one correspondence between keypoints which is more robust. Furthermore, it doesn't require a radar scan conversion which introduces extra artifacts and is less efficient.

2.4 Loop Closure

The resulting trajectory estimate from the previous section produces a significant amount of drift, especially in the rotation estimate. To reduce the drift, we use loop closures extracted from latent space descriptors as described in LatentSLAM [16]. In LatentSLAM a neural network is trained and used that yields latent space descriptors which are then mapped on to an experience map. This method is sensor agnostic and can be applied to any sensor modality. In this paper, we are interested in making a metric map hence we investigated if we can extract loop closures directly from the latent vector descriptors. We re-used the neural network in [16] that is trained on the RGB camera. We can then calculate a similarity score $\delta(\mathbf{t}_1, \mathbf{t}_2)$ between latent

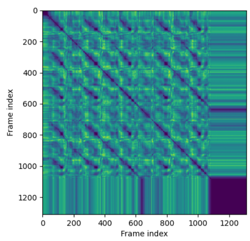


(a) The different steps in the motion estimation pipeline.

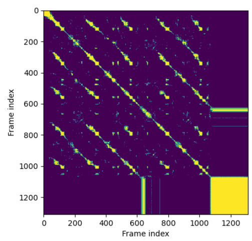


(b) Different steps within the grid search, for each proposed motion a match score is calculated between the descriptors of the radar scans. The motion with the best score is selected.

Fig. 3: Overview of the relative motion estimation algorithm.



(a) Similarity score between each observation.



(b) Similarity score after thresholding.

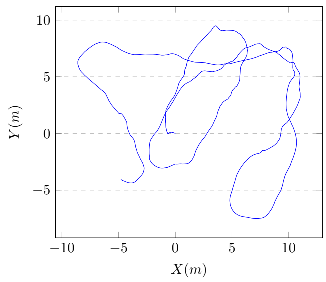
Fig. 4: Within the similarity matrix we can identify loop closures by extracting long sequences of high similarity on the sub-diagonals.

vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ according to:

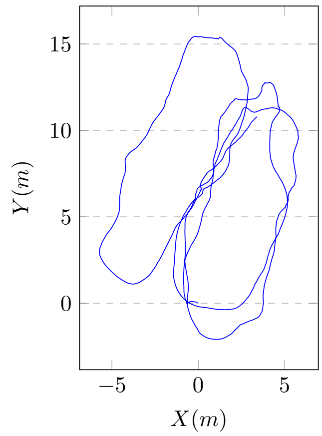
$$\delta(\mathbf{t}_1, \mathbf{t}_2) = 1 - \frac{\mathbf{t}_1 \cdot \mathbf{t}_2}{\|\mathbf{t}_1\| \|\mathbf{t}_2\|}$$

This is 1 minus the cosine similarity of the latent vectors, meaning a good similarity is close to zero. In Fig. 4a, the similarity scores are shown between each latent vector for one of our drone flights. The latent vectors consist of 32 elements. After thresholding (Fig. 4b), loop closures can be extracted by searching for long sequences of high similarity on the sub-diagonals. In practice we looked for sequences of 50 observations or more with a high similarity score.

Together with the pose estimates from Section 2.3 we build a pose graph using the g2o library [19]. For the pose variances we use values in line with the radar resolution. The effect of the performed loop closure for a drone flight is shown in Fig. 5.



(a) Odometry estimate after relative motion estimation.



(b) Odometry estimate after loop closure.

Fig. 5: On the left, the trajectory estimation obtained after motion estimation explained in section 2.3 is displayed. After loop closure, a trajectory estimation is obtained as displayed on the right. The structure of the warehouse is clearly visible.

The optimisation resolves in particular drift in the rotation estimation.

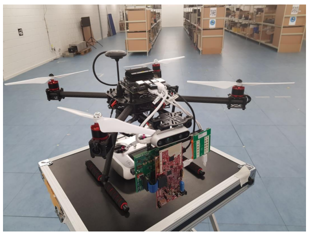
3 Dataset

We use a drone equipped with a Jetson Nano, Intel RealSense depth camera and TI IWR1443 mmWave radar (Fig. 6a). The drone is operated throughout our industrial IoT warehouse (Fig. 6b). The TI radar characteristics are shown in Table 1.

Table 1 TI IWR1443 mmWave radar characteristics.

Table	Table Column Head
Frequency	76 - 81 GHz
Number of receivers	4
Number of transmitters	3
TX power	12 dBm
Range	20m
FOV	180°
Range bins	256
Angular bins	32
Avg. power consumption	1.7-2.1W

The recorded trajectories consist of loops through the different aisles in a random order. Figure 7 visualizes the warehouse and a potential trajectory. The different aisles are difficult to disambiguate and make for an ideal challenging dataset. We evaluate our approach on multiple drone flights through our warehouse, the trajectories are shown in Figure 8. We performed multiple variations in flying through the different aisles. In total, we included three different drone flights in our dataset totaling around 10 minutes of flight time (2500 frames) and around 350m of travelled distance.



(a) Platform.

(b) Warehouse setting.

Fig. 6: Drone platform with Jetson Nano (top), Intel RealSense depth camera (middle), TI IWR1443 mmWave radar (bottom left) and Infineon 24Ghz radar (not used, bottom right).

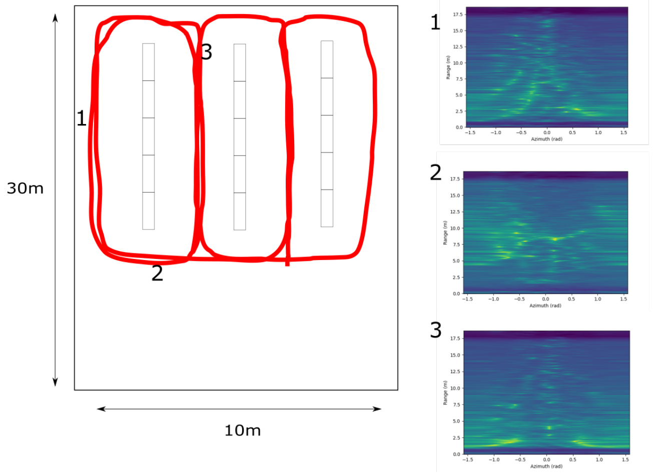


Fig. 7: Layout of the warehouse with a potential trajectory included in red. Executed trajectories are loops through varying aisles in any order. On the right, the radar range-azimuth response is displayed in various positions in the warehouse.

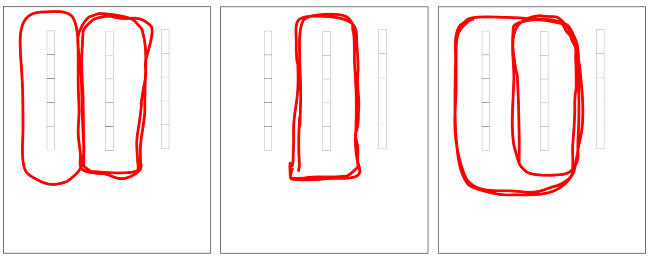


Fig. 8: The performed trajectories within the warehouse.

Due to the high cost of installing a ultra-wideband localisation in our warehouse, we don't have ground truth data available. The only sensor we can compare against is the IMU onboard the drone. In our setup, only the yaw rate of the IMU was reliably enough to make a comparison with our rotation estimate. Other measurements were not reliable due to the vibrations in the drone platform during flight.

Table 2 Evaluation results for three different drone flights. For each sequence, the amount of radar frames and extracted loop closures is indicated. After bundle adjustment, the yaw absolute trajectory estimation error is reduced (ATE_{Yaw}) significantly. The error is calculated against the yaw rate of the IMU unit.

Seq.	No. of frames	Loop closures	ATE_{Yaw} w/o BA (deg)	ATE_{Yaw} w/ BA (deg)
1	800	4	3.17	2.14
2	705	7	3.44	2.43
3	1000	7	2.76	1.98

4 Results

Within our experiments, the relative motion estimation performs first a coarse grid search over $\Delta x, \Delta y \in [-1m, 1m]$ and $\Delta\theta \in [-10°, 10°]$. The step size is respectively $0.1m$ and $1°$. Depending on the flight conditions, a coarser search grid can also work. After an initial estimate is obtained, a fine grid search is done around this estimate over $\Delta x, \Delta y \in [-0.2m, 0.2m]$ and $\Delta\theta \in [-2°, 2°]$. Here the used step size is respectively $0.05m$ and $0.5°$. The pose variances on the translation estimation is taken at $\sigma_x^2 = \sigma_y^2 = 0.1m^2$, in line with the range resolution of $0,078m$. For the rotation similarly determined to be $\sigma_\theta^2 = 0.1rad^2$. Due to the limited computational power and the decision to record raw radar cubes, our drone is able to record everything at 5Hz. The processing of the data happens offline.

We compare our obtained odometry, specifically the estimated yaw rate, with the yaw rate of the IMU data by calculating the absolute and relative trajectory errors [20, 21]. In Table 2 the absolute trajectory yaw error (ATE_{yaw}) is shown for the different sequences. The loop closures have the desired effect of reducing the yaw estimation error. In Figure 9 the estimated trajectories are shown. The middle row shows the accumulated yaw rate (before and after loop closure) of the drone compared with the accumulated yaw rate of the IMU sensor. It clearly shows that after performing loop closure the overal trajectory estimate is significantly improved. In the bottom row of Figure 9, the relative trajectory yaw error (RE_{Yaw}) is shown in function of the distance traveled, it shows that our method limits the RE_{Yaw} between 2.00°and 5.00°. We were unable to compare to other methods from literature since those failed on our dataset which is more difficult due to the reduced resolution of the radar sensor and drone platform.

There are sequences where our method leads to a trajectory estimate where the different aisles are not well enough distinguished, this can be observed in the third sequence in Figure 9. This doesn't come to a surprise to us with the limited resolution of the radar sensor. With a fine enough search grid, these situations can be avoided. More so, very long sequences can result in trajectory estimate that collapses after loop closure.

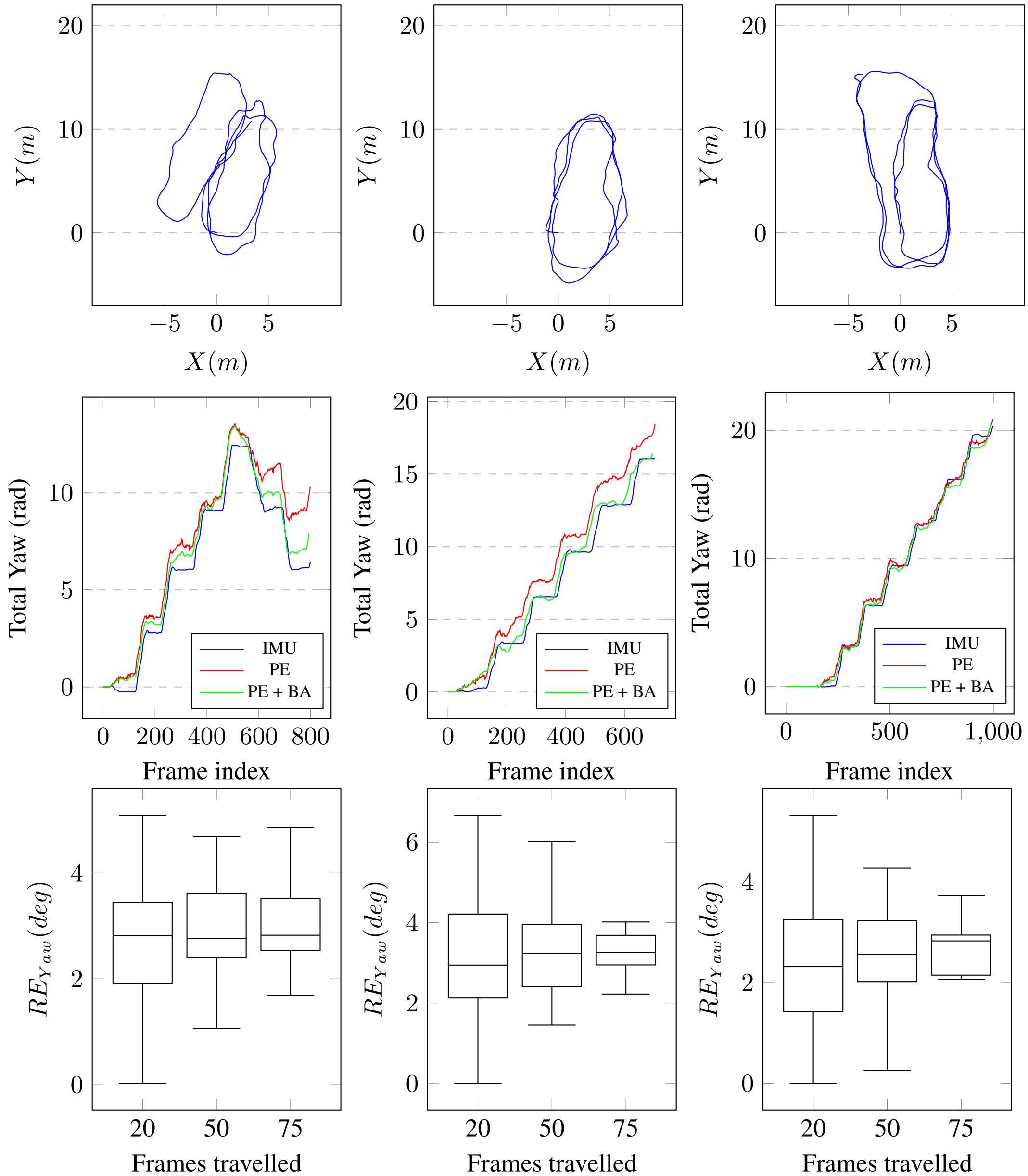


Fig. 9: Evaluation results of the proposed method on the 3 sequences. On the first row the estimated trajectories are shown, on the second row the estimated yaw-rate is compared with the IMU turning rate. The cumulative value is used to better visualise the result. On the last row, the relative trajectory yaw error (RE_{Yaw}) with respect to the IMU yaw rate is shown.

5 Conclusion

In this paper we proposed a new ego-motion estimation algorithm that can be used on low power and low cost mmWave radar. By combining a novel keypoint matching method with a new way to extract loop closures from latent space descriptors, we are able to achieve robust odometry. Our results look promising for further development of SLAM technologies for low power mmWave radar.

In the future we will look into fusing our radar data with other sensors to achieve a more accurate trajectory. Another idea where we will explore is better estimating the used variances for the pose estimations when performing bundle adjustment which could avoid problems such as the trajectory collapse discussed in the previous section.

6 Acknowledgements

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen” program.

References

[1] J. W. Marck, A. Mohamoud, E. vd Houwen *et al.*, “Indoor radar SLAM a radar application for vision and GPS denied environments,” in *2013 European Radar Conference*, pp. 471–474.

[2] J. Callmer, D. Törnqvist, F. Gustafsson *et al.*, “Radar SLAM using visual features,” vol. 2011, no. 1, pp. 1–11, publisher: Springer.

[3] F. Schuster, C. G. Keller, M. Rapp *et al.*, “Landmark based radar SLAM using graph optimization,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2559–2564, ISSN: 2153-0017.

[4] 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)*, pp. 2617–2623, ISSN: 2577-087X.

[5] M. Chandran and P. Newman, “Motion estimation from map quality with millimeter wave radar,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 808–813, ISSN: 2153-0866.

[6] D. Barnes, M. Gadd, P. Murcutt *et al.*, “The oxford radar RobotCar dataset: A radar extension to the oxford RobotCar dataset.” [Online]. Available: http://arxiv.org/abs/1909.01300

[7] 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)*, pp. 6045–6052, ISSN: 2577-087X.

[8] Z. Hong, Y. Petillot, and S. Wang, “RadarSLAM: Radar based large-scale SLAM in all weathers,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5164–5170, ISSN: 2153-0866.

[9] D. Barnes and I. Posner, “Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9484–9490, ISSN: 2577-087X.

[10] K. Burnett, D. J. Yoon, A. P. Schoellig *et al.*, “Radar odometry combining probabilistic estimation and unsupervised feature learning.” [Online]. Available: http://arxiv.org/abs/2105.14152

[11] P. Zhao, C. X. Lu, J. Wang *et al.*, “mID: Tracking and identifying people with millimeter wave radar,” in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 33–40, ISSN: 2325-2944.

[12] T.-Y. Lim, S. A. Markowitz, and M. N. Do, “RaDICaL: A synchronized FMCW radar, depth, IMU and RGB camera data dataset with low-level FMCW radar signals,” vol. 15, no. 4, pp. 941–953, conference Name: IEEE Journal of Selected Topics in Signal Processing.

[13] M. Dimitrievski, I. Shopovska, D. V. Hamme *et al.*, “Weakly supervised deep learning method for vulnerable road user detection in FMCW radar,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8.

[14] W.-Y. Lee, M. Dimitrievski, L. Jovanov *et al.*, “Spatio-temporal consistency for semi-supervised learning using 3d radar cubes,” in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 785–790.

[15] Y. Almalioglu, M. Turan, C. X. Lu *et al.*, “Milli-RIO: Ego-motion estimation with low-cost millimetre-wave radar,” vol. 21, no. 3, pp. 3314–3323, conference Name: IEEE Sensors Journal.

[16] O. Çatal, W. Jansen, T. Verbelen *et al.*, “LatentSLAM: unsupervised multi-sensor representation learning for localization and mapping.” [Online]. Available: http://arxiv.org/abs/2105.03265

[17] J. Capon, “High-resolution frequency-wavenumber spectrum analysis,” vol. 57, no. 8, pp. 1408–1418, conference Name: Proceedings of the IEEE.

[18] H. Rohling, “Ordered statistic CFAR technique - an overview,” in *2011 12th International Radar Symposium (IRS)*, pp. 631–638, ISSN: 2155-5753.

[19] G. Grisetti, R. Kümmerle, H. Strasdat *et al.*, “g2o: A general framework for (hyper) graph optimization,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9–13.

[20] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, ISSN: 1063-6919.

[21] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7244–7251, ISSN: 2153-0866.