Radar Multi Object Tracking using DNN Features

Hassan, Mujtaba; Fioranelli, Francesco; Yarovoy, Alexander; Ravindran, Satish

DOI
10.1109/RADAR54928.2023.10371032

Publication date
2023

Document Version
Final published version

Published in
2023 IEEE International Radar Conference, RADAR 2023

Radar Multi Object Tracking using DNN Features

Mujtaba Hassan
MS3 Group, EEMCS Faculty
Delft University of Technology
Delft, Netherlands
s.m.hassan@tudelft.nl

Francesco Fioranelli
MS3 Group, EEMCS Faculty
Delft University of Technology
Delft, Netherlands
f.fioranelli@tudelft.nl

Alexander Yarovoy
MS3 Group, EEMCS Faculty
Delft University of Technology
Delft, Netherlands
a.yarovoy@tudelft.nl

Satish Ravindran
Advanced Radar Solutions
NXP Semiconductors
San Jose, USA
satish.ravindran@nxp.com

2023 IEEE International Radar Conference (RADAR) | 978-1-6654-8278-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/RADAR54928.2023.10371032

***Abstract*—A single frame radar-based multi-object tracker that aims to improve data association for better tracking performance is proposed. Firstly, a baseline tracker based on track-by-detection paradigm was implemented for automotive radar. Secondly, investigation on the performance of the tracker when tracking individual classes separately versus all classes together was performed. Thirdly, appearance features were extracted from a neural network and added as an additional metric to the cost matrix for improved data association. Extensive experiments on the 2D RadarScenes dataset and a 3D proprietary Lunewave dataset (in partnership with NXP Semiconductors) showed a consistent improvement in the tracking performance using the approach proposed by adding features extracted from a neural network.**

***Index Terms*—Data association, track-by-detection, detector**

I. Introduction

In conventional automotive radar tracking literature [1], a system model is defined for state prediction and update, resorting to multiple complex procedures for data association and track management. The properties of the tracked objects such as size, position, velocity are used for generating the cost matrix to match measurements / detection with tracks (mentioned as "geometric features" in this work). However, other characteristics of the radar point cloud such as radar cross section (rcs), distribution of points, distribution of doppler etc. (mentioned as "appearance features" in this work) are not fully utilized since they are more difficult to model using hand crafted rules. This may result in sub-optimal performance of the tracker when two objects are close together, or when there are many false detections.

Recent state-of-the-art research on multi object tracking for automotive applications is mostly based on deep learning. However, most of these multi object tracking networks originate from camera or lidar domain, and hence do not incorporate radar characteristics into the network. For instance, most of the research is based on track-by-detection paradigm whereby an object is detected at each frame and its association with a corresponding track is found [2]. Neural network based feature extractors are also used, especially in the camera domain to provide appearance features which can be used to compare detections with tracks [3]. These methods produce good results in the case of camera or lidar where the object localization is fairly accurate because of the detailed information available through these sensors for each frame. However, the development of this approach for radar-based multi-object tracking is still underexplored.

A major problem with radar data is that single frame object detection is extremely challenging since radar data is sparse, exhibits miss detections and contains a lot of clutter [4]. Some approaches [5] use multiple frames for object detection and tracking, but this creates latency in the processing which is not acceptable for real time applications. To decrease the chance of missing any object detections, the score threshold can be decreased to include detections with lower confidence. However, this can increase the chances of noise being detected as objects. This makes data association more challenging since the tracker has to make data associations between tracks with not only true object detections but also noise detections. Obtaining discriminative features that helps to distinguish between true objects and noise can help the data association algorithm to separate noise from true objects.

In this paper, we propose to use a single frame, multi object tracker (MOT) based on track-by-detection approach where neural network based appearance features are utilized to improve data association. As shown in Figure 1, the proposed tracker follows the following steps inspired by [6]: (1) A Pointpillar radar object detector [7] to perform object detection from radar point cloud. (2) a Kalman filter to obtain next state prediction from the previous states. (3) a data association module to associate detections with the predicted states using motion and appearance features. (4) a Kalman filter to update the states for the tracks which are matched with detections. (5) a track management module to determine whether to assign unmatched detections to new tracks as well as whether to remove unmatched tracks.

We evaluated our proposed tracker on the opensource 2D RadarScenes dataset, containing only 2 spatial dimensions [8] as well as on a proprietary 3D Lunewave dataset [9], containing 3 spatial dimensions. Our proposed method outperforms the baseline tracker using only motion features for data association by improving MOTA (multi object tracking accuracy) [10] by 2.83% on car class and 4.40% on pedestrian class on the RadarScenes dataset as well as improving MOTA by 1.14 % on the car class on the Lunewave dataset. The main contributions of the paper are as follows: (1) We designed a baseline radar multi object tracker suitable for single frame

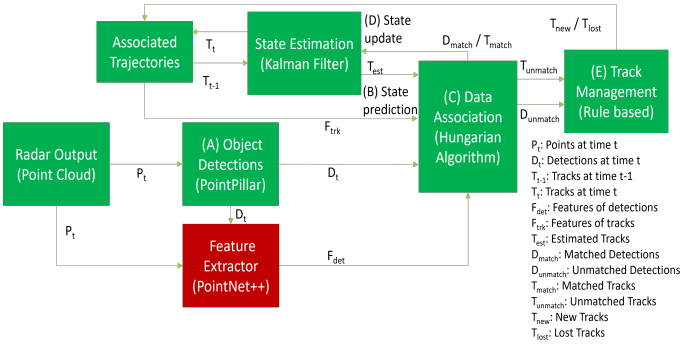


Figure 1. Proposed single frame multi object tracker for automotive radar. The difference from the baseline is that a neural network feature extractor was used to generate appearance features that were used as additional features for data association.

processing. (2) We modified the baseline tracker to use both motion and appearance features in the cost matrix, which helps in improving data association. (3) We compared the performance of tracking all classes of objects together versus tracking each class individually and showed that tracking all class of objects together may provide better result as compared to tracking individual classes for lower resolution radar.

The remainder of this paper is organized as follows: Section II reviews the related work. Section III describes the tracking framework and proposed method. Section IV describes the experiments performed with a discussion on the results. Section V concludes this work.

II. Related Work

Paper [6] showed that applying a simple track-by-detection pipeline with detections generated by a state-of-the-art detector can give very promising tracking performance for lidar. Since a similar technique using automotive radar is developed in this work, research related to this technique for multi object tracking is briefly reviewed in this section.

Radar Object Detection: [11] performs joint object classification / detection using PointNets [12]. [13] uses GNNs (graph neural networks) [14] to perform object detection. [7] uses PointPillars [15] to perform object detection. In our proposed approach, we used [7] for radar object detection because of its simplicity, good performance and wide adaptability.

Discriminating Features: [16] uses micro-Doppler signatures to extract appearance features which are combined with motion features to perform tracking using Deep Metric Learning [17]. [18] performs joint non-line-of-sight object detection and tracking using temporal sequence of doppler velocity and positional measurements. However, both of these methods need temporal frames for feature extraction. [19] uses classification as an additional feature together with motion features for tracking. Inspired by this, we used a similar approach but instead of using hard classification labels, we opted for the softer appearance features coming from a classification network [20] as the additional features.

Data Association: Some methods [21] learn to estimate cost matrix whereas; other methods use deterministic distance functions to estimate cost function for data association including IoU (Intersection-over-Union) distance [6], Mahalanobis distance [22], and Generalized IoU distance [23]. Here, we used the latter approach because of limited training data for training the tracker.

III. Proposed Approach

A. Baseline Tracker

We developed a tracking pipeline inspired by [6] to perform MOT using automotive radar, as shown in Figure 1.

Radar Object Detection: The first step in the pipeline is object detection. This helps to ease the task of data association as the number of detected objects are much smaller than the total number of radar points, reducing the number of possible data associations. PointPillars [7] was used which provides both detection and classification of the desired objects (3 road users: car, pedestrian and bicycle). The input was a radar point cloud which was processed through the network to obtain the object detections.

State Prediction: The users present in the scene are dynamic and hence, can change their location at each frame. In order to track their state parameters, we need a motion model of the system to predict the states of the object in the next frames. The states include object's center (x, y, z), velocity (v_x, v_y, v_z), size (l, w, h) and orientation (θ). Since we perform tracking at a high frame rate (10 Hz) for road users following a relatively smooth motion, we used a constant velocity to model the movement of the objects.

Data Association: The object detection block generated detections at the current frame whereas the Kalman filter state prediction module generated the predicted location of the tracks. Next, detections are matched with the tracks. Typically, a cost matrix is used which computes the cost of assigning each track with each detection. In the baseline approach, difference in motion features between the detected objects and tracks was used as the cost matrix. For the data association module, a single hypothesis strategy was applied where each track was compared with all detections that are within a certain distance to the track. This helps to reduce the complexity of the problem and was solved using the Hungarian Algorithm [24]. The outcome of the algorithm were a set of matched detections (D_{match}) and tracks (T_{match}) as well as unmatched detections ($D_{unmatch}$) and unmatched tracks ($T_{unmatch}$).

State Update: To account for the inaccuracies in state prediction, we used the matched detections to correct the states of the tracks. Bayes rule was used to update the final states of the matched objects based on the measured detections and predicted track states. In this respect, the states were converted into estimated measurements and these were compared with the actual measurements (detections) to provide the required vectors for measurement update equation. Since the object measurements (x,y,z,l,w,h,θ) are a part of the state vector, we used a linear Kalman filter as a measurement equation to convert from states to measurements.

Track Management: A track management module is also needed to manage the addition of new objects that appear in

the scene and removal of objects that are no longer present in the scene. Here, the unmatched detections (D_{match}) can be potential new objects and the unmatched tracks ($T_{unmatch}$) can be potentially deleted. We followed [6] by keeping a minimum number of frames (T_{birth}) where an object is constantly detected before it is assigned to a new track and a minimum number of frames (T_{death}) where the track is not detected before it is deleted.

B. Adding Appearance Features to Improve Data Association

Although motion features can be used to compare detections with tracks, using only them can fail in the case when two objects are very close together or when a false detection or missed detection is matched with a track. In these cases, additional information can help perform correct data association. For this, a neural network can be trained to extract discriminative features for each of the object that is being tracked. These discriminative "appearance" features can then be used as an additional value in the cost matrix to compare between detected objects and tracks. Detections coming from the same object will have similar features but those from other objects or noise will have dissimilar features. So, matching these features between detected objects and tracks can improve data association. Moreover, since there are many noisy object detections in radar data, these appearance features can help to distinguish them from true objects reducing the possibility of incorrect data associations.

In this respect, a neural network was trained for classification and its intermediate features were used as discriminative features to match detections with tracks. The motivation to use a classifier network as compared to other common methods in camera literature (such as Siamese architecture [25]) is that unlike camera domain, datasets in radar domain are limited and so, training a classifier is a relatively simpler task than training a neural network for siamese matching or reidentification tasks.

To compare the appearance features of objects with tracks, a cosine similarity based measure was used. In order to obtain the appearance features for the objects, the following procedure was followed. Firstly, points present within the object bounding box were extracted. Secondly, these points were passed through a classification network which uses an architecture similar to [12]. Lastly, the intermediate features from this network were taken as the appearance features. On the other hand, the appearance features for the tracks were taken from the features of the matched detected objects on the previous frame. The overall cost matrix was obtained using (1), (2), (3):

$$c_{app} = 1 - \frac{F_{det} * F_{trk}}{\|F_{det}\|\|F_{trk}\|}, \tag{1}$$

$$c_{dis} = dist(x_{det}, x_{trk}), \tag{2}$$

$$c_{tot} = c_{dis} + \lambda c_{app}, \tag{3}$$

where F_{det}: appearance features from detection, F_{trk}: appearance features from track, x_{det}: motion features from detection, x_{trk}: motion features from track, λ: positive real number

C. Tracking All Classes Together

Because of the sparsity of radar data, there are very few detections per object, especially for pedestrians. This makes the classification of these objects challenging causing the object detector to output inconsistent class labels for different frames. If we use a tracker that tracks each class separately as is done in [6], we observed a decrease in tracking performance for some cases. So, we also investigated a more conventional tracking approach where all the objects of different classes were tracked together without using the classification labels. This was beneficial for tracking since in many cases object classifications from the object detections were not very robust.

IV. Experimental Results and Discussion

A. Datasets

A major challenge in the radar domain is the lack of open source datasets. Since machine learning algorithms are highly dependent on quality of data, it is important to have a dataset that is large in size and that is representative of the problem to address. Most of the research in automotive radar community is done by industry with proprietary data.

RadarScenes Dataset: We used the opensource 2D RadarScenes dataset [8] for our experiments which provides ground truth label of classes and tracks ids for each moving point. In spite of some limitations such as the lack of height information, and the availability of the annotations only for moving objects, we decided to use this dataset because it is large, provides high quality annotations for each individual points and has a decent azimuth resolution of 0.5°-2° . Also, it can be a good reference dataset to compare our method with the existing works [5] in future.

In our experiments, we use the data coming from sensor 3 which is tilted by 25° from the car's front axis. Based on the point annotations for each point, we generated ground truth bounding boxes for each dynamic object annotated in the scene with a corresponding track id. Then, we trained a modified version of PointPillar network [7] to perform object detection using point cloud coming from single frame. Separately, for each annotated object, we extracted the points present within that object and used these cluster of points to train a modified version of PointNet++ architecture [12] for classification to generate the appearance features. These appearance features were then used to modify the cost matrix part of the data association module to improve tracking.

Lunewave Dataset: Since RadarScenes dataset is a 2D dataset with no height information, we also use the 3D Lunewave dataset in our experiments to assess the performance of the tracker. Here, 3D bounding box annotations with corresponding track ids are available for each object in the scene. We performed a similar procedure of training a 3D object detector based on PointPillar architecture and a classification network based on PointNet++ architecture to obtain object detections and corresponding appearance features which were used to perform data association.

B. Metrics

To evaluate our method, we used the widely used CLEAR MOT metrics [10] that are also used by other autonomous driving benchmarks such as KITTI [26] and NuScenes [27]. The metrics are described as follows:

- MOTA (multi object tracking accuracy): This metric evaluates overall tracking accuracy by combining three error sources: false positives, missed targets and identity switches.
- MOTP (multi object tracking precision): This metric considers the object localization performance via the misalignment between the annotated and the predicted bounding boxes.
- F1: This metric combines precision and recall into a single metric using harmonic mean.

C. Results for RadarScenes Dataset

Quantitative Analysis: Table I, II provide a comparison of the quantitative performance for a tracker tracking individual classes with a tracker tracking all classes together for car and pedestrian class respectively, on the validation set of RadarScenes dataset. Improved performance is obtained if we use a tracker that tracks detections coming from all the classes together instead of separately tracking each of the classes. This is because there are many objects with very few radar points and the pointpillar object detector often does not provide a very accurate class prediction for these cases. So, in these cases, tracking individual classes can cause the tracker to track different classes at different frames.

Table I, II also show the quantitative performance of our tracking strategy of adding appearance features compared with the baseline track-by-detection approach on car and pedestrian class on the validation set of RadarScenes dataset. The major effect of this strategy is on increasing MOTA since this method focuses on the improving data association using appearance features, whereas MOTP remains fairly constant since the improvement in state prediction was not targeted.

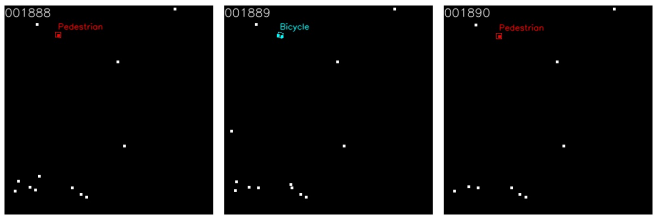
Table I
TRACKING RESULTS ON RADARSCENES DATASET FOR CAR CLASS

Tracking Metrics	Tracker Using Individual Classes	Tracker Using All Classes	Tracker Using Appearance Features
MOTA	65.62	68.40	68.45
MOTP	51.51	51.12	51.65
F1	86.68	86.76	87.42

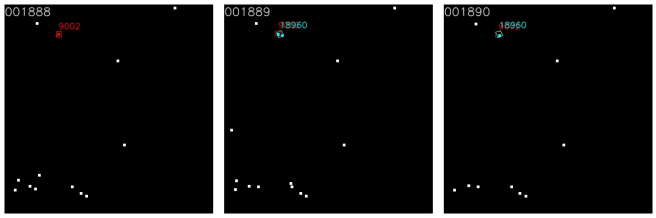
Table II
TRACKING RESULTS ON RADARSCENES DATASET FOR PEDESTRIAN CLASS

Tracking Metrics	Tracker Using Individual Classes	Tracker Using All Classes	Tracker Using Appearance Features
MOTA	39.79	42.36	44.19
MOTP	47.97	47.62	48.52
F1	63.14	63.89	65.98

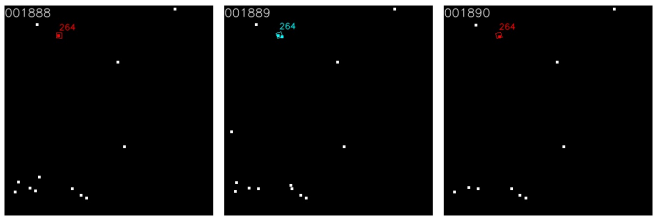
Qualitative Analysis: Figure 2 provides a visualization of the performance between a tracker that tracks individual classes and a tracker that tracks all classes together on a sequence of images. Figure 2a shows the PointPillar detection for 3 consecutive frames. A pedestrian is classified incorrectly as bicycle in frame 2. Figure 2b shows the tracking results for a tracker that tracks individual classes. The tracker starts to track two objects at frame 2. This is because at frame 2, it initiates a new track and starts to track bicycle as a new tracked object separate from the tracked pedestrian object; whereas the track for pedestrian object is also being kept as an unmatched track for that frame. Figure 2c shows the tracking results for a tracker that tracks all classes. For this case, the tracker is able to correctly track a single object since the object detector outputs only a single detection at different frames. This shows that tracking all objects from different classes can be beneficial for the case when the object classifications are not robust.



(a) Detections for 3 consecutive frames



(b) Tracking individual classes

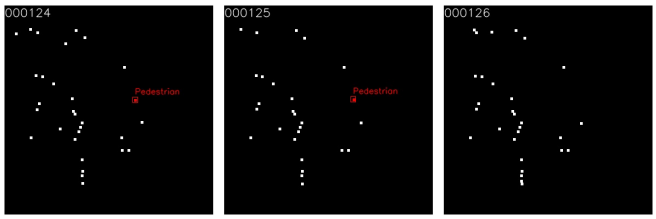


(c) Tracking all classes together

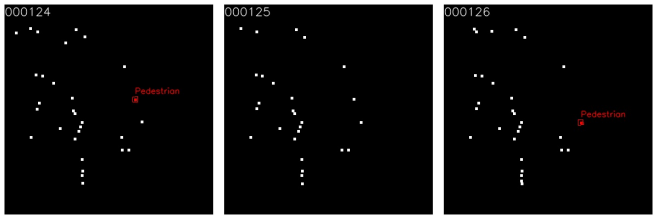
Figure 2. Tracking of a pedestrian, which is classified incorrectly by the PointPillar detector, as seen in 2D radar view (RadarScenes Dataset). (a) Detections for 3 consecutive frames: incorrect classification at frame 2 (bicycle in blue instead of pedestrian in red). (b) Tracking individual classes: incorrect data association after frame 2. (c) Tracking all classes together: correct data association.

Figure 3 provides a visualization of the performance between a tracker that uses only motion features and a tracker that uses additional appearance features for a sequence of images. Figure 3a shows the sequence of images from the ground truth. There is a pedestrian (red label) which is present at frame 1 and 2 but not in frame 3. Figure 3b shows the output of the PointPillar detector for the same sequence of scenes. The detector outputs pedestrian detection in frame 1, misses the detection in frame 2, and makes an incorrect noisy pedestrian class prediction coming from a noise point close to

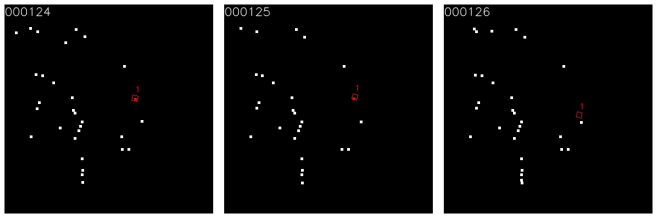
the ground truth in frame 3. Figure 3c shows the output when using a tracker that uses only motion features. The tracker makes incorrect data association at frame 3 by associating the track with the noisy detection since the tracker cannot distinguish between the track and noise detection based only on motion features. Figure 3d shows the output when a tracker that uses additional appearance features for tracking was used. In this case, the tracker makes correct data association at frame 3 by not associating the track with the noise detection. This is because the appearance features of the tracked object and the noise detection are different causing the cost between the tracked object and noise detection to increase.



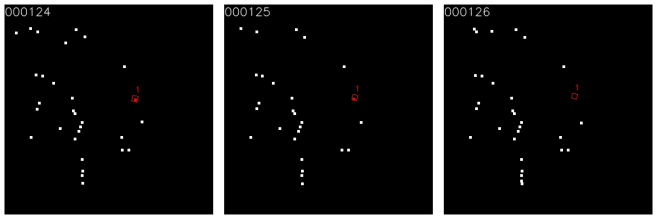
(a) Ground truth labels for 3 consecutive frames



(b) Detections for 3 consecutive frames



(c) Tracks using only motion features



(d) Tracks using additional appearance features

Figure 3. Tracking of a pedestrian (label in red), which is detected incorrectly by the detector, as seen in 2D radar view (RadarScenes Dataset). (b) Detections for 3 consecutive frames: false detection at frame 3. (c) Tracks using only motion features: tracker makes incorrect data association. (d) Tracks using additional appearance features: tracker discards the incorrect detection at frame 3 based on appearance features.

D. Results for Lunewave Dataset

Quantitative Analysis: Table III shows the quantitative performance of the proposed MOT approach on car class of the Lunewave validation dataset compared with the baseline track-by-detection approach. Again, we can observe that adding the classification features helps in improving MOTA performance.

Table III
TRACKING RESULTS ON LUNEWAVE DATASET FOR CAR CLASS

Tracking Metrics	Tracker Using Individual Classes	Tracker Using All Classes	Tracker Using Appearance Features
MOTA	58.98	58.63	60.12
MOTP	60.57	60.08	60.32
F1	78.45	78.45	79.71

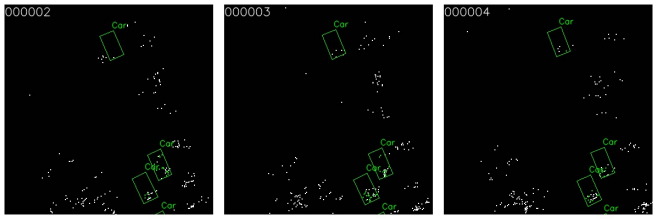
Qualitative Analysis: Figure 4 provides a visualization of the performance between a tracker that uses only motion features and a tracker that uses additional appearance features for a sequence of images. Figure 4a shows the sequence of images from ground truth. Figure 4b shows the output of the PointPillar detector for the same sequence of scenes. The detector outputs incorrect car detections at frame 2 and 3. Figure 4c shows the output when using a tracker that uses only motion features. The tracker makes data association between the two noise detections causing a false track to appear at frame 3. Figure 4d shows the output when a tracker that uses additional appearance features for tracking was used. In this case, the tracker does not create a new track because the appearance features of the two noise detections are different.

E. Limitations

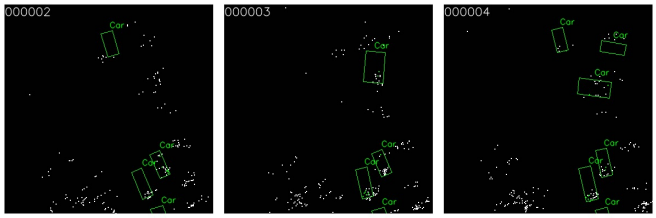
Overall, we see that the proposed approach improves the data association, but does not improve the localization performance. This is because adding neural network features does not improve state estimation. So, for future work, this motivates us to use other approaches such as object detectors with velocity estimation that can provide additional information to state estimators which can result in improving the localization performance. Moreover, radar multi object tracker performance is much lower than what we can expect from a single frame lidar based tracker. This is because the performance of the tracker is highly dependent on the object detector, which is not very robust for a single frame radar data. For future work, we also plan to explore training the feature extractor within the tracking framework instead of reusing the features from classifier to enable better intra-class separation.

V. CONCLUSION

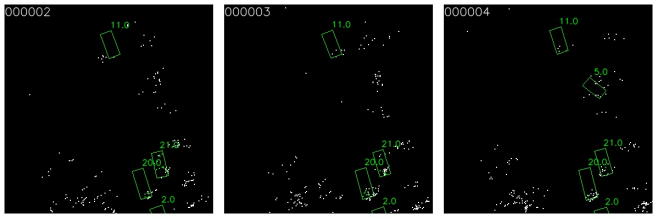
In this work, we proposed a novel real-time multi-object tracker using single frame automotive radar data that aims to solve the problems of data association for improved tracking performance. There are two key concepts in the proposed approach: (1) implementing a baseline tracker for single frame radar-based processing and comparing performance between tracking individual classes with all classes. (2) using appearance features to provide additional information that can



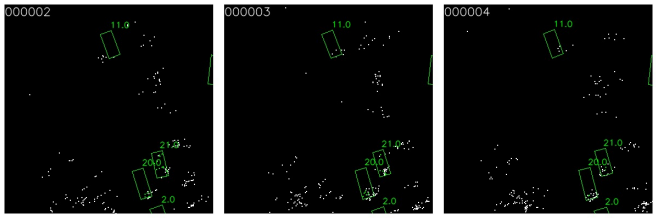
(a) Ground truth labels for 3 consecutive frames



(b) Detections for 3 consecutive frames



(c) Tracks using only motion features



(d) Tracks using additional appearance features

Figure 4. Tracking of car objects (label in green, Lunewave Dataset). (a) Ground truth labels for 3 consecutive frames. (b) Detections for 3 consecutive frames: false detections at frame 2 and 3. (c) Tracks using only motion features: tracker initiates a new track. (d) Tracks using additional appearance features: tracker does not initiate a new track.

improve data association. Based on the experiments performed on the 2D RadarScenes dataset and 3D Lunewave dataset, we observed a consistent improvement in the tracking performance when tracking all classes together along with appearance features.

REFERENCES

[1] N. Floudas, A. Polychronopoulos, and A. Amditis, "A survey of filtering techniques for vehicle tracking by radar equipped automotive platforms," in *7th International Conference on Information Fusion*, vol. 2, 2005.

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings - International Conference on Image Processing, ICIP*, vol. 2016-August, 2016.

[3] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings - International Conference on Image Processing, ICIP*, vol. 2017-September, 2018.

[4] A. Prabhakara, T. Jin, A. Das, G. Bhatt, L. Kumari, E. Soltanaghaei, J. A. Bilmes, S. Kumar, and A. G. Rowe, "High resolution point clouds from mmwave radar," *ArXiv*, vol. abs/2206.09273, 2022.

[5] J. F. Tilly, S. Haag, O. Schumann, F. Weishaupt, B. Duraisamy, J. Dickmann, and M. Fritzsche, "Detection and tracking on automotive radar data with deep learning," in *Proceedings of 2020 23rd International Conference on Information Fusion, FUSION 2020*, 2020.

[6] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *IEEE International Conference on Intelligent Robots and Systems*, 2020.

[7] A. Palffy, E. Pool, S. Baratam, J. F. Kooij, and D. M. Gavrila, "Multi-Class Road User Detection with 3+1D Radar in the View-of-Delft Dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, 2022.

[8] O. Schumann, M. Hahn, N. Scheiner, F. Weishaupt, J. F. Tilly, J. Dickmann, and C. Wohler, "Radarscenes: A real-world radar point cloud data set for automotive applications," in *Proceedings of 2021 IEEE 24th International Conference on Information Fusion*, 2021.

[9] "Automotive radar sensor," https://lunewave.com/automotive-radar-sensor, accessed: May 22, 2023.

[10] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *Eurasip Journal on Image and Video Processing*, vol. 2008, 2008.

[11] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer, "2D Car Detection in Radar Data with PointNets," in *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, 2019.

[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings - IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] P. Svenningsson, F. Fioranelli, and A. Yarovoy, "Radar-PointGNN: Graph Based Object Recognition for Unstructured Radar Point-cloud Data," in *IEEE Radar Conference - Proceedings*, vol. 2021-May, 2021.

[14] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.

[15] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019.

[16] A. Dubey, A. Santra, J. Fuchs, M. Lubke, R. Weigel, and F. Lurz, "A Bayesian Framework for Integrated Deep Metric Learning and Tracking of Vulnerable Road Users Using Automotive Radars," *IEEE Access*, vol. 9, 2021.

[17] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Lecture Notes in Computer Science (subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9370, 2015.

[18] P. Ondrúška and I. Posner, "Deep tracking: Seeing beyond seeing using recurrent neural networks," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016.

[19] S. Haag, B. Duraisamy, W. Koch, and J. Dickmann, "Classification assisted tracking for autonomous driving domain," in *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2018.

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, vol. 2017-December, 2017.

[21] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.

[22] H. K. Chiu, J. Li, A. Rares, and J. Bohg, "Probabilistic 3d multi-modal, multi-object tracking for autonomous driving," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2021.

[23] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," in *Computer Vision – ECCV 2022 Workshops*. Cham: Springer Nature Switzerland, 2023, pp. 680–696.

[24] H. W. Kuhn, "Variants of the hungarian method for assignment problems," *Naval Research Logistics Quarterly*, vol. 3, no. 4, 1956.

[25] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SIAMRPN++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, 2019.

[26] 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*. IEEE, Jun. 2012.

[27] 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 Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.