A Camera–Radar Fusion Method Based on Edge Computing

Yanjin Fu, Daxin Tian, Xunting Duan, Jianshan Zhou, Ping Lang, Chunmian Lin, Xin You
Beijing Advanced Innovation Center for Big Data and Brain Computing
School of Transportation Science and Engineering
Beihang University
Beijing, China

Abstract—**Multi-access edge computing provides a low-latency and high-performance network environment for the Internet of Vehicle services by migrating computing and storage resources to the edge of networks and supports the deployment of more regional-specific vehicle-to-everything services. In this paper, we propose a fusion perception method for roadside cameras and millimeter wave radar as an application of edge computing. First, we used YOLOv3 to process camera data and DBSCAN clustering method to process radar data to obtain the position, speed, and category of detections. Next, we used joint calibration to spatially synchronize camera and radar data and a direct update method to temporally synchronize camera and radar data. Furthermore, we used the Munkres algorithm to associate camera and radar detections, and Kalman filter to track the proposed fusion method's perception results. Finally, we conducted simulation experiments to evaluate the proposed method. The simulation results demonstrate the effectiveness of our algorithm.**

Keywords-edge computing; sensor fusion; camera; radar; roadside perception

I. Introduction

In recent years, the development of autonomous driving technology has increased the attention paid to vehicle perception. While most automated vehicles rely on vehicular sensors for environment perception, detection constraints are inherently present when these vehicles encounter ultra-long distances, complex intersections, or vehicle obstruction conditions. Consequently, how to address the above limitations such as sensing range and improve vehicle perception ability is still an open question in the fields of Internet of Vehicle and connected vehicle and infrastructure systems.

Generally, any environment perception system used for automated or connected vehicles should satisfy at least these two requirements: 1) the system can process a large amount of sensory data in real-time, and 2) the system can send processed results to surrounding connected vehicles. This paper investigates an alternative method for edge perception to address the vehicular perception problem. Based on multi-source information acquired by roadside sensors, real-time data processing is conducted to analyze traffic conditions, and computational results can be shared between roadside facilities and vehicles. Furthermore, the powerful computing and storage capabilities provided by multi-access edge computing (MEC) have significantly reduced the transmission delay between vehicles and infrastructures, and several additional services can be achieved. The overall edge perception architecture with MEC is illustrated in Fig. 1.

Cameras have advantages in terms of target classification and procurement cost, while millimeter wave (mmWave) radar has advantages in terms of speed detection, longitude distance detection, and stability.

In this paper, we propose a fusion perception method using roadside cameras and mmWave radar based on MEC. First, YOLOv3 is used to preprocess camera data, and DBSCN clustering algorithm is used to preprocess radar data. Then, the joint calibration of the camera and radar is used for spatial synchronization, and a direct updating method for time synchronization is proposed. Data fusion is carried out based on previous space–time synchronization. Next, the Munkres algorithm and Kalman filtering are used to associate and track multiple objects, respectively. We verify the performance of our proposed method by simulation and confirm the method outperforms single sensor perception.

The remainder of this paper is organized as follows: Section II introduces the related works on the subject, and Section III provides a detailed description of the fusion algorithm architecture, which is simulated and evaluated in Section IV. Finally, Section V summarizes and concludes the paper.

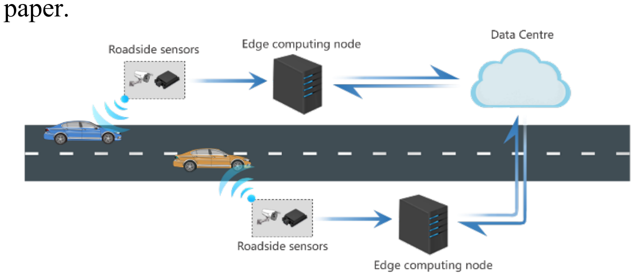

Figure 1. Roadside perception method based on edge computing

II. Related Work

The type of camera perception method chosen has a significant effect on the effectiveness and accuracy of the proposed fusion algorithm. Environment perception aims to detect and classify target objects, such as cars, pedestrians, and cyclists, in traffic scenes. Traditional object detection algorithms such as SIFT, HOG, SURF use sliding windows

978-1-7281-8254-4/20/$31.00 ©2020 IEEE
DOI 10.1109/EDGE50951.2020.00009

of different sizes to delineate regions that need to be detected on an image and extract feature points, and machine learning methods are used to classify these features. In recent years, the improvement of computer hardware has spurred the use of deep learning (DL) methods in this regard. A two-step method such as the recurrent convolution neural network (R-CNN) algorithm series, including R-CNN, Fast R-CNN, and Faster R-CNN, achieves good recognition results. On the contrary, single-step algorithms such as YOLO and single shot multibox detector (SSD) achieve good object detection results.

Presently, few studies have focused on roadside sensors fusion, whereas, there are several studies and applications of vehicular fusion. The following are several studies proposing different fusion methods for camera data and mmWave radar data.

Wu et al. [1] propose a method of fusing data from stereo-camera and mmWave radar. First, they fit the contour of a target vehicle from stereo depth information and find the closest point on the contour. Then, the fused closest point is obtained by fusing observations of radar and the vision closest point. Next, they translate the fitted contour to the fused closest point to obtain the fused contour. Finally, the fused contour is tracked using rigid body constraints to estimate the location, size, pose, and motion of the target vehicle.

Chen et al. [2] present a multi-sensor fusion algorithm with central level architecture. First, they measure the distances of the closest targets using different perception sensors. Then, they use the multi-sensor information fusion algorithm to detect the nearest target in the whole region. Finally, they use Kalman filter to track the target. Their method can improve target detection probability compare to those adopting the sensor level fusion architecture.

Yu et al. [3] propose a fusion method based on Kalman filter and Bayesian estimation. First, they use an SSD method to process images and coordinate transformation matrix to unify the coordinate systems of cameras and mmWave radar. Next, they adopt parallel Kalman filters to track targets from radar and the cameras, respectively. Finally, they use Bayesian estimation to fuse the radar and camera detection results. Their experiment shows that Kalman filters can be used to reduce noise in measurement and fusion algorithms can improve estimation accuracy.

Nobis et al. [4] propose a fusion method based on neural networks (NNs). They designed an NN that can simultaneously receive data from mmWave radar and a camera and output the received data. They use uScenes and a simulated TUM dataset to evaluate the effectiveness of the NN, which shows a good perception performance for their network.

III. Algorithm Architecture

This section mainly introduces the proposed multi-sensor fusion algorithm that consists of data preprocessing, spatial synchronization, time synchronization, and multi-object tracking, as demonstrated in Fig. 2.

A. Data Preprocessing

Valid detections cannot be directly received from cameras and radar, which confirms the necessity of data preprocessing.

1) Camera Data Preprocessing

This paper uses YOLOv3 [5], which is an extremely fast and accurate DL method, to process camera image data. Also, YOLOv3 is a real-time object detection system, which processes images at 30 FPS (frames per second) and has an mAP of 57.9% on COCO test-dev with Pascal Titan X.

2) Radar Data Preprocessing

Radar has some built-in algorithms to process original reflection points to detect a target. However, the reflected detection points are not always accurate, and sometimes, there will be multiple detections on the same vehicle. Hence, a clustering method is required to further process the reflected results. In this paper, DBSCAN clustering method [6] is used to process radar detections. This method does not need to know the final number and shape of the clusters after processing, and it can identify noise points at the same time. However, for samples at the boundary between clusters, their belonging may fluctuate according to which cluster is detected first.

B. Spatial Synchronization

Due to multi-source information obtained by various sensors at different locations, temporal and spatial data synchronization is a prerequisite for sensor fusion. Spatial synchronization for camera and radar data is done through joint calibration. However, the camera and radar must be calibrated separately in advance.

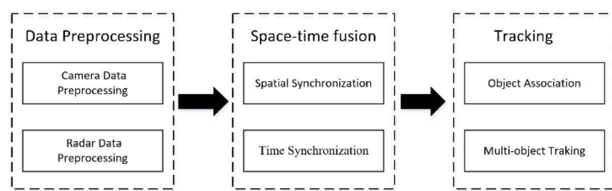

Figure 2. Fusion algorithm flow

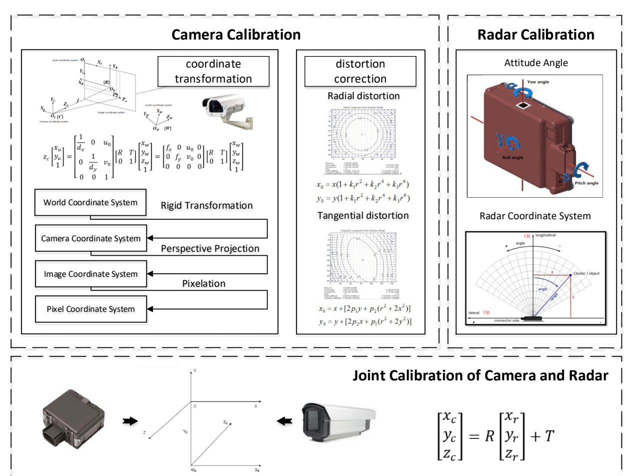

Figure 3. Spatial synchronization

Camera calibration includes coordinate transformation and distortion correction, and the specific steps are shown in Fig. 3. The coordinate transformation mainly involves four coordinate systems, including the world coordinate system, the camera coordinate system, the image coordinate system, and the pixel coordinate system. Distortion correction is caused by camera lens or by installation errors, including radial distortion and tangential distortion. The camera calibration is now finalized using the chessboard calibration method [7]. Radar calibration is mainly achieved by precise installation based on the expected position and angle.

The purpose of the joint calibration of cameras and radar is to transform the measurements from these two sensors into the same coordinate system. In this paper, the measurements from the radar coordinate system are converted into the camera coordinate system. For the three-dimensional to three-dimensional conversion, we placed radar parallel to the ground so that the detection range of radar will be at the same level with the camera, regardless of the attitude angle.

C. *Time Synchronization*

Due to the different sampling frequencies of radar and cameras, time synchronization is required after spatial synchronization to improve the accuracy of target positioning for each frame and the comprehensive utilization of different sensor data. This paper proposes a multi-threaded time synchronization method, including radar thread, camera thread, and fusion thread, as shown in Fig. 4.

The radar thread and camera thread are used to collect data from radar and cameras, respectively, with different sampling frequencies, depending on the sensor. Data obtained in real-time will be sent to a preprocessing module containing algorithms such as YOLOv3 and DBSCAN. Finally, the preprocessed result obtained will be stored in a buffer and updated in real-time using a direct update method. Only one frame of effective radar detection is stored in the buffer. If the new targets have been detected and clustered by cameras and radar, the data in the buffer will be immediately replaced.

The fusion thread consists of three parts, including judgment module, detection assignment, and measurement fusion. The judgment module is controlled by time, and the fusion interval must be longer than the camera and radar sampling intervals. When the system time reaches the fusion time, data from radar and camera buffers will be extracted for subsequent processing. Detection assignment is an object association technique for pairing camera and radar detections to ensure that the two can be matched to each other in the case of multiple targets. The assignment algorithm or clustering algorithm can be used to assign detections, and the Munkres algorithm is selected in this regard. For the measurement fusion, the speed of the final detection is the velocity detected by radar; the type of the final detection is the target type identified by cameras; the average position of camera detections and radar detections is considered to be the final position.

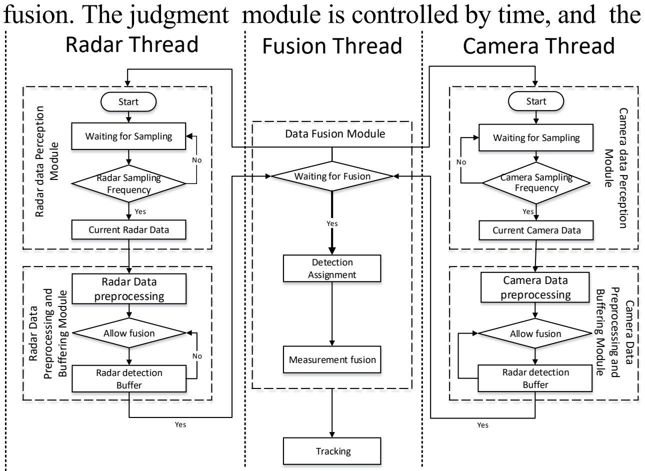

Figure 4. Time synchronization

D. *Multi-object Association and Tracking*

Multi-object tracking with radar and cameras needs to focus on the following points: 1) Object association—radar detections should be associated with camera detections, and current fusion detections must be associated with existing tracks; 2) Tracking—a suitable filter should be used for tracking; 3) Tracks Management—a tracking objects library should be maintained, and the tracking process, including initialization, updating, and deletion, should be defined. The entire multi-target tracking process is illustrated as Fig. 5.

1) Object Association

The Munkres algorithm [8] is used for object association, which aims to solve the assignment problem. The number of assignments and the number of tasks in this algorithm needs to be in one-to-one correspondence. Suppose there are M targets that need to be matched with N targets, the Munkres algorithm will be used to construct a square matrix, and a traditional Hungarian algorithm will be used to obtain the matched result.

To associate radar and camera detections, the results can be divided into three categories: assigned detections, unassigned radar detections, and unassigned camera detections. Measurement fusion is performed directly on assigned detections. Unassigned camera detections are used as valid detections. Unassigned radar detections will be deleted considering the high misdetection rate of mmWave radar.

Similarly, for multi-object tracking, detection results can be divided into three categories: unassigned detections, unassigned tracks, and assigned tracks. In subsequent

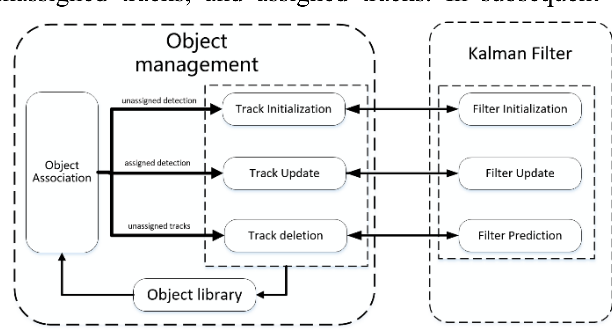

Figure 5. Multi-object Tracking

processing, unassigned detections are used to generate new tracks in the object library; assigned tracks are used to update existing tracks; unassigned tracks are used to delete existing tracks.

2) Tracking

Kalman filtering is widely used for position estimation and object tracking [9]. It comprises two processes and five basic equations. In this method, two-dimensional Kalman filter is used to track vehicles on the road in longitudinal and lateral directions.

3) Object Management

For object management, track management rules are required. A track will be confirmed if it receives at least four detections in the last five updates, and a track will be deleted if no detection was assigned to it in the last five updates.

IV. Simulation Experiment

A. Simulation Environment

The proposed fusion algorithm is evaluated using the Automated Driving Toolbox–MATLAB, which provides the following functionalities: 1) road simulation tools for building specific road scenarios; 2) vehicle simulation tools for simulating vehicles on the road, and setting the accurate trajectory of each vehicle at a specific time; and 3) sensor simulation tools that can simulate the perception results of the mmWave radar and cameras.

A road intersection without channelization and signal lights is set as the simulation scenario for evaluating the proposed fusion perception method. The scenario comprises two four-lane and two-way roads in east-west and north-south directions, respectively, and each lane is 3.6 m wide. The simulation step for this scenario is 0.01 s. A vehicle is set to move with a constant speed of 15 m/s in the longitudinal direction, and the center of the vehicle is located at 1.8 m on the horizontal coordinates.

B. Evaluation Metrics

Three metrics, including mean residual (MR), variance, and residual sum of squares (RSS), are used to evaluate the effectiveness of the proposed algorithm. In the following formulas, $x_{ideal,i}$ represents the actual position of the vehicle, $x_{estimated,i}$ represents the estimated position of the vehicle through fusion perception, and n represents the total times that the vehicle position is estimated during the piroid

Mean residual calculates the average error between the result of a vehicle's perception and the actual running trajectory, and MR formula is given by:

$$\mathrm{MR} = \frac{\sum_{i=1}^{n}\left(x_{ideal,i} - x_{estimated,i}\right)}{n} \tag{1}$$

Variance reflects the stability of the motion obtained by different detection methods. The formula is given by:

$$\mathrm{Variance} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n} \tag{2}$$

RSS represents the sum of squared errors between ideal and estimated values at each sampling point and is used for evaluation. The formula is given by:

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(x_{ideal,i} - x_{estimated,i}\right)^2 \tag{3}$$

C. Sensor Parameter Settings

An mmWave radar is installed in the center of the south-bound lane of the intersection at 0.2 m from the ground. The installation attitude angle is parallel to the ground, which aims to avoid the angle difference between the radar and the detection plane. A camera is installed in the same position at 3 m for a better perception view.

The mmWave radar is set with reference to the parameters of continental radar ARS 404-21 Premium. The long-range detection distance of the radar is 170 m, and the close-range detection distance is 70 m. The long-distance resolution is 0.75 m, and short-distance resolution is 0.4 m. The long-distance detection azimuth is (−9°, +9°), and the measurement interval is 50 ms.

For simulation camera parameters, the default detection angle is set to (−45°, 45°), and the maximum forward detection length is 150 m. The camera target recognition accuracy is dependent on the image processing algorithm used, and the recognition effect is mainly determined by the accuracy of the bounding box and the number of false detections. Hence, in this simulation, the accuracy of the bounding box is set to 20 and the number of false detections is set to 0.5.

D. Experiment Results

Following the simulation, object tracking is achieved and the trajectories of a real-time vehicle and the perceptions of the camera, radar, and proposed fusion algorithm are shown in Fig. 6, which will be analyzed from longitudinal and lateral directions, and the evaluation metrics are shown in Table 1.

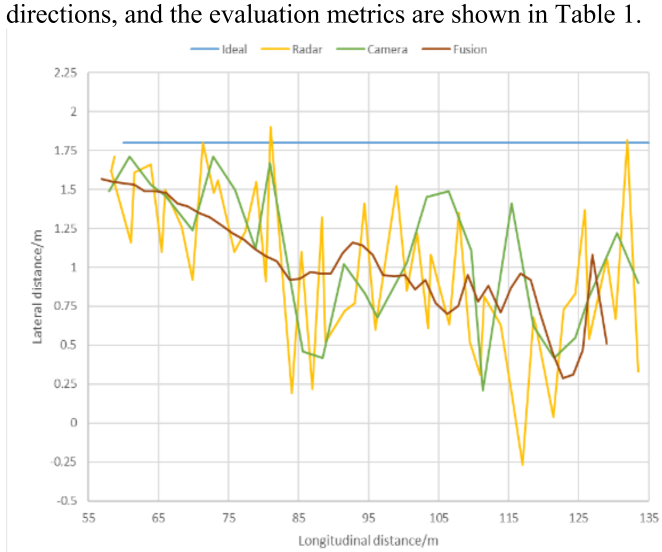

Figure 6. Trajectories from the ideal, radar, camera, and fusion method

TABLE I. EVALUATION METRICS

Longitudinal Direction			
	Camera	Radar	Fusion
MR	1.19	1.94	2.12
VAR	0.85	0.29	0.2
RSS	115.67	206.94	226.68
Lateral Direction			
	Camera	Radar	Fusion
MR	0.73	0.82	0.78
VAR	0.2	0.27	0.11
RSS	37.23	48	34.22

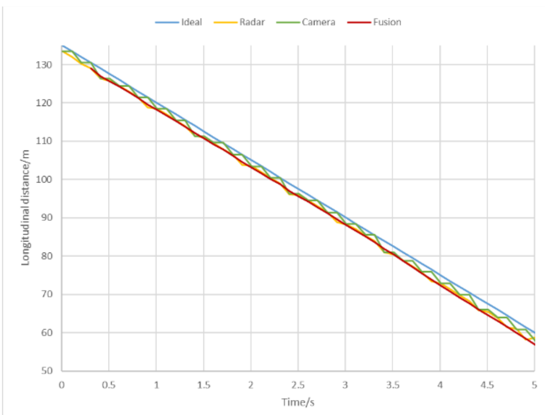

Figure 7. The tracking results in the longitudinal direction

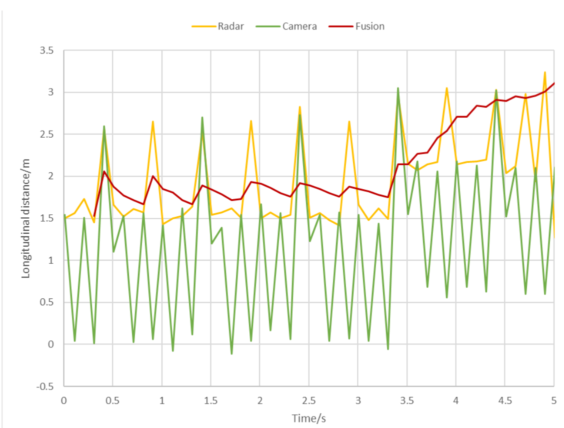

Figure 8. Residual in the longitudinal direction

1) Longitudinal Direction Analysis

Fig. 7 shows the perception results of the camera, radar system, and proposed fusion algorithm for the vehicle in the longitudinal direction and that object tracking is achieved. Since there is no evident difference in the perception results of the camera, radar, and proposed fusion algorithm, we plot a residual perception figure, as shown in Fig. 8.

According to the variance of longitudinal distance perceived by the camera, radar, and proposed fusion algorithm, we find that the detections obtained by the radar are not stable, as well as the detections obtained by the camera. After the application of our proposed algorithm, the stability of perception in the longitudinal direction is greatly improved.

The RSS value increased to 226.68 after the application of the proposed fusion algorithm, and the reason for this phenomenon, based on the residual in the longitudinal direction, is that the fusion algorithm perceptions may have moved from the center of the vehicle to the front when the vehicle approached the roadside sensing unit.

2) Lateral Direction Analysis

Fig. 9 shows the camera, radar, and the fusion algorithm perception results for the vehicle in the lateral direction, and Fig. 10 shows the residual perception of the camera, radar, and fusion algorithm, from which we can locate the vehicle approaching the roadside sensing unit. According to the residual in the lateral direction, the fusion result is getting closer to the true value when the vehicle approaching, and the noise generated by the camera and radar is also largely reduced.

In this situation, the vehicle would be considered a rectangle rather than a point. From this perspective, it can be found that all the lateral positioning points of the vehicle are within the width of the vehicle, which means the proposed fusion algorithm has achieved a good perception result in the horizontal direction.

Figure 9. The tracking results in the lateral direction

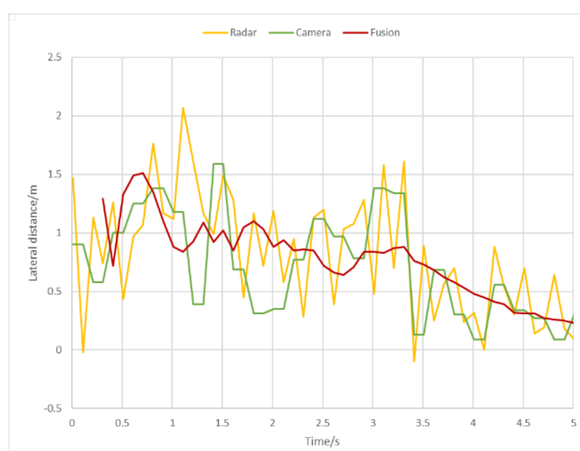

Figure 10. Residual in the lateral direction

V. Conclusion

In this work, we proposed a fusion perception algorithm for roadside cameras and radar as an application of edge computing and evaluated the fusion algorithm through simulations. The simulation results show that the algorithm can effectively improve the stability of a camera and radar detections in both longitudinal and lateral directions. In future works, we will deploy the fusion algorithm on roadside MEC devices and make further improvements based on test results.

Acknowledgment

This research was supported in part by the National Natural Science Foundation of China under Grant No. 61672082 and 61822101, Beijing Municipal Natural Science Foundation No. L191001.

References

[1] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath, “Collision sensing by stereo vision and radar sensor fusion,” Proc. IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 4, Dec. 2009, pp. 606–614, doi: 10.1109/TITS.2009.2032769.

[2] S. Chen, L. Huang, J. Bai, H. Jiang, and L. Chang, “Multi-sensor information fusion algorithm with central level architecture for intelligent vehicle environmental perception system,” SAE Technical Paper, no. 2016-01-1894, September 2016, doi:10.4271/2016-01-1894.

[3] Z. Yu, J. Bai, S. Chen, L. Huang, and X. Bi, “Camera-radar data fusion for target detection via Kalman filter and Bayesian estimation,” SAE Technical Paper, no. 2018-01-1608, August 2018, doi:10.4271/2018-01-1608.

[4] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, “A deep learning-based radar and camera sensor fusion architecture for object detection,” 2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 2019, pp. 1–7, doi: 10.1109/SDF.2019.8916629.

[5] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv preprint arXiv:1804.02767, April 2018.

[6] M. Ester, H.P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial datasets with noise,” Proc. of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, vol. 96, no. 34, Aug. 1996, pp. 226–231.

[7] Z. Zhang, “A flexible new technique for camera calibration,” Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, Nov. 2000, pp. 1330–1334, doi: 10.1109/34.888718.

[8] J. Munkres, “Algorithms for the assignment and transportation problems,” J. Soc. Ind. Appl. Math., vol. 5, no. 1, pp. 32–38, March 1957.

[9] D. Reid, “An algorithm for tracking multiple targets,” Proc. IEEE Transactions on Automatic Control, vol. 24, no. 6, Dec. 1979, pp. 843–854, doi: 10.1109/TAC.1979.1102177.