2020 IEEE International Conference for Innovation in Technology (INOCON)
Bengaluru, India. Nov 6-8, 2020

Multiple Object Tracking using Radar and Vision Sensor Fusion for Autonomous Vehicle

Kishore Reddy Kurapati, Suma M, Ameet Chavan
Department of Electronics and Communications Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India – 501 301
kishorerddy7@gmail.com, yadavsuma043@gmail.com, ameetchavan@sreenidhi.edu.in

***Abstract*—Autonomous Vehicle (AV) employs multiple sensors to sense the surroundings and take decisions accordingly. The aim of the research presented in this paper is to design a sensor fusion and tracking algorithm that tracks the objects based on data fusion of RADAR and Camera. The developed algorithm helps in not only accurately identifying the objects but also determining object's position. The algorithm is tested on a testbed vehicle by fusing RADAR and Vision sensor data. The clutters in RADAR data are detected and removed thus increasing reliability. To avoid the problem due to false data from the sensors, the sensor data is fed to an Interactive Multiple Model (IMM)filter comprising of three Kalman filter. Fusion analysis data shows that the designed IMM filter gives better results as compared to a single Kalman filter.**

***Index Terms*—RADAR, Vision Sensor, Sensor Fusion, IMM Filter**

I. Introduction

Around the world extensive research is being carried on to fuse technologies to make Autonomous Vehicle (AV) a reality. A full AV is meant to be driver-less, must possess the capability to sense, perceive, decide and act on its own. Sensors give AV the ability to sense the surroundings. Sensor fusion and Tracking algorithms give AV the ability to perceive, decide and ultimately act. Variety of sensors is used in AV to sense wide range of parameters. Radio Detection and Ranging (RADAR), Light Detection and Ranging (LiDAR), Ultrasonic sensor and Vision sensor provide data required for object detection and classification. Global Positioning System (GPS), Inertial Measurement Unit (IMU) and Odometer provide data required for navigation [1]. To enhance the capability of the AV to perceive, fusion of data from multiple sensors is done. Sensor fusion algorithms combine data from sensors that are complement to each other to achieve enhanced performance by attaining more reliability and accuracy.RADAR provides accurate position information of objects in the AV surroundings. Vision Sensor detects and classifies objects in the AV surroundings. Fusion of RADAR and Vision sensor data gives a better understanding of the AV surroundings [2],[3],[4].GPS provides ground position information of the AV. IMU sensor provides 3-dimensional acceleration and rotational rate components of the AV. Fusion of GPS data and IMU data provides better understanding about the localization of the AV [5],[6].

Tracking algorithms involve assigning new detections to existing tracks. In Radar data and Vision data fusion, assigning detections obtained from sensor fusion algorithm to physical objects is carried out using Hungarian algorithm. At times sensors have the tendency to produce false data due to various reasons. Mathematical models like Kalman filtering help to reduce the effect of false data on the functioning of AV[7],[8]. Kalman filtering model predicts next possible state of the system based on the previous states which is used when the state estimation due to sensor data is not available[9].

II. Design Methodology

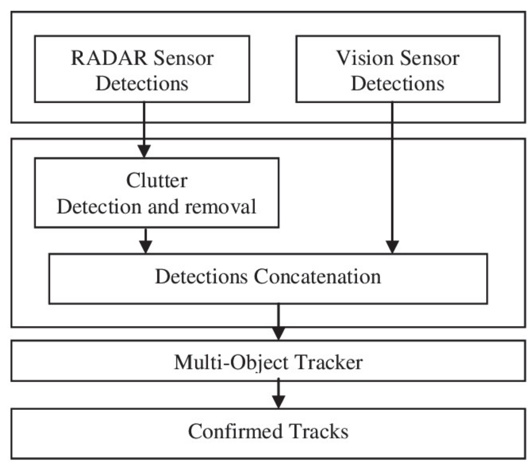

Fig. 1. Sensor Fusion Algorithm.

The design of the sensor fusion and tracking algorithms involve processing the data coming from RADAR and Vision sensors, fusing the processed sensor data and assigning tracks to the detections obtained after fusing the data from both RADAR and Vision sensors. Initially clutters are removed from the RADAR sensor data and the detections obtained from RADAR after removing clusters are fused with the Vision sensor detections. The detections are classified into tracks based on the cost matrix processing using Hungarian algorithm. Fig.1 represents the block diagram of the design. To avoid problems due to noise in the sensor data or missing sensor data, Interacting Motion Model (IMM) filter is used. Fig.2 depicts the IMM filtering process.

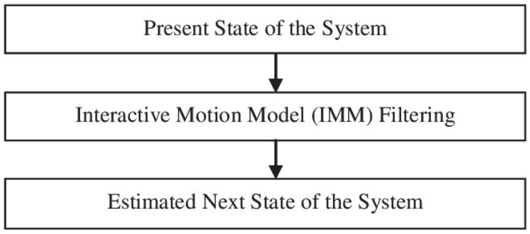

Fig. 2. IMM Filtering.

A. Removing clutters from RADAR data

Detections due to stationary objects in the surroundings of the AV are considered as clutters. Removing clutters from the RADAR data improves efficiency of the sensor fusion algorithm. This helps in focusing on the objects in the area of interest around the AV. This is achieved by identifying stationary object detections and removing them from the sensed RADAR data.

B. Assigning Tracks to Detections

For proper classification of detections, it is important to correlate the detections from consecutive samples in time. The classified detections are referred to as tracks. The newly occurring detections may be due to an already present object or due to new objects in the vicinity of AV. The assignment of tracks to detections is carried out by applying Hungarian algorithm on the Cost matrix.

1) Cost Matrix: The Cost Matrix is a matrix of order m x n, where m is the number of existing tracks and n is the number of detections. Fig.3 represents a cost matrix of order 4 x 4 where D indicates new detections, T represents existing tracks and Cxy indicates cost of assigning Tx track to Dy detection . Initially tracks are assigned to the available detections. When new detections occur, cost of assigning existing tracks to the detections is calculated.The cost of assigning a track to detection is the euclidean distance between the position of the existing track and the position of detection.

$$\begin{array}{c c} & \begin{array}{cccc} D1 & D2 & D3 & D4 \end{array} \\ \begin{array}{c} T1 \\ T2 \\ T3 \\ T4 \end{array} & \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{pmatrix} \end{array}$$

Fig. 3. Cost Matrix.

2) Hungarian Algorithm: The Hungarian algorithm is a combinatorial optimization algorithm used to solve assignment problem. From a cost matrix it finds the combination whose sum of costs is minimum. In this context it is applied to find the tracks that are closest to the new detections such that sum of euclidean distance is minimum. If the number of detections is greater than the number of existing tracks, a new track is created and is assigned to the unassigned detection. If a track is not assigned to any detections for a fixed number of times then the track is deleted. The consecutive number of times a track must not be unassigned before it is deleted is based on the algorithm design. In this project the number is taken as five.

III. IMM Filtering

The sensor readings are always not accurate. Noise may result in false sensor readings. To avoid the problems due to false readings or missing readings of the sensors a mathematical model is employed to predict the next state of the system based on the previous states of the system. In this work an Interacting Motion Model (IMM) filter is used to predict the next state of the system based on the previous state. IMM filter is a multi-model filter which has more than one Kalman filter, each with a different State Transition Model, working in parallel. Each Kalman filter estimates the next state of the system based on their respective State Transition model. The final state estimate of the system is arrived at by adding fractions of each estimate. The fractions are determined in the update step of Kalman filtering.

IV. Kalman Filtering

Kalman filtering is an algorithm that estimates the unknown variables accurately based on a series on previous measurements containing noise or other inaccuracies. Fig.4 represents the Kalman filtering process to estimate the position of an AV. Kalman filtering mainly involves two steps: Predict and Update. In the predict step, the next state and the uncertainty of a variable is predicted based on previous state values and the State Transition Matrix. In the update step the predicted state and uncertainty values are corrected to obtain accurate values of state and uncertainty of the system.

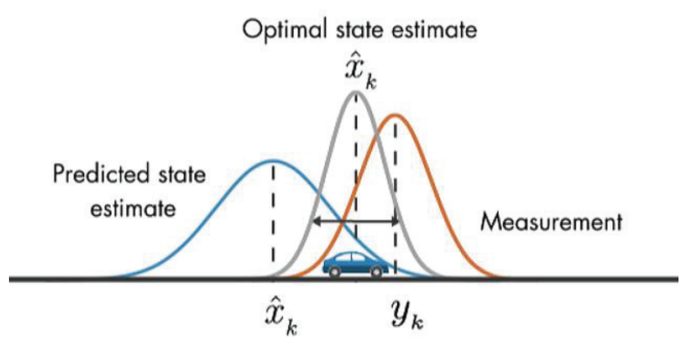

Fig. 4. Kalman Filtering.

Based on the algorithm used, Kalman filters are classified into three types:

A. Kalman filter (KF)

The basic Kalman filter (KF) provides accurate state estimates for linear systems. KF works based on Gaussian distribution and the state estimates drift away from accurate values for non-linear systems.(1) and (2) are used to predict the state estimate,x and the state uncertainty,P. (3) - (6) are used to update the predicted x and P based on the measured value.

$$x' = Fx + u \tag{1}$$

$$P' = FPF^{\mathrm{T}} + Q \tag{2}$$

$$y = z - Hx' \tag{3}$$

$$S = HP'H^{\mathrm{T}} + R \tag{4}$$

$$K = P'H^{\mathrm{T}}S^{\text{-}1} \tag{5}$$

$$x = x' + Ky \tag{6}$$

$$P = (I - KH)P' \tag{7}$$

where x is previous state estimate, x' is predicted state estimate, P is previous state uncertainty, P' is predicted state uncertainty, F is Transition matrix from t-1 to t, u is the noise added, Q is covariance matrix including noise, y is the difference between actual measurement and prediction, S is the estimated system error, H is the transition matrix, R is the covariance matrix and K is the Kalman gain.

B. Extended Kalman Filter (EKF)

Unlike KF, the EKF can estimate the state of non-linear systems by linearization of the non-linear model.(8) - (11) are used to predict the state estimate,x and the state uncertainty,P. (12) - (16) are used to update the predicted x and P based on measured value.

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \tag{8}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{9}$$

$$F_k = \frac{\partial f}{\partial x}|_{\hat{x}_{k-1|k-1}, u_k} \tag{10}$$

$$H_k = \frac{\partial h}{\partial x}|_{\hat{x}_{k-1|k-1}} \tag{11}$$

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}) \tag{12}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{13}$$

$$K_k = P_{k|k-1} H_k^T S_K^{-1} \tag{14}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{15}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{16}$$

C. Unscented Kalman Filter (UKF)

When predict and update functions and are highly nonlinear, the EKF gives poor performance. UKF employs unscented transformation (UT) to provide state estimates in such cases. UKF uses sigma vectors to arrive at the state and the uncertainty of the system.
Assume a random variable,x of dimension L propagating through a non-linear function,f such that $y = f(x)$. The sigma vectors χ_i and their respective weights W_i are defined as in (17) - (22).

$$\chi_0 = \bar{x} \tag{17}$$

$$\chi_i = \bar{x} + (\sqrt{(L+\lambda)P_x})_i \quad i = 1, ..., L \tag{18}$$

$$\chi_i = \bar{x} - (\sqrt{(L+\lambda)P_x})_{i-L} \quad i = L+1, ..., 2L \tag{19}$$

$$W_0^{(m)} = \lambda/(L+\lambda) \tag{20}$$

$$W_0^{(c)} = \lambda/(L+\lambda) + (1 - \alpha^2 + \beta) \tag{21}$$

$$W_i^{(m)} = W_i^{(c)} = 1/2(L+\lambda) \quad i = 1, ..., 2L \tag{22}$$

where$\lambda = \alpha^2(L+\kappa)$ is a primary scaling parameter, $\kappa$is a secondary scaling parameter, α denotes the spread of sigma vector points around $\bar{x}$ and β is the distribution parameter.

$$Y_i = f(X_i) \quad i = 1, ..., 2L \tag{23}$$

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^{(m)} Y_i \tag{24}$$

$$P_y \approx \sum_{i=0}^{2L} W_i^{(c)} [Y_i - \bar{y}][Y_i - \bar{y}]^T \tag{25}$$

$\bar{y}$ and P_y are the weighted mean and covariance of the non-linear function y.
Based on the State Transition Models, Kalman filters are classified into three types:

D. Constant Velocity Kalman Filter (CVKF)

In CVKF, the system is considered to be moving with a constant velocity. The state variables obey the kinematic equations having constant velocity.
The state vector for two-dimensional (2-D) CVKF is represented as follows:

State vector = [x ; v_x ; y ; v_y]

The State Transition Matrix for a 2-D constant velocity process, after a time step T, is a block matrix as shown in Fig.5.

$$\begin{bmatrix} x_{k+1} \\ v_{x,k+1} \\ y_{k+1} \\ v_{y,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ vx_k \\ y_k \\ vy_k \end{bmatrix}$$

Fig. 5. State Transition Matrix of CVKF.

E. Constant Acceleration Kalman Filter (CAKF)

In CAKF, the system is considered to be moving with a constant acceleration. The state variables obey the kinematics equations having constant acceleration.
The state vector for two-dimensional (2-D) CAKF is represented as follows:

State vector = [x ; v_x ; a_x ; y ; v_y ; a_y]

The State Transition Matrix for a 2-D constant acceleration process, after a time step T, is a block matrix as shown in Fig.6.

$$\begin{bmatrix} x_{k+1} \\ vx_{k+1} \\ ax_{k+1} \\ y_{k+1} \\ vy_{k+1} \\ ay_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ vx_k \\ ax_k \\ y_k \\ vy_k \\ ay_k \end{bmatrix}$$

Fig. 6. State Transition Matrix of CAKF.

F. Constant Turn Kalman Filter (CTKF)

In CTKF, the system is considered to be moving with a constant angular velocity. The state variables obey the kinematics equations having constant angular velocity, Omega. The state vector for two-dimensional (2-D) CAKF is represented as follows:

State vector = [x ; v_x ; y ; v_y ; omega]

EKF provides more precise results than the other two Kalman filters in this Sensor fusion Algorithm. So an IMM filter with a constant velocity EKF, a constant acceleration EKF and a constant turn EKF running in parallel is employed to predict the next state of the system.

G. Most Important Object (MIO)

Track in the same lane and closest in front of the AV is referred to as MIO. Based on Euro NCAP Test criteria, Forward Collision Warning (FCW) condition is determined based on the relative distance and relative velocity between the MIO and ALV.The Euro NCAP Test Protocol defines the forward collision warning distance as in (26).

$$d_{FCW} = 1.2 * rel\_vel + \{rel\_vel^2/2 * max\_decel\} \quad (26)$$

whererel_vel is the relative velocity between AV and the object, max_decel is the maximum deceleration that can be applied to the AV.
Three cases of FCW are used in the project:

1) Green: When there is no MIO or the distance from MIO to AV is greater than d_{FCW} and the MIO is moving away from AV.
2) Yellow: When the distance from MIO to AV is greater than d_{FCW} and the MIO is moving towards AV.
3) Red: When the distance from MIO to AV is less than d_{FCW}.

V. Simulation

MATLAB tools are used to work with the data from various sensors and different filter models. Tools facilitating multi-object tracking, simulating kalman filters with different State Transition Models are bundled into toolboxes named Sensor Fusion and Tracking Toolbox and Automated Driving Toolbox. The designed sensor fusion and tracking algorithm is tested by simulating it in MATLAB. The sensor data obtained by performing tests on a real vehicle equipped with a vision sensor, a long range and a short range RADAR sensor, is used for the testing purpose. 500 samples of RADAR detections and Vision detections each with a time step of 0.05 seconds are feed to the algorithm and the simulation results are studied.

VI. Results

Sensor data for a period of 25 seconds with a time step of 0.05 seconds obtained from real tests is used for analyzing the sensor fusion and tracking algorithm. Nearly 9800 RADAR detections and 1000 Vision detections are processed by the sensor fusion and tracking algorithm. Clutters are removed from RADAR detections resulting in approximately 6200 RADAR detections. As a result the processing load on the processor is reduced by 36%. These RADAR detections are fused with the Vision detections to get a better understanding of the AV surroundings.

Tracking involves calculation of cost matrix for the available tracks and detections and assigning tracks to the detections based on the Hungarian algorithm. This process of calculating cost matrix and assigning tracks to the detections is carried out for every sample of time. Depending on the frequency of arrival of the detections the tracks are updated i.e., new tracks are added on the arrival of new detections and old tracks are removed on the departure of existing detections. Bird's eye plot is used for displaying the outcomes of sensor fusion and tracking algorithms. The same outcomes are plotted on the recorded video of the test scenario. Fig.7 represents the bird's eye plot simulated in MATLAB. Fig.8 and Fig.9 are the snapshots of the test simulation of sensor fusion and tracking algorithm in MATLAB. In the bird's eye plot detections due to vision sensor are represented with △ ,detections due to radar are represented with ◯ and the tracked objects are represented with □ .The lane markings are represented with yellow lines and MIO is represented with red, yellow or green tracks based on d_{FCW}.

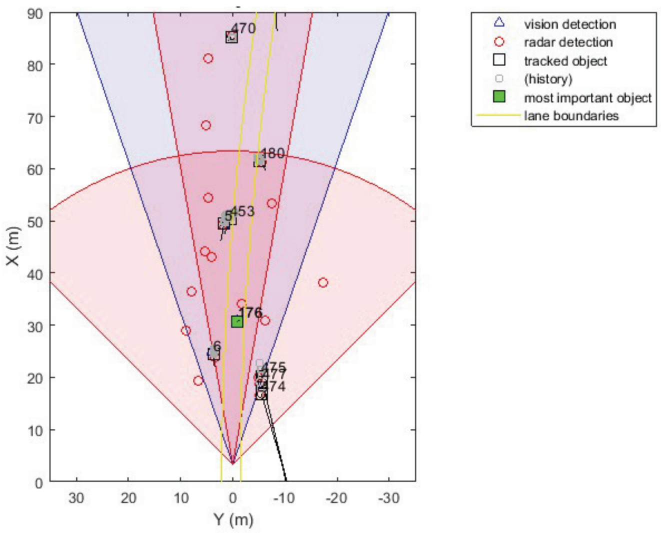

Fig. 7. Bird's Eye Plot.

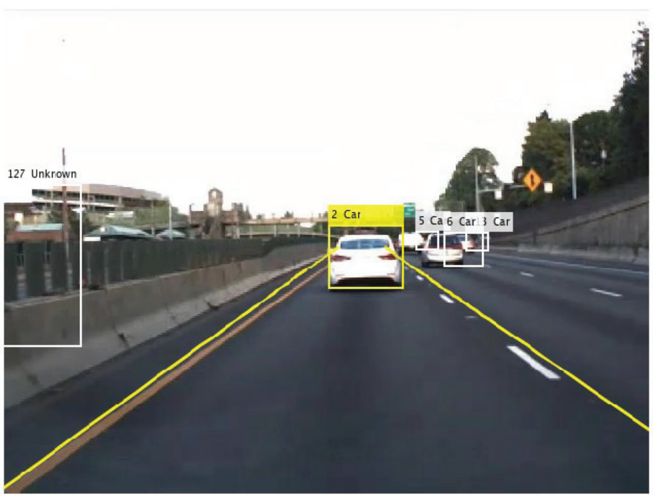

Fig. 8. Detections displayed on the Recorded Video-1.

VII. Conclusion and Future Scope

In this paper, the design and simulation results of sensor fusion and tracking algorithm in MATLAB are discussed. Automated Driving Toolbox and Sensor Fusion and Testing Toolbox available in MATLAB are used in designing the sensor fusion and tracking algorithm.The algorithm has two parts, namely sensor fusion part and tracking part. Initially sensor fusion is performed to process and fuse the RADAR and vision sensor data. Tracking is then performed on the object detections obtained from sensor fusion algorithm. An

Fig. 9. Detections displayed on the Recorded Video-2.

Interacting Motion Model(IMM) is designed containing three kalman filters namely CVKF,CAKF and CTKF to help us with missing data. The sensor data available in MATLAB is used for testing the algorithms. The multi-object tracking algorithm combines the sensor data and assigns tracks to the object detections. The IMM filter predicts the next state of the system based on the previous state and the sensor data.

The future work of this research would be to convert the designed sensor fusion and tracking algorithms from MATLAB code into either C/C++ code or python code so that the designed sensor fusion and tracking algorithms can be implemented on a hardware platform and tested. The mentioned process could be applied for designing sensor fusion and tracking algorithms to combine data from different sensors used in AVs to arrive at better understanding of the AV surroundings thereby having better control and decision making capabilities for the AV.

Acknowledgment

We sincerely thank Mr.M.R.K.Naidu, Head of Research and Development, Pedvak Technologies Pvt. Ltd. for his valuable support and guidance throughout the duration of this work which greatly helped us to present this paper.

References

[1] Kocić, N. Jovičić and V. Drndarević, "Sensors and Sensor Fusion in Autonomous Vehicles," 2018 26th Telecommunications Forum (TELFOR), Belgrade, 2018, pp. 420-425, doi: 10.1109/TELFOR.2018.8612054.
[2] R. Kumar and S. Jayashankar, "Radar and Camera Sensor Fusion with ROS for Autonomous Driving," 2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, 2019, pp. 568-573, doi: 10.1109/ICIIP47207.2019.8985782.
[3] N. S. Zewge, Y. Kim, J. Kim and J. Kim, "Millimeter-Wave Radar and RGB-D Camera Sensor Fusion for Real-Time People Detection and Tracking," 2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA), Daejeon, Korea (South), 2019, pp. 93-98, doi: 10.1109/RITAPP.2019.8932892.

[4] K. Kim, C. Lee, D. Pae and M. Lim, ”Sensor fusion for vehicle tracking with camera and radar sensor,” 2017 17th International Conference on Control, Automation and Systems (ICCAS), Jeju, 2017, pp. 1075-1077, doi: 10.23919/ICCAS.2017.8204375.

[5] B. Suwandi, T. Kitasuka and M. Aritsugi, ”Low-cost IMU and GPS fusion strategy for apron vehicle positioning,” TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, 2017, pp. 449-454, doi: 10.1109/TENCON.2017.8227906.

[6] K. Saadeddin, M. F. Abdel-Hafez and M. A. Jarrah, ”Estimating vehicle state by GPS/IMU fusion with vehicle dynamics,” 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, 2013, pp. 905-914, doi: 10.1109/ICUAS.2013.6564776.

[7] A. Chandra Babu, R. K. Karri and N. M.S., ”Sensor Data Fusion Using Kalman Filter,” 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, 2018, pp. 29-36, doi: 10.1109/ICDI3C.2018.00015.

[8] D. Barbosa, A. Lopes and R. E. Araújo, ”Sensor fusion algorithm based on Extended Kalman Filter for estimation of ground vehicle dynamics,” IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, 2016, pp. 1049-1054, doi: 10.1109/IECON.2016.7793145.

[9] A. Assa and F. Janabi-Sharifi, ”A Kalman Filter-Based Framework for Enhanced Sensor Fusion,” in IEEE Sensors Journal, vol. 15, no. 6, pp. 3281-3292, June 2015, doi: 10.1109/JSEN.2014.2388153.

[10] H. Chen, C. Xue, S. Liu, Y. Sun and Y. Chen, ”Multiple-object Tracking based on Monocular Camera and 3-D Lidar Fusion for Autonomous Vehicles,” 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 2019, pp. 456-460, doi: 10.1109/ROBIO49542.2019.8961438.

[11] H. Karunasekera, H. Wang and H. Zhang, ”Multiple Object Tracking With Attention to Appearance, Structure, Motion and Size,” in IEEE Access, vol. 7, pp. 104423-104434, 2019, doi: 10.1109/ACCESS.2019.2932301.

[12] B. Suwandi, W. S. Pinastiko and R. Roestam, ”OBD-II Sensor Approaches for The IMU and GPS Based Apron Vehicle Positioning System,” 2019 International Conference on Sustainable Engineering and Creative Computing (ICSECC), Bandung, Indonesia, 2019, pp. 251-254, doi: 10.1109/ICSECC.2019.8907036.

[13] M. Taraba, J. Adamec, M. Danko and P. Drgona, ”Utilization of modern sensors in autonomous vehicles,” 2018 ELEKTRO, Mikulov, 2018, pp. 1-5, doi: 10.1109/ELEKTRO.2018.8398279.

[14] M. L. Fung, M. Z. Q. Chen and Y. H. Chen, ”Sensor fusion: A review of methods and applications,” 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, 2017, pp. 3853-3860, doi: 10.1109/CCDC.2017.7979175.

[15] J. Alonso-Mora, A. Wallar and D. Rus, ”Predictive routing for autonomous mobility-on-demand systems with ride-sharing,” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 3583-3590, doi: 10.1109/IROS.2017.8206203.

[16] M. Daily, S. Medasani, R. Behringer and M. Trivedi, ”Self-Driving Cars,” in Computer, vol. 50, no. 12, pp. 18-23, December 2017, doi: 10.1109/MC.2017.4451204.

[17] R. O. Chavez-Garcia and O. Aycard, ”Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking,” in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 2, pp. 525-534, Feb. 2016, doi: 10.1109/TITS.2015.2479925.

[18] D. Bajpayee and J. Mathur, ”A comparative study about autonomous vehicle,” 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2015, pp. 1-6, doi: 10.1109/ICIIECS.2015.7193002.

[19] K. Bimbraw, ”Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology,” 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Colmar, 2015, pp. 191-198.

[20] J. Burlet and M. DallaFontana, ”Robust and efficient multi-object detection and tracking for vehicle perception systems using radar and camera sensor fusion,” IET and ITS Conference on Road Transport Information and Control (RTI 2012), London, 2012, pp. 1-6, doi: 10.1049/cp.2012.1553.

[21] A.Yilmaz, ”Sensor Fusion in Computer Vision,” 2007 Urban Remote Sensing Joint Event, Paris, 2007, pp. 1-5, doi:10.1109/URS.2007.37183.