IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 June 2023, accepted 18 August 2023, date of publication 24 August 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3308152

RESEARCH ARTICLE

Adaptive Decentralized Sensor Fusion for Autonomous Vehicle: Estimating the Position of Surrounding Vehicles

KYUSANG YOON[1], JAEHO CHOI[1], AND KUNSOO HUH[2], (Member, IEEE)
[1]Department of Automotive Engineering (Automotive-Computer Convergence), Hanyang University, Seoul 04763, Republic of Korea
[2]Department of Automotive Engineering, Hanyang University, Seoul 04763, Republic of Korea

Corresponding author: Kunsoo Huh (khuh2@hanyang.ac.kr)

This work was supported by the Ministry of Trade, Industry, and Energy (MOTIE), South Korea, through the Technology Innovation Program (Development on Automated Driving with Perceptual Prediction Based on T-Car/Vehicle Parts to Intelligent Control/System Integration for Assessment) under Grant 20018101.

ABSTRACT The tracking accuracy of nearby vehicles determines the safety and feasibility of driver assistance systems or autonomous vehicles. Recent research has been active to employ additional sensors or to combine heterogeneous sensors for more accurate tracking performance. Especially, autonomous driving technologies require a sensor fusion technique that considers various driving environments. In this research, a novel method for high-level data fusion is proposed to improve the accuracy of tracking surrounding vehicles. In response to the changing driving environment, the locations of the vehicles are estimated in real-time using an adaptive track-to-track fusion technique and an interacting multiple model filter. Asynchronous measurements from multiple sensors such as radar, camera, and LiDAR, are utilized for the estimation. For each sensor, two motion models representing the vehicle's movement are applied to increase the estimation accuracy. Utilizing a multimodal network-based track-to-track fusion approach, it combines the estimates of the target vehicle position from each sensor into a single estimate. The inputs of the network are intended to determine the reliability of each sensor, considering the driving conditions that may affect sensor accuracy. Also, multiple embeddings in the network are created so that the corresponding data maintains its relevance and enables the real-time computing. The proposed method is verified using real driving data collected from various environments.

INDEX TERMS Perception, sensor fusion, autonomous vehicle, advanced driver assistance system, track-to-track fusion, interacting multiple model filter, multimodal learning.

I. INTRODUCTION

As advanced driver assistance system (ADAS) and autonomous driving technology advances, there is a growing need for reliable perception techniques. Recognizing the surrounding environment should be accurate because path prediction and control strategies are established based on the result. There exist several perception technologies for estimating the accurate position of a surrounding vehicle.

Research on estimating a state value has been actively conducted in the perception area, and techniques for enhancing the estimation performance include combining multiple models and using various sensors. In the case of estimation using a single model, an error occurs if an object shows a different motion that the model cannot express. Instead, there have been efforts to utilize several motion models such as the Interacting Multiple Model (IMM) [1]. Research on the use of the IMM algorithm to improve the estimation performance is being actively conducted [2], [3], [4]. For example, Kaempchen et al. [2] combined estimates from linear and non-linear motion models to obtain the precise

The associate editor coordinating the review of this manuscript and approving it for publication was Xiangxue Li.

speed of a stop-and-go maneuvering object through the IMM filter. Jo et al. [3] reduced the localization error by employing kinematic and dynamic models with the IMM filter to cover both high-speed and low-speed cases. Xu et al. [4] estimated not only the vehicle's position but also its velocity, acceleration, yaw rate, and heading angle by using the constant turn rate acceleration model in the IMM framework with the parameter adaptation scheme.

In the case of using various sensors, a lot of research has been conducted to produce better estimation results than using a single sensor. It is essential to determine the features of each sensor and develop complementary perception algorithms, a process known as sensor fusion. The fusion system can be classified according to the fusion structure [5]. In particular, the high-level sensor fusion can have either a centralized or a decentralized fusion structure depending on the estimator function [6]. In the centralized approach, a single estimator uses observations from multiple sensors to compute the state estimates. On the other hand, the decentralized approach includes a separate estimator for each sensor measurement. Traditionally, decentralized structures have been preferred for their computational efficiency and mathematical convenience. For the final state, each local track needs to be merged, called track-to-track fusion, and its research on estimating the optimal state has been actively conducted in various ways [7], [8], [9], [10], [11], [12].

In driving environments, the sensor's detection accuracy varies based on sensing conditions such as distance. Even if the situation does not change, it is difficult to accurately determine the cross-covariance of sensors; thus, computing an optimal state is challenging. Julier et al. [7] suggested the Covariance Intersection (CI) method which combines the two local states with an unknown cross-covariance. The approach has been widely used because it does not require computing the cross-covariance of sensors. Chen et al. [8] proved the solution of the CI method computationally efficient in that it searches for one dimension rather than the whole parameter space. In addition, several studies have been conducted to reduce the computational burden of the CI approach [13]. CI approaches allow the optimal value to be determined by reducing the trace or determinant of the error variance, which requires a non-linear convex optimization process. Niehsen et al. [14] and Franken et al. [15] proposed methods to reduce the computational cost and speed up the calculation. In addition, the ellipsoidal intersection method [9], [10] was introduced with specifying the unknown correlation as an error term. The inverse covariance intersection method [11], [12] was also suggested to calculate the optimal weight through the boundary of the inverse covariance ellipsoid.

CI method is widely used to obtain the sub-optimal track-to-track fusion weights. However, since the CI method numerically calculates the optimal value based on local covariance, implementing it for tracking in autonomous driving raises several problems. The calculated optimal result has errors because the covariance of the sensor can change depending on the circumstances. The CI method cannot cope with the changing environment because the covariance converges within a few steps according to the initial setting. Besides, since the result value is calculated numerically, the computation cost increases rapidly as the number of sensors increases. ADAS and autonomous driving generally require a large number of sensors to achieve better perception performance and, thus, the CI approach for track-to-track fusion is not adequate for this task.

In recent years, many studies have used various machine learning approaches for sensor fusion. In particular, multimodal learning is a type of machine learning that utilizes not only unimodal information, but also data with multiple modalities, such as text, image, and audio [16]. There have been many studies on how to effectively utilize multimodal information and achieve better results [17], [18], [19]. For example, Akbari et al. [17] utilized multimodal data and extracted multimodal representations via transformers architecture. Based on these representations, it showed good performance in a variety of tasks, including video-action recognition and audio event classification. Recently, various methods for sensor fusion have been reported and each feature is compared in Table 1.

In this paper, in order to overcome the limitations of the CI method, an estimation algorithm is proposed using a multimodal learning technique while taking advantage of the IMM filter. The IMM filter is designed first to combine estimates from multiple models because IMM-based estimators have strengths in tracking motions. Secondly, the multimodal learning approach is used to address the shortcomings of the existing track-to-track fusion technique. The multimodal learning is applied because sensor data with multiple modalities can be comprehensively utilized for determining the optimal reliability of each sensor.

As a decentralized sensor fusion, after the high-level data is acquired from camera, LiDAR, and radar, the positions of surrounding vehicles are estimated using an IMM filter and the adaptive track-to-track fusion. The overall architecture of the proposed algorithm is illustrated in Fig.1.

The main contributions of this paper can be summarized as follows:

- Based on the high-level data from LiDAR, radar, and camera, the IMM filter is designed separately to combine the position estimation using multiple models.
- Using multimodal learning in track-to-track fusion, the proposed method sets the weight value for each sensor considering the characteristics of the driving environment: the shape of the road, the behavior and position of the host and surrounding vehicles, and the weather.
- The adaptive track-to-track fusion method is developed to compensate for uncertainties in process models and measurement sensors.
- The performance of the proposed method is verified using actual vehicle data.

TABLE 1. Comparison of recent sensor fusion methods.

Ref.	Types of sensor	Fusion structure	Goal	Input data	Purpose of the network	Characteristic
[20]	Camera, LiDAR	Intermediate fusion	Object detection	Raw data	Extract feature	Separately extract features from each modality using its own network, and then integrate these networks using additional central networks
[21]	Trifocal camera, fisheye camera, radar, LiDAR	High-level fusion	Tracking	High-level data, raw image, point cloud	Get fused object state	Combining two kinds of sensor data fusion methods: a model-based approach utilizing the unscented kalman filter and a data-driven approach to stabilize the position
[22]	Electro-optical camera, radar	Mid-level fusion	Tracking	Raw data	-	Using adaptive Kalman Filter (KF), considering sampling rate and signal loss, to adjust the fusion weights of various sensors based on their reliability and the characteristics of the tracked objects
[23]	Infrared camera, LiDAR	Low-level fusion	Object detection	Raw data	-	Direct external parameter correction algorithm between infrared cameras and LiDAR sensors
[24]	LiDAR, camera	Low-level fusion	Semantically enhanced 3D point cloud	Raw data	2D panoptic segmentation, point cloud segmentation	Fusing geometry information from 3D point clouds with semantic data from multiple cameras
[25]	Camera, radar	High-level fusion	Tracking	Raw data	YOLO algorithm to obtain 2D spatial position	Tri-KF-based Hungarian algorithm to associate predicted positions with measurements
Our method	Camera, LiDAR, radar	High-level fusion	Tracking	High-level data, lane coefficient, covariance value, raw image	To calculate the optimal reliability of each sensor	Obtaining local tracks from IMM-KF and merging them using multimodal learning

The rest of this paper is organized as follows. Section II explains the IMM filter design using multiple models and the track-to-track fusion method based on the multimodal learning. In Section III, the performance of the proposed algorithm is verified through real-vehicle experimental data obtained in various environments.

II. PROPOSED METHOD

To reduce the detection errors of the surrounding vehicles' positions using multiple sensors, a novel sensor fusion method based on multimodal learning are designed. The overall architecture of this method is shown in Fig.1. Using three types of sensors, including camera, LiDAR, and radar, each sensor calculates its own local state using the IMM filter. The local states are integrated into the final state through multimodal learning at the final stage of the process.

A. ASYNCHRONOUS MULTI-SENSOR FUSION WITH IMM FILTERS

The IMM approach combines multiple models to reduce the error that can arise when estimating with only a single model. Because vehicle motion characteristics change frequently over time, it is preferable to estimate their states using an IMM approach rather than a single model. Its algorithm is largely divided into four steps: interaction, model specific filtering, model probability update, and combination [2]. Unlike a typical single-model-based KF [26], there are additional considerations for the IMM method that utilizes multiple models. To combine many models, the model probability, which represents the reliability of each mode, must be computed. Then, the final estimate is computed by considering the Transition Probability Matrix (TPM) which is the prior knowledge of the transition possibility of each mode.

In this study, the location of the surrounding vehicle is estimated through high-level data obtained from three sensors: camera, radar, and LiDAR. The decentralized sensor fusion approach is selected because it is robust to sensor faults and does not require relatively high processing data rates [27]. Local states for each sensor are estimated asynchronously through the IMM filter.

Driving maneuvers can be described as the motion models. There are numerous models such as constant velocity (CV), constant acceleration (CA), constant turn rate and velocity (CTRV), and constant turn rate and acceleration (CTRA) models, which are made on the assumption that a specific motion is maintained for a certain amount of time [28]. In this study, vehicle motion is expressed using two models: the CV and CA models. Even if these models are simple, they can exhibit general driving conditions. Besides, they do not need coordinate transformation with the heading angle information which is not readily available from the in-vehicle sensors. To illustrate the motion of the vehicle, the state is selected as:

$$x_k = \begin{bmatrix} x & y & V_x & V_y & a_x & a_y \end{bmatrix}^T, \tag{1}$$

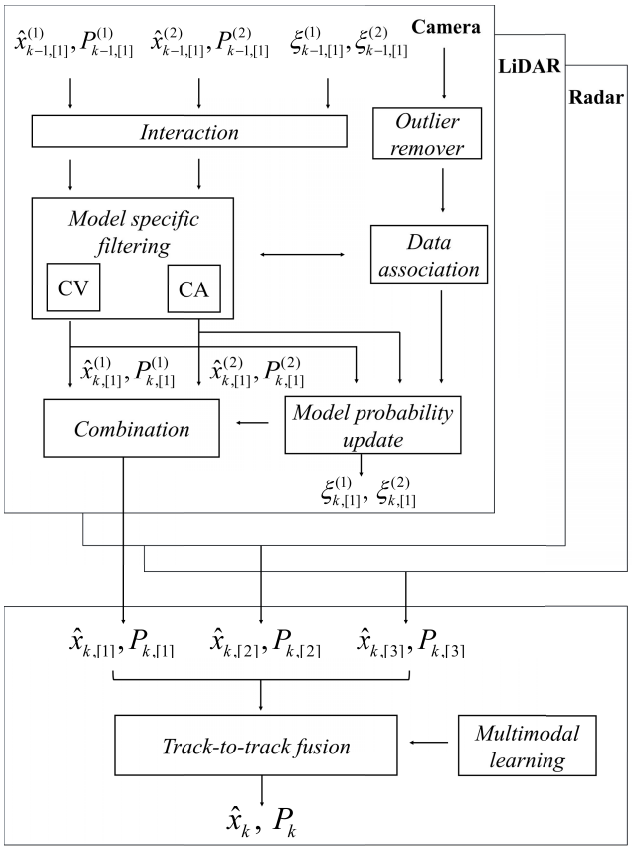

FIGURE 1. Overall architecture of the proposed method with the IMM filter and the adaptive track-to-track fusion module.

where x : longitudinal position of the vehicle
y : lateral position of the vehicle
V_x : longitudinal velocity of the vehicle
V_y : lateral velocity of the vehicle
a_x : longitudinal acceleration of the vehicle
a_y : lateral acceleration of the vehicle

- Interaction

In general driving situations, the probability of maintaining the current motion is higher than the probability of not. In this study, it is assumed that each vehicle's behavior has an 80% chance of remaining unchanged in the next step and a 20% chance of switching to another one. Thus, the TPM is assumed as follows between CV and CA models.

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}, \tag{2}$$

where Π : TPM
π_{ij} : transition probability from mode j to mode i
The initial state and the covariance of each mode are updated using the formulations [2].

$$\hat{x}^{(0i)}_{k-1,[\alpha]} = \sum_{j=1}^{2} \xi^{ij}_{k-1,[\alpha]} \hat{x}^{(j)}_{k-1,[\alpha]} \tag{3}$$

$$P^{(0i)}_{k-1,[\alpha]} = \sum_{j=1}^{2} \xi^{ij}_{k-1,[\alpha]} [P^{(j)}_{k-1,[\alpha]} + (\hat{x}^{(j)}_{k-1,[\alpha]} - \hat{x}^{(0i)}_{k-1,[\alpha]}) \cdot (\hat{x}^{(j)}_{k-1,[\alpha]} - \hat{x}^{(0i)}_{k-1,[\alpha]})^T] \tag{4}$$

with $\xi^{ij}_{k,[\alpha]} = \eta^{i}_{[\alpha]} \pi_{ij} \xi^{j}_{k,[\alpha]}$,
$(\eta^{i}_{[\alpha]})^{-1} = \sum_{j=1}^{2} \pi_{ij} \xi^{j}_{k,[\alpha]}$,
where $\hat{x}^{(0i)}_{k-1,[\alpha]}$: mixed initial state of each mode
$P^{(0i)}_{k-1,[\alpha]}$: mixed initial covariance of each mode
$\xi^{j}_{k,[\alpha]}$: model probability for mode i and sensor α
$\xi^{ij}_{k,[\alpha]}$: mixing probability from mode j to i for sensor α
$\eta^{i}_{[\alpha]}$: normalizing constant
superscript i, j : mode state (1: CV model, 2: CA model)

k is a sequence index and subscript $[\alpha]$ represents the type of sensor with $\alpha \in \{1, 2, 3\}$, which refer to camera, LiDAR, and radar, respectively.

- Model specific filtering

The CV model is based on the assumption that the target moves at a constant speed and the motion of the vehicle is described below.

$$\bar{x}^{(1)}_{k,[\alpha]} = \Phi^{(1)}_{k,[\alpha]} \hat{x}^{(01)}_{k-1,[\alpha]}, \tag{5}$$

with the state transition matrix:

$$\Phi^{(1)}_{k,[\alpha]} = \begin{bmatrix} 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{6}$$

where dt is the time step.

The CA model is a model created under the assumption that an object moves with constant acceleration and the motion of the vehicle is described below.

$$\bar{x}^{(2)}_{k,[\alpha]} = \Phi^{(2)}_{k,[\alpha]} \hat{x}^{(02)}_{k-1,[\alpha]}, \tag{7}$$

with the state transition matrix:

$$\Phi^{(2)}_{k,[\alpha]} = \begin{bmatrix} 1 & 0 & dt & 0 & 0.5dt^2 & 0 \\ 0 & 1 & 0 & dt & 0 & 0.5dt^2 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Using the initial state obtained in the interaction step, the state of each mode at the current step k can be predicted using the KF formulation [26]. Also, the covariance can be predicted based on the previous value.

$$\bar{x}^{(i)}_{k,[\alpha]} = \Phi^{(i)}_{k,[\alpha]} \hat{x}^{(0i)}_{k-1,[\alpha]} \tag{9}$$

$$\bar{P}^{(i)}_{k,[\alpha]} = \Phi^{(i)}_{k,[\alpha]} P^{(i)}_{k-1,[\alpha]} (\Phi^{(i)}_{k,[\alpha]})^T + Q^{(i)}_{k,[\alpha]} \tag{10}$$

Then, the state and covariance are corrected using the measurement vector.

$$\hat{x}^{(i)}_{k,[\alpha]} = \bar{x}^{(i)}_{k,[\alpha]} + L^{(i)}_{k,[\alpha]}(z^{(i)}_{k,[\alpha]} - C_k\bar{x}^{(i)}_{k,[\alpha]}) \tag{11}$$

$$P^{(i)}_{k,[\alpha]} = \bar{P}^{(i)}_{k,[\alpha]} - L^{(i)}_{k,[\alpha]}\bar{S}^{(i)}_{k,[\alpha]}(L^{(i)}_{k,[\alpha]})^T \tag{12}$$

with $\bar{S}^{(i)}_{k,[\alpha]} = C_k\bar{P}^{(i)}_{k,[\alpha]}C^T_k + R_{k,[\alpha]}$,
$L^{(i)}_{k,[\alpha]} = \bar{P}^{(i)}_{k,[\alpha]}C^T_k(\bar{S}^{(i)}_{k,[\alpha]})^{-1}$,
where $z^{(i)}_{k,[\alpha]}$: measurement vector
$R_{k,[\alpha]}$: covariance of the observation noise
$Q^{(i)}_{k,[\alpha]}$: covariance of the process noise
$\bar{S}^{(i)}_{k,[\alpha]}$: residual covariance
$L^{(i)}_{k,[\alpha]}$: Kalman gain matrix

In this calculation, the value of the $Q^{(i)}_{k,[\alpha]}$ is set using the mathematical formulation [29], and the value of the $R_{k,[\alpha]}$ is heuristically selected. C_k is the observation matrix and is defined as:

$$C_k = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \end{bmatrix}.$$

- Model probability update

The model probability for mode i and sensor type α can be obtained based on the residual covariance [2]:

$$\xi^i_{k,[\alpha]} = \eta_{[\alpha]}N(z^{(i)}_{k,[\alpha]}; \bar{z}^{(i)}_{k,[\alpha]}, \bar{S}^{(i)}_{k,[\alpha]}) \cdot \sum_{j=1}^{2}\pi_{ij}\xi^j_{k-1,[\alpha]} \tag{13}$$

with

$$\eta^{-1}_{[\alpha]} = \sum_{i=1}^{2}(N(z^{(i)}_{k,[\alpha]}; \bar{z}^{(i)}_{k,[\alpha]}, \bar{S}^{(i)}_{k,[\alpha]}) \cdot \sum_{j=1}^{2}\pi_{ij}\xi^j_{k-1,[\alpha]}),$$

$$N(z^{(i)}_{k,[\alpha]}; \bar{z}^{(i)}_{k,[\alpha]}, \bar{S}^{(i)}_{k,[\alpha]}) = \frac{1}{\sqrt{2\pi\bar{S}^{(i)}_{k,[\alpha]}}}exp[-\frac{1}{2}(z^{(i)}_{k,[\alpha]} - \bar{z}^{(i)}_{k,[\alpha]})^T(\bar{S}^{(i)}_{k,[\alpha]})^{-1} \times (z^{(i)}_{k,[\alpha]} - \bar{z}^{(i)}_{k,[\alpha]}),$$

where $\eta_{[\alpha]}$ is the normalizing constant, and $N(z^{(i)}_{k,[\alpha]}; \bar{z}^{(i)}_{k,[\alpha]}, \bar{S}^{(i)}_{k,[\alpha]})$ is the likelihood of the mode i.

- Combination

As the final step, the final state and covariance are computed as

$$\hat{x}_{k,[\alpha]} = \sum_{i=1}^{2}\xi^i_{k,[\alpha]}\hat{x}^{(i)}_{k,[\alpha]}, \tag{14}$$

$$P_{k,[\alpha]} = \sum_{i=1}^{2}\xi^i_{k,[\alpha]}[P^{(i)}_{k,[\alpha]} + (\hat{x}^{(i)}_{k,[\alpha]} - \hat{x}_{k,[\alpha]}) \cdot (\hat{x}^{(i)}_{k,[\alpha]} - \hat{x}_{k,[\alpha]})^T]. \tag{15}$$

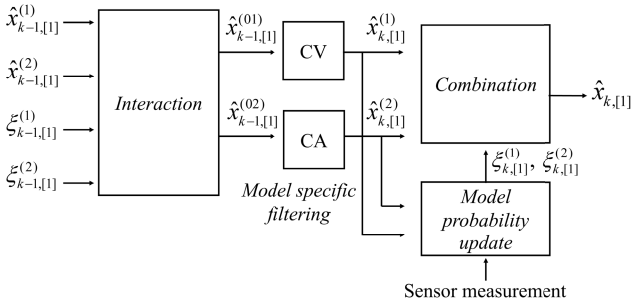

FIGURE 2. Structure of IMM filter for camera sensor case.

Fig.2 illustrates the configuration of the IMM filter using camera data. The local track state is obtained through the interaction, model specific filtering, model probability update, and combination processes. It should be noted that the local tracks for each sensor are predicted and corrected through IMM filters at different frequencies.

B. ADAPTIVE TRACK-TO-TRACK FUSION

In the decentralized sensor fusion architecture, a track-to-track fusion method is required to merge the local tracks estimated by each sensor. In particular, to minimize the local error covariance obtained by track-to-track fusion, cross-covariance information of the sensors is needed. Because determining the cross-covariance among sensors is typically challenging, the CI methods [7], [8] have been utilized to reduce the upper bound of the fused error matrix instead and to acquire optimal weights without computing the cross-covariance. The fused track based on the CI method is calculated as [30]

$$\hat{x}_{k,CI} = (\sum_{\alpha=1}^{3}{w_\alpha}^*P^{-1}_{k,[\alpha]})^{-1}\sum_{\alpha=1}^{3}{w_\alpha}^*P^{-1}_{k,[\alpha]}\hat{x}_{k,[\alpha]}. \tag{16}$$

${w_\alpha}^*$ is the optimized weight and is calculated as

$$w^*_\alpha = \arg\max_{w_\alpha\in[0,1]} tr(\sum_{\alpha=1}^{3}w_\alpha P^{-1}_{k,[\alpha]}), \tag{17}$$

where w_α is the free parameter with $\sum_{\alpha=1}^{3}w_\alpha = 1$.

However, in the above formulations, covariance values can change depending on driving conditions and their values from (12) may be inaccurate due to erroneous process noise and observation noise covariances. Therefore, instead of using (17), the weight values of each sensor are determined using multimodal learning in this study. The variables that greatly affect the covariance on the driving environment are considered as inputs to the network. In the model specific filtering part, the covariance varies if the model, process noise, or observation noise changes. For instance, driving environment affecting the process noise and observation noise are listed in Table 2. In addition, because IMM combines several models, merging the local covariance of

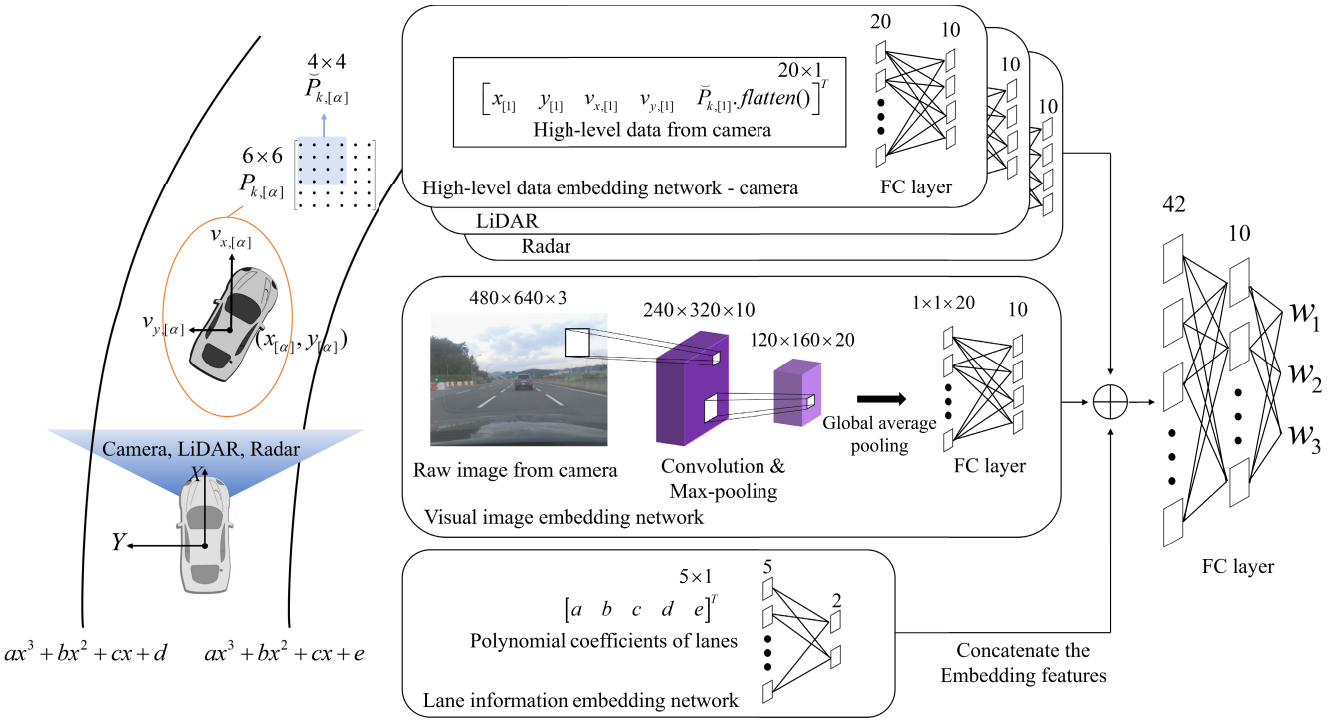

FIGURE 3. Configuration of the multimodal network with five embedding networks for high-level data, visual image, and lane information.

TABLE 2. Examples of covariance changes.

Factors that change process noise	Road geometry (straight/curve)
	Behavior of the target vehicle and the host vehicle
Factors that change observation noise	Reflective material of the target vehicle surface
	The angle between the target vehicle and the host vehicle
	Distance between the target vehicle and the host vehicle
	Weather (sunny/rainy/foggy)

each model can cause a change in the fused covariance of (15).

Considering the factors listed in Table 2, as inputs to the network, high-level data from sensors (LiDAR, radar, camera), lane information from the camera, and raw images from the camera are selected. Specifically, covariance matrix elements are used to implement the CI concept. Here, we exclude the acceleration component and use $\breve{P}_{k,[\alpha]}$ since sensors cannot measure acceleration. In addition, the position and movement characteristics of surrounding vehicles are expressed using the state of each sensor's local track obtained through the IMM filter: $x_{[\alpha]}$, $y_{[\alpha]}$, $v_{x,[\alpha]}$, $v_{y,[\alpha]}$. Based on the lane information such as the polynomial coefficients of lanes acquired from the camera, it is possible to determine if the current driving road is curved or straight. Furthermore, the raw images from the camera are utilized to express the information such as weather, location of surrounding vehicles, and motion properties. The overall architecture of the proposed network is illustrated in Fig.3.

To extract the information from visual images, a convolutional neural network (CNN) is used for extracting the image features. Compared to CNN with many layers, CNN with a shallow layer is more effective at this task in terms of computing speed and performance. Thus, CNN with only four layers is used. Encoders are used to reduce the number of parameters to adapt to a dynamic environment and provide a real-time calculation. As shown in Fig.3, five embedding networks are designed to ensure that similar information remains relevant. The size of the input and the number of feature maps are also denoted. To obtain the final embedding features, a method of concatenating embedding features with different characteristic data is applied.

The final output of the network is the weight value of each sensor. The loss function is the sum of L2 loss and normalized loss as shown in (18). The L2 loss is expressed as the surrounding vehicle positions from the DGPS sensor and the estimated track position. Utilizing the normalized loss, which consists of the normalizing constant and the weight value of each sensor, prevents the case where the weight value of a particular sensor approaches 1.

$$loss = \sum_{i=0}^{1} [\tilde{\tilde{x}}_k(i) - \hat{x}_k(i)]^2 + \lambda \sum_{j=0}^{2} w_j{}^2, \tag{18}$$

where $\tilde{\tilde{x}}_k(i)$ is the ground truth value from the DGPS sensor, $\hat{x}_k(i)$ is the ith index value of the estimated state, λ is

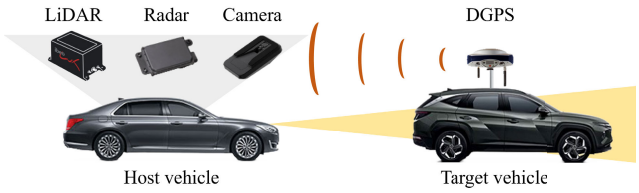

FIGURE 4. Sensor installation for experimental set-up.

the normalizing constant, and w_j is the weight value of each sensor calculated from the proposed network. In this formulation, the value of λ is set to 0.1.

To minimize the loss function, we use an Adam optimizer. The learning rate is set to 0.001, and the batch size is set to 16. The model is trained for 30 epochs. This methodology was developed using ROS-based C++ and Python scripts on Ubuntu 20.04. Using the network depicted in Fig.3, Python was used to calculate the optimal weight values for each sensor, while the remaining parts were written in C++.

III. EXPERIMENTAL VALIDATION

The proposed algorithm is validated using custom data from actual vehicles. As illustrated in Fig.4, the experimental vehicle with radar, LiDAR, and the camera is utilized to measure the position of the target vehicle. The specifications of the sensors are listed in Table 3. The target vehicle's position is additionally acquired using the DGPS sensor as the ground truth data. For verification in various situations, the custom data is collected on sunny days, rainy days, curved roads, and straight roads. The collected data consists of various driving scenarios such as lane changing, acceleration, and deceleration, while the host vehicle and target vehicle are driven independently. At the moment of GPS data acquisition, the target vehicle's location from the GPS sensor, the estimated state calculated from the CI method ($\hat{x}_{k,CI}$), and the estimated state calculated from the multimodal network ($\hat{x}_k$) were stored in a single sample. The total number of data samples obtained is 29,536, with approximately 80% used for training and 20% for testing. The proving ground was chosen as the main location for data collection because there are no adjacent tall buildings, which leads to the DGPS sensor's good accuracy, and because the absence of close vehicles allows for data collection in a safe environment. The proposed method is operated on a PC and the computing specifications are shown in Table 3. The computation time is measured in the GPU environment.

The performance is expressed using the mean square error (MSE) and maximum error between the surrounding vehicle position obtained from the DGPS sensor and the estimated track position.

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(z_k - \hat{z}_k)^T(z_k - \hat{z}_k), \tag{19}$$

where z_k is the ground truth position of the target vehicle by the DGPS sensor, and $\hat{z}_k$ is the estimated output state

TABLE 3. Experimental specifications.

Description	Specification		Sampling time
Sensor	Camera	Mobileye 500 Series	55 ms
	LiDAR	Ibeo Lux 2010	90 ms
	Radar	Delphi ESR 2.5 LRR	50 ms
	DGPS	MBC TDR 3000	100 ms
Processor	GPU	NVIDIA GeForce GTX 1080 Ti	-
	Memory	16GB	-
	OS	Ubuntu 20.04 LTS	-

TABLE 4. MSE for each scenario: (a) straight road on a sunny day (b) curved road on a rainy day (c) curved road on a sunny day.

Method	MSE [m]		
	(a)	(b)	(c)
CI	0.3316	0.5887	0.5406
Proposed	0.2057	0.4104	0.4074

defined as

$$\hat{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \hat{x}_k. \tag{20}$$

The accuracy of the proposed method is compared with the CI approach [8]. In the case of the CI, random sampling with a sample size of 30 is utilized for the numerical optimization. Fig.5 shows the driving environment and estimation errors under different conditions such as weather and road shape. Compared to the result of track-to-track fusion with the CI method, the proposed technique reduces both the total error value and the maximum error. MSE values of the proposed method are compared with the CI approach in Table 4. In each case, the suggested approach reduces the MSE by $25\% \sim 40\%$.

Even with the improved CI method, the covariance value cannot accurately reflect the driving situation. On the other hand, in this study, the adaptive track-to-track fusion method is developed to respond to changes in the surrounding environment. Utilizing multimodal information, the proposed method sets the weight value for each sensor in consideration of diverse driving conditions. The weight value of each sensor is dominantly influenced by the relative distance, as shown in Fig.6. As the relative distance increases, the camera's reliability decreases, whereas the LiDAR's reliability increases. The reliability of radar does not demonstrate a significant difference. Specifically, when the relative distance is approximately 35 m, the weight value of LiDAR is 0.47 and the weight value of the camera is 0.17. When the relative distance is about 5 m, the weight value of LiDAR is 0.37 and the weight value of the camera is near 0.3. While the relative distance changes, the reliability of radar remains relatively stable between 0.33 and 0.36. Table 5 shows the average weight value for each sensor. Regardless of the considered driving environment, LiDAR has the highest weight value, whereas the camera has the lowest weight value. It should be noted that when it rains, the weight value of the LiDAR drops, the camera increases, and the radar appears to remain constant. In comparison to the change

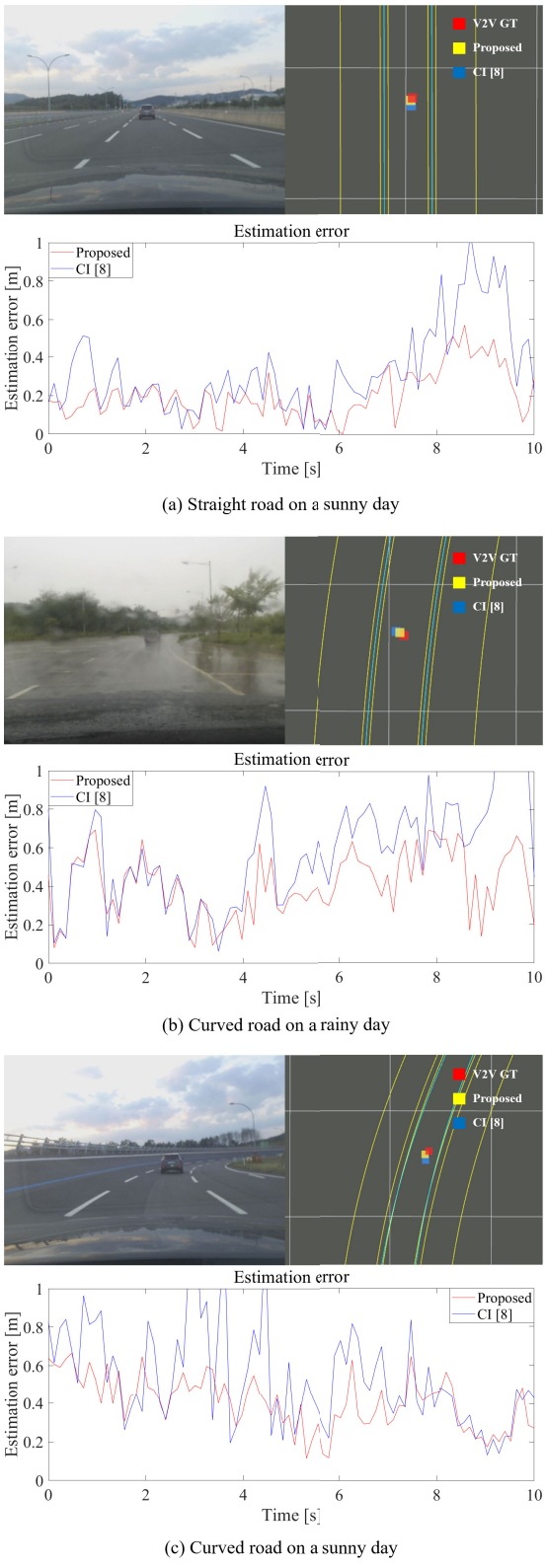

FIGURE 5. Estimation results for different scenario cases: (a) straight road on a sunny day (b) curved road on a rainy day (c) curved road on a sunny day.

caused by the relative distance, the weight is not very sensitive to the weather or road shape. Fig.7 shows the tracking results

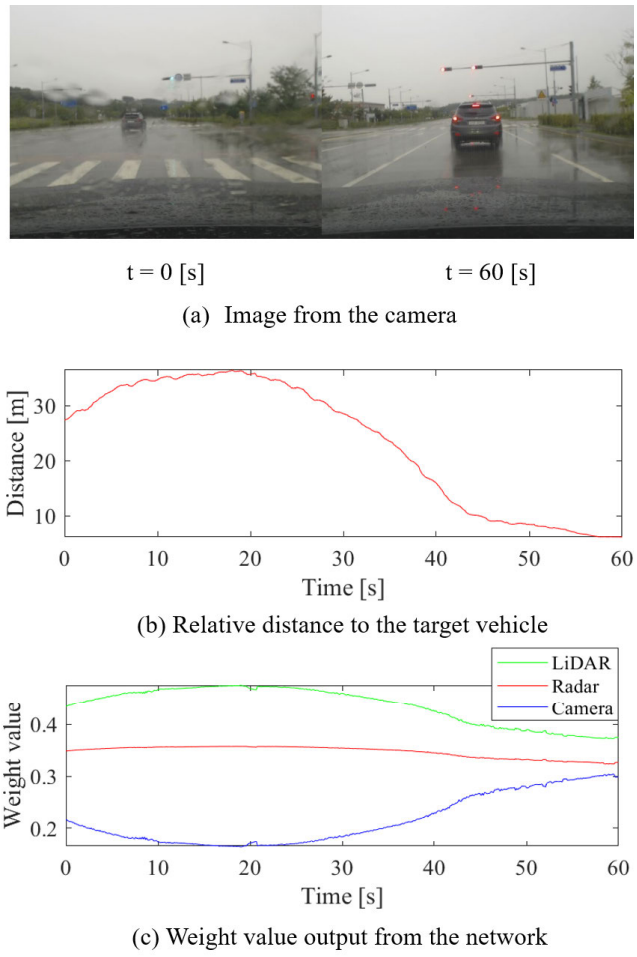

FIGURE 6. Variation in sensor weight value depending on the relative distance to the target vehicle.

TABLE 5. Determined weight values for sensors.

		Radar	LiDAR	Camera
Individual scenario	(a) straight road on a sunny day	0.354	0.460	0.186
	(b) curved road on a rainy day	0.347	0.437	0.216
	(c) curved road on a sunny day	0.347	0.436	0.217
	Overall scenario	0.348	0.438	0.214

with multiple vehicles in front, where the proposed method detects the target vehicle more accurately than the CI method. In particular, the MSE for the CI method is 0.38 m, while the MSE for the proposed approach is 0.3 m. It exhibits a 22% improvement in accuracy. Compared to the presence of a single vehicle in the nearby areas, the performance improvement is slightly smaller. We believe that this is related to a relatively small amount of training data.

Repeated experiments with five tests per scenario in Fig.5 are conducted under the same conditions and the maximum estimation errors are compared in Table 6. It shows that the proposed method reduced the maximum error by around 40%. The results demonstrate that the proposed method, which updates the reliability of each sensor

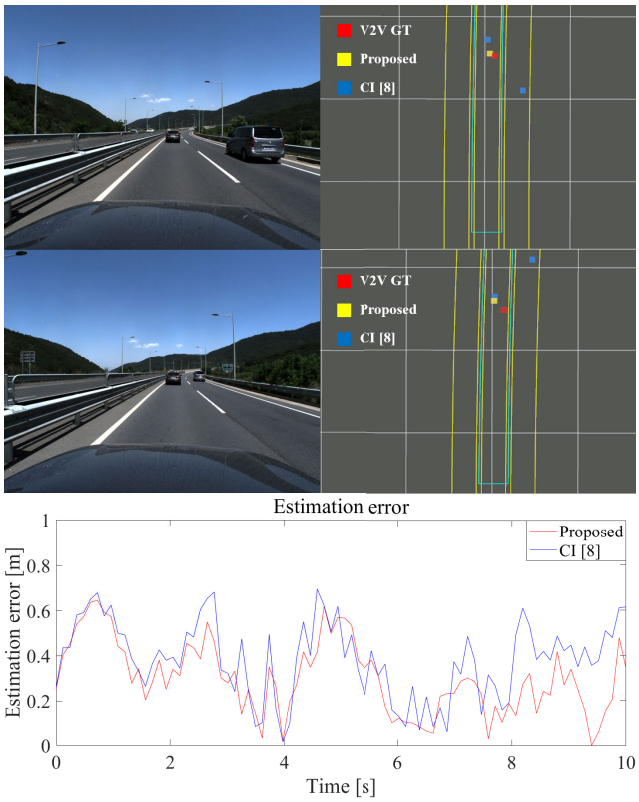

FIGURE 7. Estimation results with multiple vehicles in front.

TABLE 6. Overall detection results: computation time and maximum detection error.

Method	Track-to-track fusion computation time per track [ms]	Maximum estimation error [m]
CI	**0.280**	1.691
Proposed	1.132	**0.977**

based on the surrounding environment information, shows better estimation performance than the method employing the CI. The proposed track-to-track approach exhibited real-time computation capability with less than 10 ms per track in a GPU environment. The computation time of the proposed method takes much longer than the CI method as shown in Table 6, but this pattern is expected to reverse as the number of sensors or the number of IMM models increases.

IV. CONCLUSION AND FUTURE STUDIES

In this paper, a novel sensor fusion algorithm is proposed by utilizing the multimodal learning technique. The position of nearby vehicles is estimated based on the data from LiDAR, radar, and camera sensors. The proposed algorithm combines the IMM filter and the track-to-track fusion method based on multimodal learning. By utilizing the IMM filter, multiple models are suggested to describe the vehicle motion better and are combined to estimate the vehicle position. For track-to-track fusion, to improve the disadvantages of being vulnerable to changes in the surrounding environment, the optimal weight for each sensor is set by using the multimodal learning approach. The weight for each sensor is determined in real-time based on the information such as weather, how the vehicle acts, where the target vehicle is, and the shape of the road. The adaptive track-to-track fusion method is developed to respond to changes in the surrounding environment. The proposed method is verified through real-vehicle experimental data in various driving conditions such as straight or curved roads on a sunny or rainy day. The results demonstrate that the proposed method shows robustness in the sensor fusion accuracy of estimating the positions of surrounding vehicles. Future research will investigate multimodal networks further to consider other changes in driving environment such as slippery road, complex downtown driving, intersection driving, foggy weather, etc.

REFERENCES

[1] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.

[2] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. J. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 825–830.

[3] K. Jo, K. Chu, K. Lee, and M. Sunwoo, "Integration of multiple vehicle models with an IMM filter for vehicle localization," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 746–751.

[4] Y. Xu, W. Zhang, W. Tang, C. Liu, R. Yang, L. He, and Y. Wang, "Estimation of vehicle state based on IMM-AUKF," *Symmetry*, vol. 14, no. 2, p. 222, Jan. 2022.

[5] M. Aeberhard and N. Kaempchen, "High-level sensor data fusion architecture for vehicle surround environment perception," in *Proc. 8th Int. Workshop Intell. Transp.*, vol. 665, 2011, pp. 1–7.

[6] M. Mukherjee, A. Banerjee, A. Papadimitriou, S. S. Mansouri, and G. Nikolakopoulos, "A decentralized sensor fusion scheme for multi sensorial fault resilient pose estimation," *Sensors*, vol. 21, no. 24, p. 8259, Dec. 2021.

[7] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. Amer. Control Conf.*, 1997, pp. 2369–2373.

[8] L. Chen, P. O. Arambel, and R. K. Mehra, "Estimation under unknown correlation: Covariance intersection revisited," *IEEE Trans. Autom. Control*, vol. 47, no. 11, pp. 1879–1882, Nov. 2002.

[9] J. Sijs, M. Lazar, and P. J. V. D. Bosch, "State fusion with unknown correlation: Ellipsoidal intersection," in *Proc. Amer. Control Conf.*, Jun. 2010, pp. 3992–3997.

[10] P. Zhang, S. Zhou, P. Liu, and M. Li, "Distributed ellipsoidal intersection fusion estimation for multi-sensor complex systems," *Sensors*, vol. 22, no. 11, p. 4306, Jun. 2022.

[11] B. Noack, J. Sijs, and U. D. Hanebeck, "Inverse covariance intersection: New insights and properties," in *Proc. 20th Int. Conf. Inf. Fusion*, Jul. 2017, pp. 1–8.

[12] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Decentralized data fusion with inverse covariance intersection," *Automatica*, vol. 79, pp. 35–41, May 2017.

[13] C. Funk, B. Noack, and U. D. Hanebeck, "Conservative quantization of fast covariance intersection," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2020, pp. 68–74.

[14] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proc. 5th Int. Conf. Inf. Fusion*, 2002, pp. 901–904.

[15] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," in *Proc. 7th Int. Conf. Inf. Fusion*, 2005, p. 7.

[16] C. Zhang, Z. Yang, X. He, and L. Deng, "Multimodal intelligence: Representation learning, information fusion, and applications," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 3, pp. 478–493, Mar. 2020.

[17] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, "VATT: Transformers for multimodal self-supervised learning from raw video, audio and text," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24206–24221.
[18] F. R. Valverde, J. Valeria Hurtado, and A. Valada, "There is more than meets the eye: Self-supervised multi-object detection and tracking with sound by distilling multimodal knowledge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11607–11616.
[19] Q. Song, B. Sun, and S. Li, "Multimodal sparse transformer network for audio-visual speech recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 12, 2022, doi: 10.1109/TNNLS.2022.3163771.
[20] J. Nie, J. Yan, H. Yin, L. Ren, and Q. Meng, "A multimodality fusion deep neural network and safety test strategy for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 310–322, Jun. 2021.
[21] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and validation of 3D object position using multimodal sensor fusion and semantic segmentation," *Sensors*, vol. 20, no. 4, p. 1110, Feb. 2020.
[22] T. Lombaerts, K. Kannan, E. Kawamura, C. Dolph, V. Stepanyan, G. E. Gorospe, and C. A. Ippolito, "Distributed ground sensor fusion based object tracking for autonomous advanced air mobility operations," in *Proc. AIAA SCITECH Forum*, Jan. 2023, p. 0896.
[23] J. D. Choi and M. Y. Kim, "A sensor fusion system with thermal infrared camera and LiDAR for autonomous vehicles and deep learning based object detection," *ICT Exp.*, vol. 9, no. 2, pp. 222–227, Apr. 2023.
[24] H. Florea, A. Petrovai, I. Giosan, F. Oniga, R. Varga, and S. Nedevschi, "Enhanced perception for autonomous driving using semantic and geometric data fusion," *Sensors*, vol. 22, no. 13, p. 5061, Jul. 2022.
[25] A. Sengupta, L. Cheng, and S. Cao, "Robust multiobject tracking using mmWave radar-camera sensor fusion," *IEEE Sensors Lett.*, vol. 6, no. 10, pp. 1–4, Oct. 2022.
[26] C. Chui and G. Chen, *Kalman Filtering: With Real-Time Applications* (Springer Series in Information Sciences). Berlin, Germany: Springer, 2008. [Online]. Available: https://books.google.co.kr/books?id=4faHPOyu54C
[27] K. Salahshoor, M. Mosallaei, and M. Bayat, "Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended Kalman filter algorithm," *Measurement*, vol. 41, no. 10, pp. 1059–1076, Dec. 2008.
[28] R. Schubert, C. Adam, M. Obst, N. Mattern, V. Leonhardt, and G. Wanielik, "Empirical evaluation of vehicular models for ego motion estimation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 534–539.
[29] N. L. Baisa, "Derivation of a constant velocity motion model for visual tracking," 2020, *arXiv:2005.00844*.
[30] Z. Deng, P. Zhang, W. Qi, Y. Gao, and J. Liu, "The accuracy comparison of multisensor covariance intersection fuser and three weighting fusers," *Inf. Fusion*, vol. 14, no. 2, pp. 177–185, Apr. 2013.

KYUSANG YOON received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 2021, and the M.S. degree from the Department of Automotive Engineering (Automotive-Computer Convergence), Hanyang University, in 2023. His research interests include object detection and autonomous vehicle control. His current research interests include increment perception performance through sensor fusion and machine learning for autonomous vehicles.

JAEHO CHOI received the B.S. degree in automotive engineering from Kookmin University, Seoul, South Korea, in 2017. He is currently pursuing the Ph.D. degree with the Department of Automotive Engineering (Automotive-Computer Convergence). His research interests include LiDAR data processing and high-level fusion. He has been aiming for increment perception performance through sensor fusion and machine learning for autonomous vehicles.

KUNSOO HUH (Member, IEEE) received the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 1992. He is currently a Professor with the Department of Automotive Engineering, Hanyang University, Seoul, South Korea. His research interests include machine monitoring and control, with an emphasis on their applications to vehicular systems. His current research interests include sensor-based active safety systems, V2X-based safety systems, autonomous vehicle control, and AI applications in autonomous vehicle.