sensors

MDPI

*Article*

# Extended Kalman Filter (EKF) Design for Vehicle Position Tracking Using Reliability Function of Radar and Lidar

**Taeklim Kim [1] and Tae-Hyoung Park [2,*]**

[1] Department of Control and Robot Engineering, Chungbuk National University, Cheongju 28644, Korea; taeglem@cbnu.ac.kr

[2] School of Electronics Engineering, Chungbuk National University, Cheongju 28644, Korea

* Correspondence: taehpark@cbnu.ac.kr

Received: 20 May 2020; Accepted: 21 July 2020; Published: 24 July 2020

**Abstract:** Detection and distance measurement using sensors is not always accurate. Sensor fusion makes up for this shortcoming by reducing inaccuracies. This study, therefore, proposes an extended Kalman filter (EKF) that reflects the distance characteristics of lidar and radar sensors. The sensor characteristics of the lidar and radar over distance were analyzed, and a reliability function was designed to extend the Kalman filter to reflect distance characteristics. The accuracy of position estimation was improved by identifying the sensor errors according to distance. Experiments were conducted using real vehicles, and a comparative experiment was done combining sensor fusion using a fuzzy, adaptive measure noise and Kalman filter. Experimental results showed that the study's method produced accurate distance estimations.

**Keywords:** Kalman filter; sensor fusion; LiDAR; radar

## 1. Introduction

Vehicle position tracking studies are crucial for accurately estimating distances. Position tracking is used in autonomous vehicle research when solving situations using detection alone is difficult. These types of studies use lidar sensors and radar cameras for detection and recognition. Cameras are useful for object recognition, but there are times when measuring distances robustly is difficult; recent sensor fusion studies attempting robust distance measurements have been undertaken [1,2]. Methods utilizing cameras detect a vehicle from an image, and the distance is estimated by defining a proportional expression between the image coordinate system and the actual distance [3–5]. A recent study made use of deep learning to estimate positions; however, this method produced errors when a vehicle passed by on a slope [6].

Camera sensors have difficulties recognizing objects and estimating positions at night, and sensor fusion compensates for these problems by reducing measurement distance errors to improve detection [7–9]. Previous studies on vehicle recognition and distance measurement have used a combination of cameras and radar. However, this study used a camera and a radar to track the position of the vehicle. Combining a camera and radar reduces errors compared to using only a single sensor. In sensor fusion research, numerous studies have been conducted on sensor fusion using lidar and cameras dependent on radar sensors for distance measurement [10–17]. Radar sensors are accurate, but errors are evident when used at a close distance. A lidar sensor is used in tandem to minimize errors. In this study, vehicle detection was carried out using lidar and a camera, while radar and lidar sensors were used for distance measurement.

Previous sensor fusion studies using lidar and radar do not reflect the relationship between sensor characteristics and distance [18,19]. Sensor fusion research utilizes methods like decision trees, fuzzy logic, deep learning, and Kalman filters [20–23]. While the decision tree method cannot represent several situations, fuzzy logic can express various situations with minor difficulties defining membership functions. The deep-learning method shows excellent performance but requires a large amount of computing power. Instead, a Kalman filter was used to track the distance of the target vehicle by combining lidar and radar data.

In this study, a reliability function was designed to reflect the distance characteristics using lidar and radar by analyzing the errors between the sensors according to distance. An extended Kalman filter (EKF) was designed, which confirmed that using the reliability function improved the accuracy of distance estimation.

## 2. Problem Definition

### *2.1. Lidar and Radar Sensor Characteristics According to Target Vehicle Distance*

A lidar sensor displays the reflected object as dots when it hits the object with a laser, and the points measured by the laser produce an unstructured point cloud. Detection through deep learning includes Voxel, MV3D, and Vote3Deep [24–27]. These deep-learning networks display sufficient performance when using a high-channel lidar. A detection algorithm and an image of the frustum was created because this study used a low-channel lidar to detect the target vehicle [28]. Detection using frustum images presents highlights the problems caused by foreground and background obstructions (Figure 1).

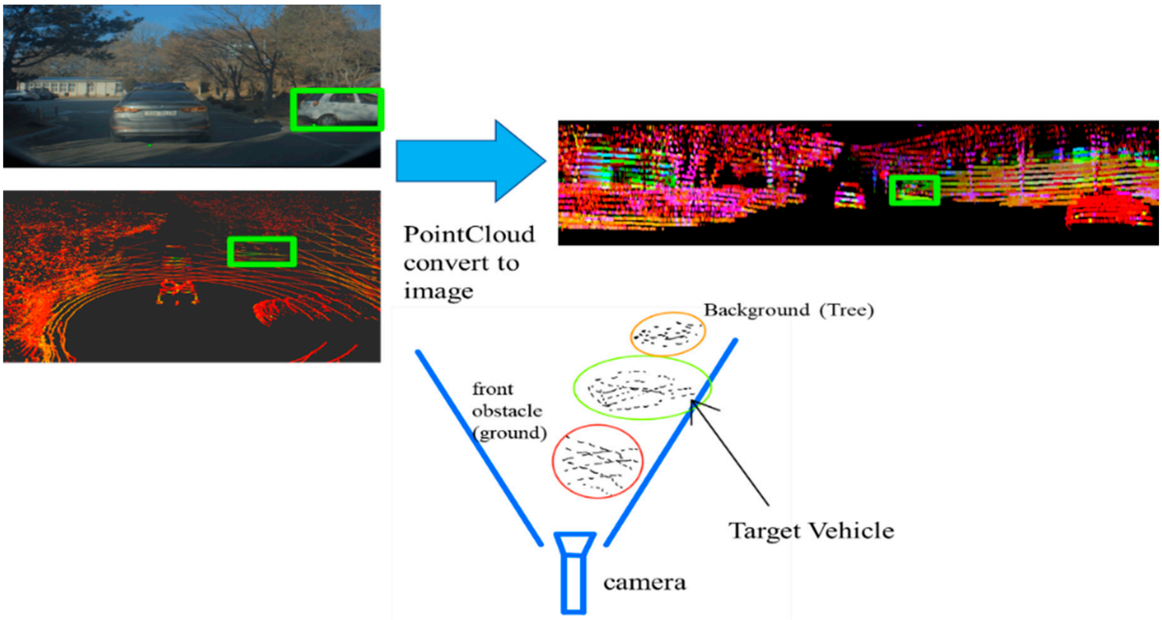


**Figure 1.** Problems with lidar detection using frustums [29]. Left: Object detected using RGB camera. Right: Detection results in frustum images. Bottom: Obstacles to be removed from frustum image: background (tree) and front obstacle (ground) in the picture.

Obstructions are included in the object detection in frustum images. However, a study using an unsupervised learning method, clustering, provided a solution to these problems [29]. We used the following methods to select as many obstacles and object candidates as possible. After changing the point cloud to a top-view image, the threshold removed numerous tree and ground points. Clustering using Euclidean distance constructed the point cloud from candidate vehicles, and conversion to frustum images utilized the candidate's point cloud. If there is an obstacle in front of the vehicle, boxes of a certain size can be removed, but if a box of the vehicle candidate size is present, these boxes cannot be removed properly. A deep-learning network could solve this problem, but the computing power of the present study's equipment was insufficient.

Even if the background and obstacles are completely removed, accurately measuring the distance from the detected region of interest (ROI) must be considered. This distance was measured by removing the obstacle from the detected vehicle area and averaging the values at the ROI's center. However, as the target vehicle moved further away, the measured value produced an error. The lidar data are dependent on various factors, such as the measurement distance and the slope angle of the scanned object [30,31]. A change in the detected vehicle's lidar beam was considered when the distance changed (Figure 2).

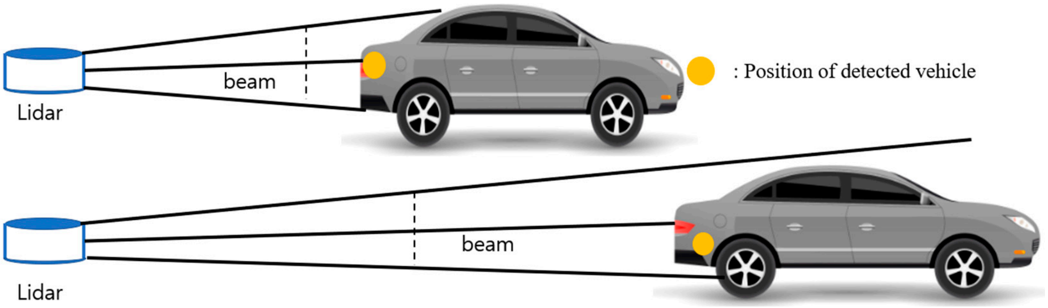


**Figure 2.** Estimated position of the forward vehicle with a lidar sensor.

If the target vehicle is close to the sensor or at a high-channel lidar, a relatively accurate value can be obtained by averaging and measuring the detected ROI. However, vehicles far from low-channel lidars have fewer beams reflected, and measuring the distance with a lidar beam in a small area was necessary. This distance was measured by averaging the lidar data measured with the point cloud of the detected ROI. Figure 2 demonstrates that the greater the distance between the target element and the scanner, the greater the mismeasured elements' effect on the readings, as shown by an error. The lidar can measure closer objects accurately, but the measurement of a distant object is inaccurate.

Figure 3 shows the change in the frustum image's target vehicle based on distance. Target vehicles at close range can be measured relatively accurately if the center area is averaged after detection. However, if the vehicle is further away, the area will be reduced, and the ROI method will produce inaccurate readings. In particular, a low-channel lidar's beam decreases rapidly as the distance increases until it cannot be detected. If the vehicle is detected further away, it will appear as a point-cloud line in the frustum image. The point cloud was fused with a radar sensor to reduce the likelihood of errors occurring at long distances.

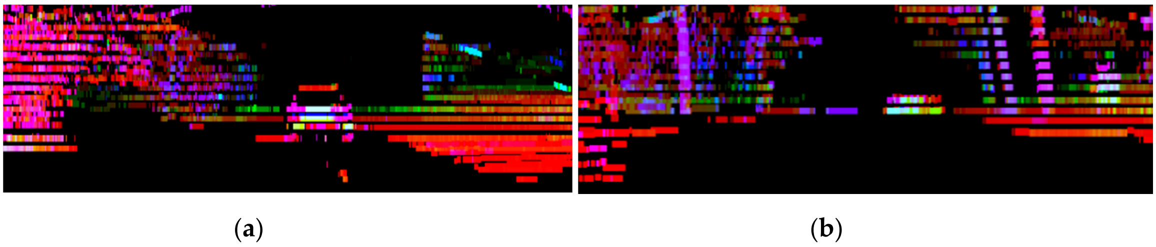


**Figure 3.** Changes in the frustum image according to the distance of the target vehicle: (**a**) close and (**b**) far.

A radar sensor is not dependent on the time of day and can measure distances more robustly than cameras. Thus, the distance was measured using a camera sensor combined with radar, which is affected by the frequency used or the target object's speed [32,33]. A radar sensor was installed on the front bumper of the ego vehicle to measure the vehicle's distance. Although it was measured with an error of 0.2 m, the measurement was inaccurate at a close distance.

The radar confirmed that the lower part of the vehicle was measured from a close distance (Figure 4). Some cases measured radio waves by diffraction from the bottom of the vehicle without returning measurements from the trunk. Distance-specific data were collected to analyze errors and account for the uncertainties of each sensor.

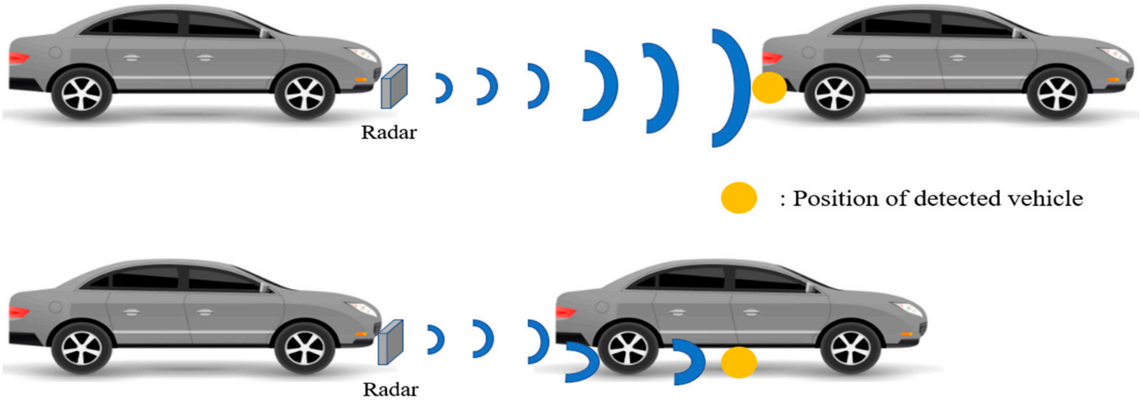


**Figure 4.** Problem in measurement values with the radar sensor.

## 2.2. Data Uncertainty Analysis

The measurement errors of the lidar and radar sensors were calculated and summarized according to distance (Figure 5). Distance errors are based on the sensors' inputs, and ground truth (GT) manually selects the data measured by lidar. After converting the frustum image into lidar, there was a large error in measuring the distance by averaging the point cloud of the detected vehicle's ROI. However, the raw point cloud was chosen as GT because it had an error of less than 5 cm. That is, the data error of the unprocessed lidar was less than 5 cm, and the value measured by the lidar was determined as GT. The error was also calculated using Euclidean distance.

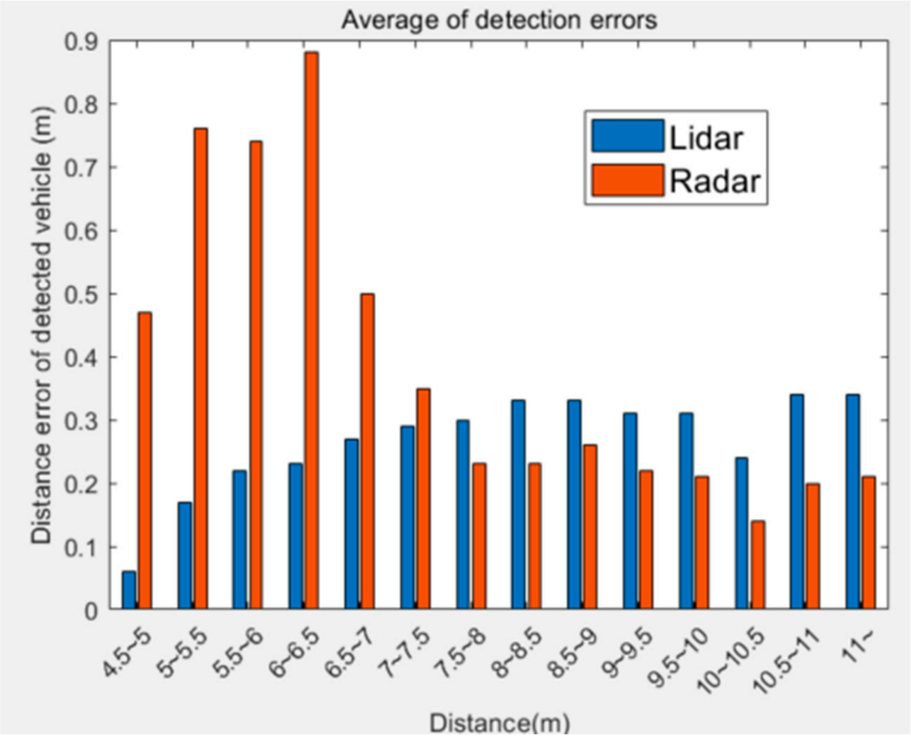


**Figure 5.** Measurement error lidar sensor and radar sensor according to distance.

Figure 5 summarizes the errors caused by the distance between the lidar and radar sensors, which reflects the characteristics of the sensor, the detector, and other various factors. Upon comparing the distance error, the average lidar error was low at closer distances, and the average of the radar data was low at greater distances. In summary, the radar produced more accurate long-distance measurements, while the lidar more was accurate at close range. An extended Kalman filter with a reliability function was designed to reflect the distance characteristics.

## 3. Vehicle Position Tracking

The system structure was divided into detection and tracking processes (Figure 6). In the detection process, a camera and a radar sensor were integrated for vehicle recognition and distance measurement purposes, and camera detection was performed using YOLO v3 [34].

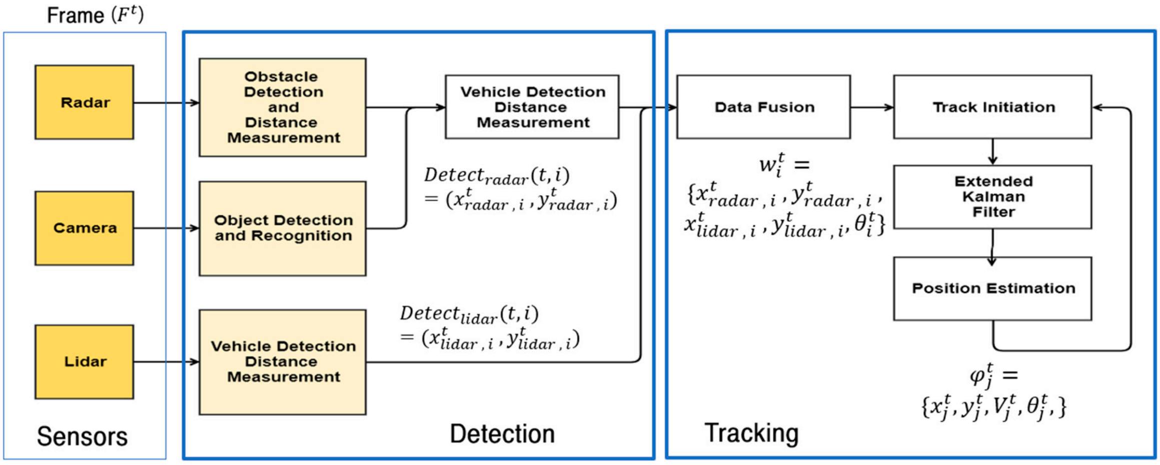

Frame $(F^t)$
Radar
Camera
Lidar
Sensors
Obstacle Detection and Distance Measurement
Object Detection and Recognition
Vehicle Detection Distance Measurement
Vehicle Detection Distance Measurement
$Detect_{radar}(t,i) = (x^t_{radar,i}, y^t_{radar,i})$
$Detect_{lidar}(t,i) = (x^t_{lidar,i}, y^t_{lidar,i})$
Detection
Data Fusion
$w^t_i = \{x^t_{radar,i}, y^t_{radar,i}, x^t_{lidar,i}, y^t_{lidar,i}, \theta^t_i\}$
Track Initiation
Extended Kalman Filter
Position Estimation
$\varphi^t_j = \{x^t_j, y^t_j, V^t_j, \theta^t_j,\}$
Tracking

**Figure 6.** System structure.

The result of the integration of camera and radar sensors is expressed using the formula $Detect_{radar}(t,i) = \left(x^t_{radar,\ i}\ y^t_{radar,i}\right)$ for I $\in \{1\ldots n\}$, where n is the number of detected objects obtained at time $t$. Lidar detection was performed using convolutional neural networks (CNNs) based on polar view [28]. The detection result of lidar is expressed as follows, using $Detect_{lidar}(t,i) = \left(x^t_{lidar\ ,i}, y^t_{lidar\ ,i}\right)$ for I $\in \{1\ldots n\}$, where n is the number of detected objects obtained at time $t$. $w^t_i = \left\{x^t_{lidar,\ i}, y^t_{lidar,i}, x^t_{radar,i}, y^t_{radar,i}, \theta^t_i\right\}$ is the detection result. Input $w^t_i(t,i)$ was used to predict the next state, and $\theta$ is the heading of the vehicle.

Recent studies have shown the possibility of obtaining a vehicle's heading using a 3D bounding box [35,36]. However, the heading often did not appear when applied correctly to a low-channel lidar. The heading of the vehicle was calculated using the direction accumulated over five frames, the amount of change in each direction was obtained, and $\theta$ was calculated using Equation (1) [37].

$$\begin{gathered}
\Delta x_{lidar} = \frac{\sum_{a=2}^{5}\left[(a-1)\left(x^{t-5+a}_{lidar} - x^{t-5+a}_{lidar}\right)\right]}{\sum_{a=2}^{5}(a-1)},\ \Delta y_{lidar} = \frac{\sum_{a=2}^{5}\left[(a-1)\left(x^{t-5+a}_{lidar} - x^{t-5+a}_{lidar}\right)\right]}{\sum_{a=2}^{5}(a-1)}, \\
\Delta x_{radar} = \frac{\sum_{a=2}^{5}\left[(a-1)\left(x^{t-5+a}_{lidar} - x^{t-5+a}_{lidar}\right)\right]}{\sum_{a=2}^{5}(a-1)}, \Delta y_{radar} = \frac{\sum_{a=2}^{5}\left[(a-1)\left(x^{t-5+a}_{lidar} - x^{t-5+a}_{lidar}\right)\right]}{\sum_{a=2}^{5}(a-1)}, \\
\Delta x_{total} = \frac{\Delta x_{lidar} + \Delta x_{radar}}{2},\ \Delta y_{total} = \frac{\Delta y_{lidar} + \Delta y_{radar}}{2}, \\
\theta^t_i = \mathrm{ATAN2}(\Delta y_{total}, \Delta x_{total}),
\end{gathered} \tag{1}$$

The extended Kalman filter was used to estimate the position and speed of the target vehicle.

### *3.1. Extended Kalman Filter Design*

The state variable of the Kalman filter is the $j$th tracking vehicle result at time $t$ (Equation (2)).

$$\varphi_j^t = \left\{x_j^t, y_j^t, V_j^t, \theta_j^t\right\}, \tag{2}$$

where $x$ is the front distance from the ego vehicle to the target vehicle, $y$ is the lateral distance from the ego vehicle to the target vehicle, $V$ is the velocity, and $\theta$ is the vehicle‘s heading. The Kalman filter‘s measurement vector is the position and direction of the target vehicle. The position and direction are measured through each detector, which determines the measurement vector and gives the position of the $i$th detection vehicle at time $t$ (Equation (3)).

$$w_i^t = \left\{x_{lidar,\ i}^t, y_{lidar,i}^t, x_{radar,i}^t, y_{radar,i}^t, \theta_i^t\right\}, \tag{3}$$

The process model of the state vector is expressed as Equation (4).

$$a_j^{t-1} = \begin{bmatrix} x_j^{t-1} + \Delta t V_j^{t-1}\cos\left(\theta_j^{t-1}\right) \\ x_j^{t-1} + \Delta t V_j^{t-1}\sin\left(\theta_j^{t-1}\right) \\ V_j^{t-1} \\ \theta_j^{t-1} \end{bmatrix}, \tag{4}$$

Equation (4) is nonlinear, and was linearized by obtaining the Jacobian matrix in Equation (5).

$$A_j^{t-1} = \begin{bmatrix} 1 & 0 & \Delta t cos\left(\theta_j^{t-1}\right) & 0 \\ 0 & 1 & \Delta t sin\left(\theta_j^{t-1}\right) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

Next, $H_j$ was defined as a relational expression between the measurement and state vectors. The updated measurement vector was used to build a model that predicted the next state based on the previous state, and the measurement vectors coming from each sensor had different noises depending on the distance. As such, Equation (6) was added to the reliability function.

$$w_i^t = \begin{bmatrix} x_{lidar\ ,i}^t \\ y_{lidar\ ,i}^t \\ x_{radar\ ,i}^t \\ y_{radar\ ,i}^t \\ \theta_{lidar\ ,i}^t \end{bmatrix}^T = \begin{bmatrix} x_j^{t-1} + \Delta \mathrm{t} V_j^{t-1}\cos\left(\theta_j^{t-1}\right) + sig_{\alpha_1\beta_1}(x) \\ y_j^{t-1} + \Delta \mathrm{t} V_j^{t-1}\sin\left(\theta_j^{t-1}\right) + sig_{\alpha_1\beta_1}(y) \\ x_j^{t-1} + \Delta \mathrm{t} V_j^{t-1}\cos\left(\theta_j^{t-1}\right) + sig_{-\alpha_2\beta_2}(x) \\ y_j^{t-1} + \Delta \mathrm{t} V_j^{t-1}\sin\left(\theta_j^{t-1}\right) + sig_{-\alpha_2\beta_2}(y) \\ \theta_j^{t-1} \end{bmatrix} \tag{6}$$

For the lidar sensor, the closer the vehicle was to the front, the more reliable the measured value became, while the opposite was true for the radar sensor. In Equation (6), $\mathrm{sig}(x)$ illustrates the reliability function that reflected measurement errors over a distance. The addition of the reliability function made the measurement transformation matrix nonlinear. An extended Kalman filter resolved the equation’s nonlinearity; the next matrix equation was linearized using the Jacobian matrix in

Equation (7). The Kalman gain was updated by comparing the predicted value with the measured value using $H_j^t$.

$$H_j^t = \frac{\partial h\left(w_i^t\right)}{\partial w_i^t} = \begin{bmatrix} 1-(\alpha_2\beta_2)*sig'_{-\alpha_2\beta_2}\left(x^t_{radar,i}\right) & 0 & \Delta tcos\left(\theta_j^{t-1}\right) & 0 \\ 0 & 1+(\alpha_1\beta_1)*sig'_{\alpha_1\beta_1}\left(y^t_{lidar,i}\right) & \Delta tsin\left(\theta_j^{t-1}\right) & 0 \\ 1-(\alpha_2\beta_2)*sig'_{-\alpha_2\beta_2}\left(x^t_{radar,i}\right) & 0 & \Delta tcos\left(\theta_j^{t-1}\right) & 0 \\ 0 & 1-(\alpha_2\beta_2)*sig'_{-\alpha_2\beta_2}\left(y^t_{radar,i}\right) & \Delta tsin\left(\theta_j^{t-1}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

### 3.2. *Reliability Function*

The position estimation of the vehicle using the Kalman filter predicted the next state using the value measured by the sensor. Numerous factors caused differences between the predicted position and the actual position, such as erroneous measurements. As explained earlier, this study focused on minimizing distance errors by designing a reliability function to reflect the distance characteristics of the sensors in the Kalman filter. In the longitudinal direction, the reliability of the lidar and radar sensors' values was measured based on distance (Figure 7).

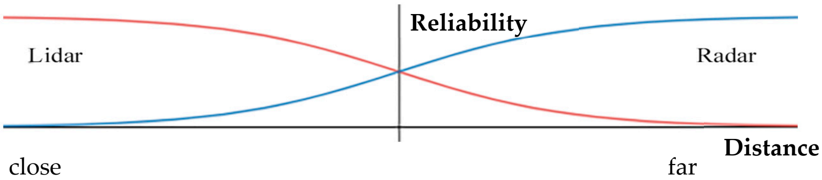


**Figure 7.** Reliability function; the x-axis is the distance of the opponent vehicle from the ego vehicle. The y-axis represents the reliability of the value measured by the sensor. When comparing the two sensors, the lidar was accurate at close range, and the radar was accurate at long distances.

The sigmoid function was used to implement the reliability function with Equation (6) by differentiating the matrix, $H_j$, which represented the relationship between the state vector and the measurement vector. Figure 8 shows the differential of the sigmoid function.

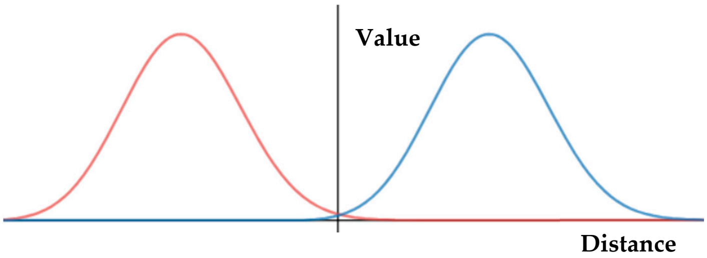


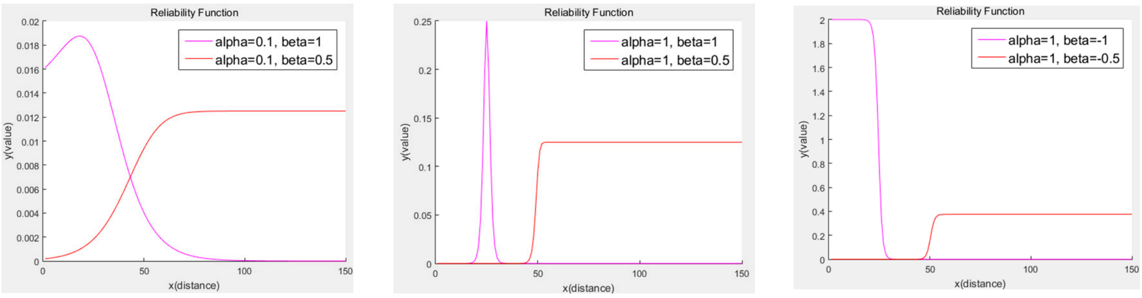
**Figure 8.** Differentiated reliability function; the x-axis is the distance of the opponent vehicle from the ego vehicle. The y-axis represents the reliability of the value.

The differential sigmoid function had the same distribution as the Gaussian function, which reflected the measurement vector's errors. Meanwhile, the reliability function was expressed as Equation (8). In general, a Kalman filter analyzes the errors of each sensor and reflects measurement noise using a Gaussian function. Measurement noise should analyze and identify as many errors as possible, but this is not easily performed. A direct substitution into the measurement transformation matrix aimed to reflect this error.

$$\begin{aligned} sig_{\alpha_1\beta_1}(\mathrm{x}) &= \frac{\beta_1}{1+e^{\alpha_1(x-X_{lidar,\ \ reli})}}, \\ sig_{-\alpha_2\beta_2}(\mathrm{x}) &= \frac{\beta_2}{1+e^{-\alpha_2(x-X_{radar,\ reli})}}, \\ sig_{\alpha_1\beta_1}(y) &= \frac{\beta_1}{1+e^{\alpha_1(y-Y_{lidar,\ \ reli})}}, \\ sig_{-\alpha_2\beta_2}(y) &= \frac{\beta_2}{1+e^{-\alpha_2(y-Y_{radar,\ reli})}}, \end{aligned} \tag{8}$$

As shown in Figure 5, the sensors' measurement errors did not exactly switch between one another. The distance characteristics of each sensor did not cross the middle value. Therefore, the sections where the sensor's error and reliability function's value were at the maxima had to match, which changed the reliability function of each sensor with parameters $X_{lidar,reli}$ and $X_{radar,reli}$. The uncertainty of the measured value increased from half the maximum detection distance. Each variable corresponded with a point where the reliability of a sensor was halved. This variable was expressed as the probability of a more reliable sensor among the two; $\alpha_1$, $\alpha_2$, $\beta_1$, $\beta_2$ were determined through experiments.

Figure 9 shows the differential result of the reliability function for each parameter change. The x-axis is the distance value measured by the sensor, and the y-axis is the output value of each parameter when input is added to the reliability function. The uncertainty of each sensor, according to the measured value, is reflected in the Kalman filter. $H_j^{t-1}$ changes adaptively and updates the Kalman filter using the reliability function.



(**a**) (**b**) (**c**)

**Figure 9.** Differentiation result of the reliability function according to parameter change: when $X_{lidar,reli} = 25$, $X_{radar,reli} = 50$, (**a**) $\alpha = 0.1, \beta = 1$ and $\alpha = 0.1, \beta = 0.5$; (**b**) $\alpha = 1, \beta = 1$ and $\alpha = 1, \beta = 0.5$; (**c**) $\alpha = 1, \beta = -1$ and $\alpha = 1, \beta = -0.5$.

### 3.3. Kalman Filter Update

The state vector $\varphi_j^t$ was estimated using the state transition matrix $\mathrm{A}_j^{t-1}$ and expressed as Equation (9).

$$\varphi_j^- = \mathrm{A}_j^{t-1}\varphi_j^{t-1} \tag{9}$$

Next, the error covariance $P_j^t$ was obtained.

$$P_j^t = \mathrm{A}_j^{t-1}P_j^{t-1}\mathrm{A}_j^{t-1^T} + Q, \tag{10}$$

In Equation (11), $Q$ is defined as system noise. The Kalman gain was obtained using the updated error covariance $P_j^t$ and the measurement transition matrix $H_j^{t-1}$.

$$K_j^t = P_j^t H_j^{t-1^T}\left[H_j^{t-1}P_j^t H_j^{t-1^T} + R_j\right]^{-1}, \tag{11}$$

For Equation (12), $R_j$ is the measurement noise. The Kalman gain, the measured sensor value, and the predicted value were used to obtain $\varphi_j^t$, thus estimating the next value of the state vector.

$$\varphi_j^t = \varphi_j^{t-1} + K_j^t\left[w_j^t - H_j^{t-1}\varphi_j^t\right], \tag{12}$$

In Equation (13), updates of the error covariance were obtained using the updated Kalman gain.

$$P_j^t = \left[I - K_j^t H_j^t\right]P_j^{t-1}, \tag{13}$$

## 3.4. Tracking Management

Tracking management (Figure 10) was designed for experiments with multiple vehicles. The experimental section describes the use of real vehicles during the experiments, while multiple vehicles were used in a simulation. In the latter case, the detection and tracking results should match. The following is an explanation of the detection and tracking results.

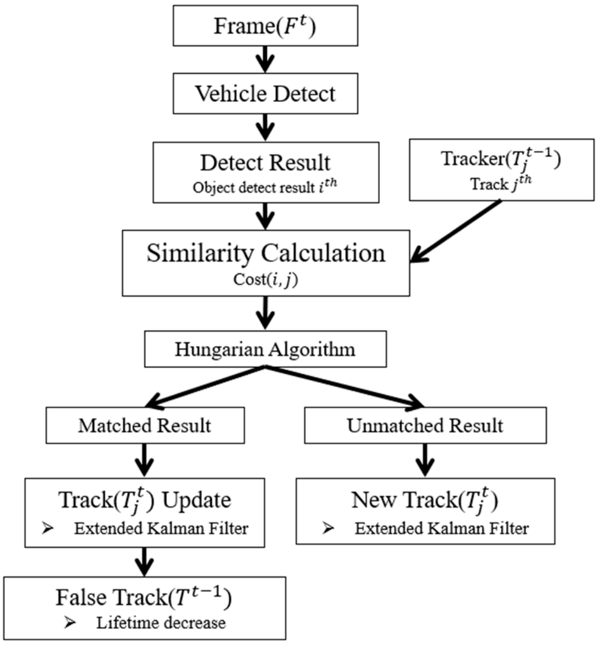


**Figure 10.** Tracking management.

The detection results of the lidar and radar obtained at the time $t$ were used as the inputs. When the detected vehicle was called $Detect_i(x, y)$, the contents of the detected vehicle were expressed as Equation (14). $Detect_i(x, y)$ includes the x and y positions and $\theta$. Even if only one sensor obtained a detection result, the vehicle tracker was updated. $\theta$ was measured in the manner previously described. If only the lidar or radar data were measured, $\theta$ was obtained using only one sensor value.

$$Detect_i(x, y, \theta) = \begin{cases} x^i_{lidar},\ y^t_{lidar,\, i},\ (only\ lidar) \\ x^i_{radar},\ y^t_{radar,\, i},\ (only\ radar) \\ \theta^t_i, \end{cases} \tag{14}$$

The detection result used as input calculated the similarity to the $T_j^{t-1}$ obtained by the tracker Equation (15).

$$T_j^{t-1}(x, y, V, \theta) = \begin{cases} \varphi_j^{t-1} = \left\{x_j^{t-1}, y_j^{t-1}, V_j^{t-1}, \theta_j^{t-1}\right\} \\ Lifetime = k,\ k < Maxlife \end{cases}, \tag{15}$$

The similarity was calculated using two matching pieces of information (Equation (16)), and the similarity calculation used Euclidean distance.

$$cost\left(Detect_i, T_j^{t-1}\right) = \sqrt[2]{\left(Detect_i(x) - T_j^{t-1}(x)\right)^2 + \left(Detect_i(y) - T_j^{t-1}(y)\right)^2} \tag{16}$$

Data association used a Hungarian algorithm, frequently used for allocation problems where cost must be lower than the threshold to allocate [38]. For example, if the threshold is 1 and the cost is 2, it is not assigned. However, if the cost is greater than the threshold, it is considered irrelevant. Thus, all detectors calculated the cost of each existing tracker and matched them accordingly. The unassigned result proceeded as follows. If the detection result was not matched with the existing tracker using the algorithm, a new tracker was created. The matched tracker updated the Kalman filter and increased the tracker's lifetime; unassigned trackers had reduced lifetimes.

## 4. Experiment

Figure 11 shows the vehicle used in the experiment, which was equipped with three Velodyne VLP-16 lidar sensors. Attached to the front of the vehicle was a Delphi ESR 2.5 radar sensor, and the coordinate system was calibrated based on the left lidar sensor. The lidar data used the detected data for the vehicles after converting the point cloud to the spherical coordinate system. Experiments were then performed using the fuzzy method and the proposed method to reflect the sensor's distance characteristics accurately. For system noise and measurement noise, a Kalman filter accurately estimated the process's variables [39].

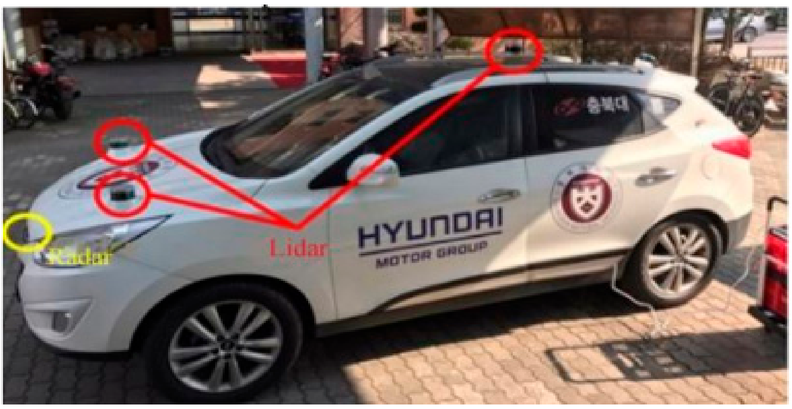


**Figure 11.** Test vehicle. It was equipped with three lidars and one radar.

Table 1 shows the noise settings of each filter used. When the measured value was affected by the same value, the experiment was conducted by setting the same parameters as follows to confirm the switching effect. However, in the case of adaptive noise, the experiment was conducted by multiplying the coefficients and changing them according to the switching situation.

**Table 1.** Noise covariances of the proposed algorithm.

| **Q (Process model noise)** | EKF (Proposed) Fuzzy Adaptive | diag (0.5, 0.05, 0.05, 0.05) |
|---|---|---|
| **R (Measurement noise)** | EKF (Proposed)<br>Fuzzy<br>Adaptive | diag (0.05, 0.05, 0.05, 0.05, 0.05)<br>diag (0.05, 0.05, 0.05)<br>α∗diag (0.05, 0.05, 0.05) |

### 4.1. Fuzzy Rule

The fuzzy algorithm sets fuzzy rules and membership functions that represent various situations. This study used sensor fusion while considering distance characteristics. Fuzzy rules were based on this idea and are shown in Table 2.

**Table 2.** Fuzzy rules.

| Input | | Output |
|---|---|---|
| **Lidar** | **Radar** | **Output Weight** |
| Close | Close<br>Middle<br>Far | Lidar<br>Lidar<br>Both |
| Middle | Close<br>Middle<br>Far | Lidar<br>Both<br>Radar |
| Far | Close<br>Middle<br>Far | Both<br>Radar<br>Radar |

When the target vehicle was close, the lidar measurement was more reliable than radar; at greater distances, radar was more reliable than lidar. This relationship served as the basis of the fuzzy rules. The output $\mathrm{f}_o$. $x_m$ was the median of the maximum distances both sensors could detect. If it was below $x_m$, the output value was determined by the weight of the lidar. However, if detected data from lidar were more than or equal to $x_m$, it was used as the radar's weight Equation (17). Figure 12 is a membership function of the fuzzy algorithm. The fuzzy membership function was used as a reference, and the parameters were changed during the experiment [40].

$$\begin{gathered} \mathrm{f}_o * x_{lidar,i} + (1-\mathrm{f}_o) * x_{radar,i} = x_f \left( if\ x_{lidar\ ,i} < x_m \right), \\ \mathrm{f}_o * y_{lidar\ ,i} + (1-\mathrm{f}_o) * y_{radar\ ,i} = y_f, \\ (1-\mathrm{f}_o) * x_{lidar\ ,i} + \mathrm{f}_o * x_{radar,i} = x_f \left( if\ x_{lidar\ ,i} \geq\ x_m \right), \\ (1-\mathrm{f}_o) * y_{lidar\ ,i} + \mathrm{f}_o * y_{radar\ ,i} = y_f, \end{gathered} \tag{17}$$

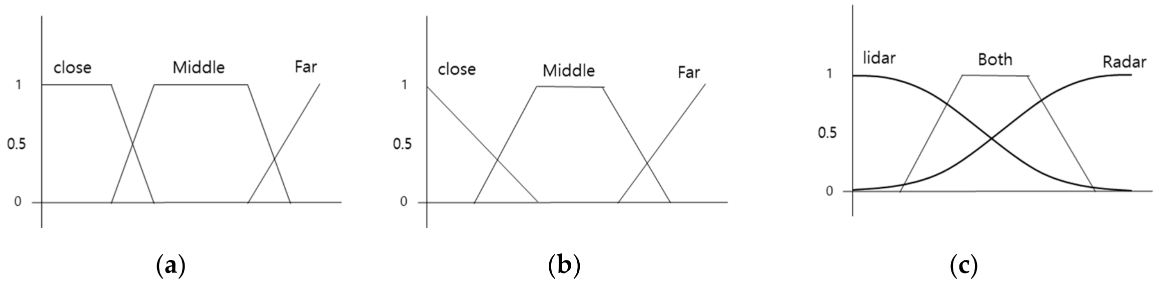


**Figure 12.** Membership function: (**a**) lidar membership function; (**b**) radar membership function; (**c**) output membership function.

### 4.2. In Reality for a Single Vehicle

Object tracking was performed using an experimental vehicle that compared the sensor fusion of the camera and radar and the fusion of the camera lidar sensor. The distance between the camera and the radar was calibrated by using the coordinates of the pillars in the image to obtain a matrix that converted the value of the image pixel into a radar distance coordinate system [41]. The camera computed the projective transformation matrix between the reference object's x and y coordinates and the distance measured by the radar. Figure 13 shows the image of calculation of the projection transformation matrix.

**Figure 13.** Test vehicle equipped with three lidars, one radar, and a dashboard camera.

The measurements used a thin object to maintain the accuracy of the camera and radar's projection transformation matrix. For these thin objects, the matrix was computed using the image's x and y ground coordinates, and the radar detection results were projected onto the image coordinate system using the calculated matrix (Figure 14).

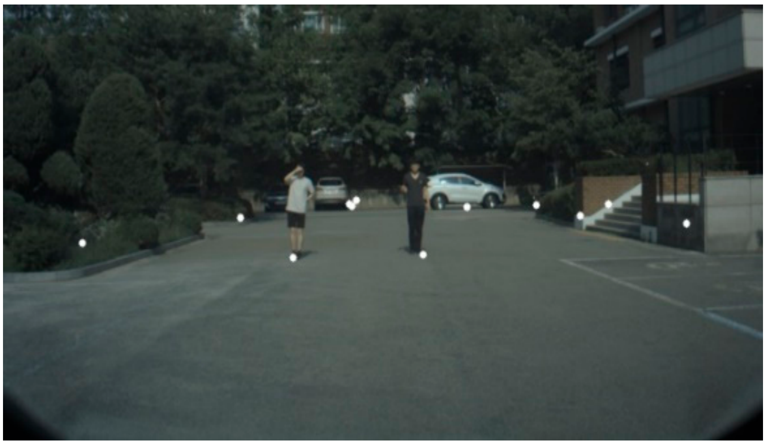


**Figure 14.** The result of projecting the radar detection result in the image coordinate system.

The perspective transformation matrix allowed the measurement of distances by comparing the results detected in the image with the radar detection results, which were also obtained using the above method. The result of the radar camera sensor fusion and lidar camera sensor fusion was expressed as Equation (18), and the algorithm used in the experiment was a Kalman filter. Said equation is equal to Equation (18).

$$\begin{gathered}\varphi_j^t = \left\{x_j^t, y_j^t, V_j^t, \theta_j^t\right\} \\ w_i^t = \left\{x_{lidar,\ i}^t, y_{lidar,i}^t, \theta^t{}_i\right\} \text{ (In case of sensor fusion of lidar and camera)} \\ w_i^t = \left\{x_{radar,i}^t, y_{radar,i}^t, \theta^t{}_i\right\} \text{ (In case of sensor fusion of radar and camera)}\end{gathered} \tag{18}$$

$$A_j^{t-1} = \begin{bmatrix} 1 & 0 & \Delta t cos\left(\theta_j^{t-1}\right) & -\Delta t V_{x,j}^{t-1} \sin\left(\theta_j^{t-1}\right) \\ 0 & 1 & \Delta t sin\left(\theta_j^{t-1}\right) & \Delta t V_{x,j}^{t-1} \cos\left(\theta_j^{t-1}\right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

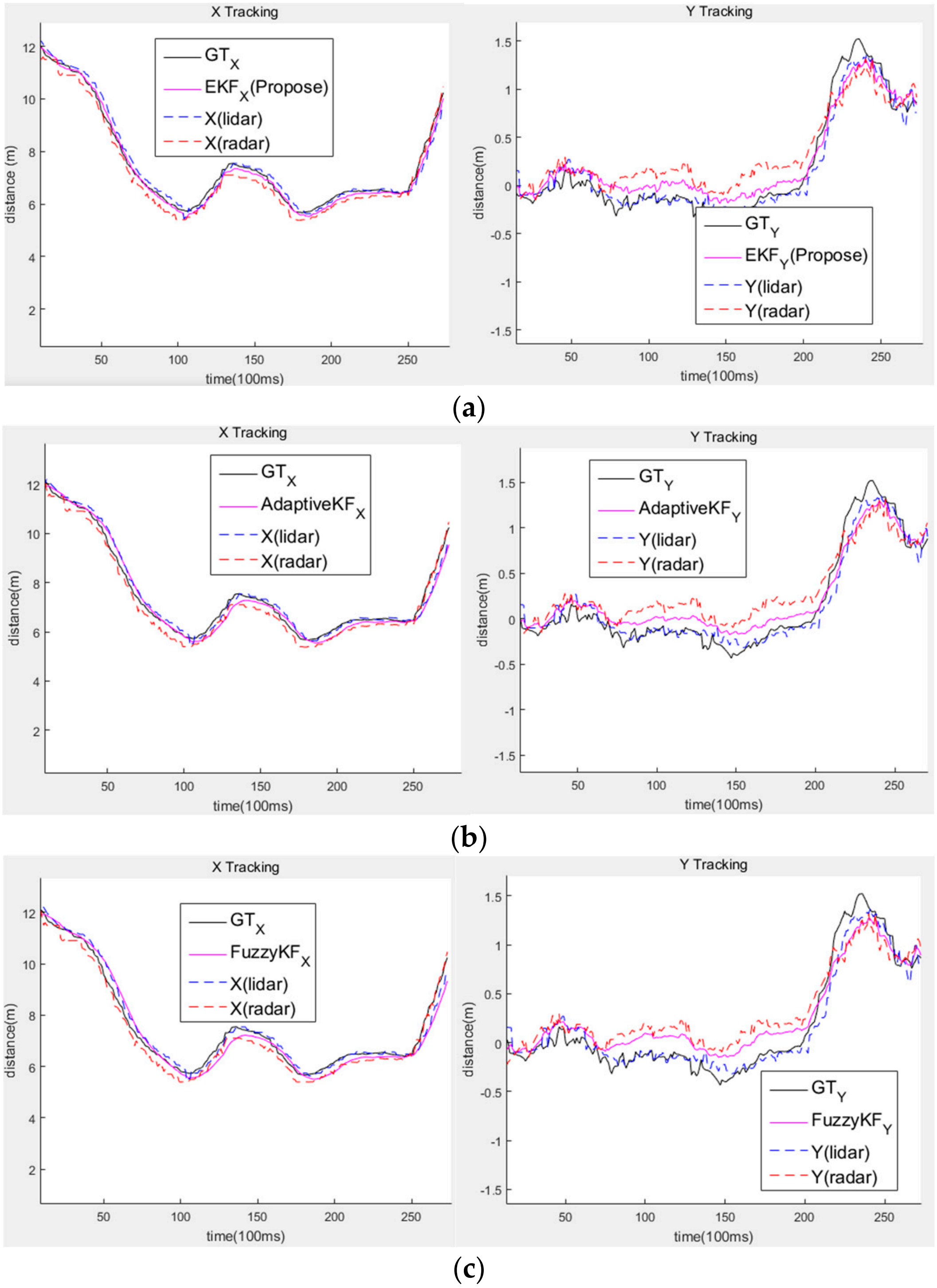
$$H_j^{t-1} = \frac{\partial h\left(w_i^t\right)}{\partial w_i^t} = \begin{bmatrix} 1 + \Delta \mathrm{t} V_j^{t-1} \cos\left(\theta_j^{t-1}\right) & 0 & -\Delta t V_{x,j}^{t-1} \sin\left(\theta_j^{t-1}\right) \\ 0 & 1 + \Delta \mathrm{t} V_j^{t-1} \sin\left(\theta_j^{t-1}\right) & \Delta t V_{x,j}^{t-1} \cos\left(\theta_j^{t-1}\right) \\ 0 & 0 & 1 \end{bmatrix},$$

The data used in the study were obtained using the experimental vehicle. A total of 17 data scenarios were generated by combining left and right turns and approaching or moving away from the ego vehicle. GT was determined manually by viewing the lidar data, and the distance accuracy was evaluated by calculating the GT and the root-mean-squared error (RMSE) of the tracking results. Table 3 shows the results of comparing the distances between different sensor fusions. As a comparative experiment, fuzzy and adaptive measurement noises were used, and the results confirmed that the proposed method was more accurate at estimating positions by showing the errors according to the distance of the sensor.

**Table 3.** Comparison of the results of the proposed method with other sensor fusion methods.

| Method RMSE (Frame) | Proposed Method | | Adaptive Measure Noise | | Fuzzy | |
|---|---|---|---|---|---|---|
| | Lidar/Radar/Camera | | Lidar/Radar/Camera | | Lidar/Radar/Camera | |
| | x (m) | y (m) | x (m) | y (m) | x (m) | y (m) |
| Far (681) | 0.29 | 0.30 | 0.41 | 0.38 | 0.28 | 0.34 |
| Close (547) | 0.32 | 0.22 | 0.33 | 0.22 | 0.32 | 0.29 |
| Left turn (152) | 0.55 | 0.35 | 0.54 | 0.36 | 0.61 | 0.33 |
| Right turn (210) | 0.87 | 0.31 | 0.91 | 0.31 | 1.07 | 0.31 |
| Left curve (210) | 0.28 | 0.20 | 0.41 | 0.24 | 0.46 | 0.25 |
| Right curve (72) | 0.39 | 0.25 | 0.47 | 0.30 | 0.25 | 0.22 |
| Total (1882) | 0.38 | 0.27 | 0.50 | 0.33 | 0.40 | 0.30 |

Figure 15 also shows that the method proposed in this study reflected more accurate distance measurements and position estimation results. The distance characteristics of the sensor made accurately measuring distance characteristics challenging, and changing the measurement noise made estimating accurate distances difficult because the error covariance update was constantly changing. Therefore, the method proposed in this study was suitable for reflecting distance characteristics.



(**a**)

(**b**)

(**c**)

**Figure 15.** Comparison experiment result of other sensor fusion methods: (**a**) proposed method; (**b**) adaptive measure noise and Kalman filter; (**c**) fuzzy method and Kalman filter.

### *4.3. In Simulation for Multiple Vehicles*

As testing multiple vehicles in real-world environments was difficult, experiments were conducted using simulations. The data were generated using a Prescan simulator with multiple vehicles in three different scenarios (Figure 16). Sensor fusion and comparison experiments were carried out using the fuzzy algorithm.

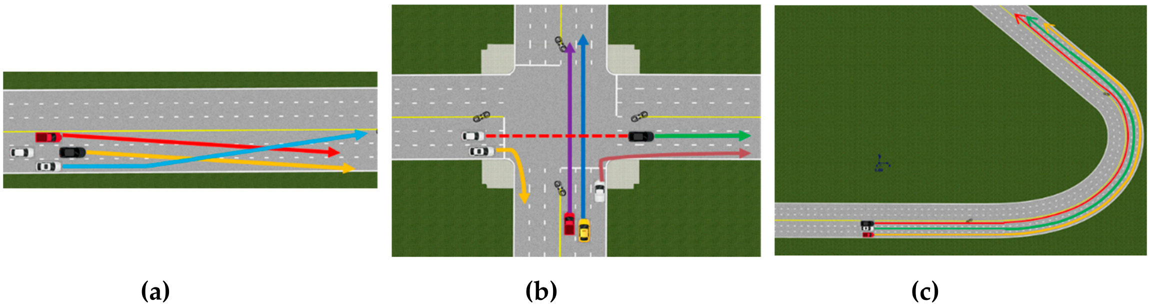


**Figure 16.** Simulation of three scenarios: (**a**) vehicles changing lanes; (**b**) intersection scenario; (**c**) curve scenario.

The first scenario showed several vehicles changing lanes; the proposed method was similar to GT (Table 4). The multiple object tracking precision (MOTP) calculated and checked the precision of the estimated tracked position [42]. Using the calculated results, the method proposed in the first scenario reflected accurate position estimations.

**Table 4.** Comparison of the results of the proposed method with fuzzy method in Scenario 1.

| Method MOTP (Frame) | Proposed Method | | Fuzzy Method | |
|---|---|---|---|---|
| | Lidar/Radar/Camera | | Lidar/Radar/Camera | |
| | x (m) | y (m) | x (m) | y (m) |
| Vehicle 1 (112) | 1.12 | 1.04 | 1.19 | 1.02 |
| Vehicle 2 (65) | 1.08 | 0.63 | 1.25 | 0.68 |
| Vehicle 3 (128) | 0.93 | 0.67 | 1.04 | 0.66 |
| Total | 1.01 | 0.65 | 1.20 | 0.67 |

The next scenario illustrated a vehicle passing an intersection. In this case, the vehicle's lateral movement was tracked. Table 5 shows the results in Scenario2. For the y-axis movement, the proposed method was imprecise. However, the proposed method provided an estimate close to the GT along the x-axis.

**Table 5.** Comparison of the results of the proposed method with fuzzy method in Scenario 2.

| Method MOTP (Frame) | Proposed Method | | Fuzzy Method | |
|---|---|---|---|---|
| | Lidar/Radar/Camera | | Lidar/Radar/Camera | |
| | x (m) | y (m) | x (m) | y (m) |
| Vehicle 1 (60) | 1.07 | 0.64 | 1.36 | 0.67 |
| Vehicle 2 (67) | 1.12 | 0.91 | 1.17 | 0.89 |
| Vehicle 3 (40) | 1.07 | 0.53 | 1.27 | 0.51 |
| Total | 1.09 | 0.72 | 1.26 | 0.72 |

The final scenario showed the vehicle moving closer to and further away from a curve. As it moved closer, then further away, the data obtained by the sensors were inaccurate. When the vehicle moved away from or close to the ego vehicle, the values measured by the lidar or radar sensors were imprecise. In this scenario, the simulation data were generated, and the experimental results showed that the proposed method was more precise than the sensor fusion using fuzzy results. The results of the experiment are summarized in Table 6. Finally, calculating the MOTP by reflecting each situation frame by frame showed that the proposed method reduced errors by 0.22 m. The results of all scenarios are shown in Table 7.

The above results confirmed that the distance estimation along the x-axis was more accurate. Although the simulation and actual vehicle test results were similar, the y-axis method was correct, but the performance only showed a small improvement.

**Table 6.** Comparison of the results of the proposed method with fuzzy method in Scenario 3.

| Method MOTP (Frame) | Proposed Method | | Fuzzy Method | |
|---|---|---|---|---|
| | Lidar/Radar/Camera | | Lidar/Radar/Camera | |
| | x (m) | y (m) | x (m) | y (m) |
| Vehicle 1 (155) | 1.12 | 0.35 | 1.42 | 0.32 |
| Vehicle 2 (137) | 0.93 | 0.33 | 1.20 | 0.31 |
| Total | 1.03 | 0.72 | 1.32 | 0.72 |

**Table 7.** Comparison of the results of the proposed method with fuzzy method in all scenarios.

| Method MOTP (Frame) | Proposed Method | | Fuzzy Method | |
|---|---|---|---|---|
| | Lidar/Radar/Camera | | Lidar/Radar/Camera | |
| | x (m) | y (m) | x (m) | y (m) |
| Total | 1.04 | 0.69 | 1.26 | 0.70 |

## 5. Conclusions

In this study, a fused sensor combined the characteristics of lidar and radar sensors according to distance. A point cloud, measured with lidar, provided a top-view image that removed ground and obstacle points before being converted into a frustum image. Meanwhile, the radar was fused with the camera for detection and distance measurement. The lidar obtained accurate measurements when objects were close to it, and the radar sensor was accurate when measuring distant objects. An extended Kalman filter was constructed to reflect the characteristics of each sensor while obtaining measurements. A combination of the two sensors was created using a Kalman filter, which was designed as an extended filter that reflected distance characteristics by adding a reliability function. The experiment utilized an actual vehicle to evaluate the method's performance. While estimating the distance by identifying the characteristics of the sensor, the system noise of the Kalman filter was compared with a fuzzy method or the proposed method. The study confirmed that the accuracy of distance measurements was improved as a result of the lidar and radar sensor fusion, and the method that reflected distance errors was more accurate in the extended Kalman filter's composition.

**Author Contributions:** Conceptualization, T.K., and T.-H.P.; methodology, T.K.; project administration, T.-H.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was financially supported by the Ministry of Small and Medium-sized Enterprises(SMEs) and Startups(MSS), Korea, under the "Regional Specialized Industry Development Program(R&D, S2874252)" supervised by the Korea Institute for Advancement of Technology(KIAT).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Song, H.; Choi, W.; Kim, H. Robust Vision-based Relative-Localization Approach using an RGB-Depth Camera and LiDAR Sensor Fusion, Trans. *Ind. Electron.* **2016**, *63*, 3726–3736. [CrossRef]
2. Kang, D.; Kum, D. Camera and Radar Sensor Fusion for Robust Vehicle Localization via Vehicle Part Localization. *IEEE Access* **2020**, *8*, 75223–75236. [CrossRef]
3. Zaarane, A.; Slimani, I.; Hamdoun, A.; Atouf, I. Vehicle to vehicle distance measurement for self-driving systems. In Proceedings of the International Conference on Control, Decision and Information Technologies, Paris, France, 23–26 April 2019.
4. Kim, G.; Cho, J.S. Vision-based Vehicle Detection and Inter-Vehicle Distance Estimation. In Proceedings of the International Conference on Control, Automation and Systems, Jeju, Korea, 17–21 October 2012.
5. Liu, Z.; Lu, D.; Qian, W.; Ren, K.; Zhang, J.; Xu, L. Vision-based inter-vehicle distance estimation for driver alarm system. *IET Intell. Transp. Syst.* **2019**, *13*, 927–932. [CrossRef]

6. Huang, L.; Zhe, T.; We, J.; Pei, C.; Chen, D. Robust Inter-Vehicle Distance Estimation Method based on Monocular Vision. *IEEE Access* **2019**, *7*, 46059–46070. [CrossRef]
7. Chen, B.; Pei, X.; Chen, Z. Research on Target Detection based on Distributed Track Fusion for Intelligent Vehicles. *Sensors* **2020**, *20*, 56. [CrossRef]
8. Wang, X.; Xu, L.; Sun, H.; Xin, J.; Zheng, N. On-road Vehicle Detection and Tracking using MMW Radar and Monovision Fusion. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2075–2084. [CrossRef]
9. Hsu, Y.W.; Lai, Y.H.; Zhong, K.W.; Yin, T.K.; Perng, J.W. Developing an on-road Object Detection System using Monovision and Radar Fusion. *Energies* **2020**, *13*, 116. [CrossRef]
10. Rangesh, A.; Trivedi, M.M. No blind spots: Full-Surround Multi-Object Tracking for Autonomous Vehicles using Cameras and LiDARs. *IEEE Trans. Intell Transp. Syst.* **2019**, *14*, 588–599. [CrossRef]
11. Ruiz, C.D.; Wang, Y.; Chao, W.L.; Weinberger, K.; Campbell, M. Vision-only 3D tracking for self-driving cars. In Proceedings of the 15th International Conference on Automation Science and Engineering, Vancouver, BC, Canada, 22–26 August 2019.
12. Omar, R.; Garcia, C.; Aycard, O. Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 525–534.
13. Fang, Y.; Zhao, H.; Zha, H.; Zhao, X.; Yao, W. Camera and LiDAR fusion for on-road vehicle tracking with reinforcement learning. In Proceedings of the IEEE Intelligent Vehicles Symposium, Paris, France, 9–12 June 2019.
14. Shan, M.; Alvis, C.D.; Worrall, S.; Nebot, E. Extended vehicle tracking with probabilistic spatial relation projection and consideration of shape feature uncertainties. In Proceedings of the IEEE Intelligent Vehicles Symposium, Paris, France, 9–12 June 2019.
15. De Silva, V.; Roche, J.; Kondoz, A. Robust Fusion of LiDAR and Wide-Angle Camera Data for Autonomous Mobile Robots. *Sensors* **2018**, *18*, 2730. [CrossRef]
16. Koci?, J.; Jovi?i?, N.; Drndarevi?, V. Sensors and sensor fusion in autonomous vehicles. In Proceedings of the 26th Telecommunications Forum TELFOR, Belgrade, Serbia, 20–21 November 2018.
17. Homm, F.; Kaempchen, N.; Ota, J.; Burschka, D. Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection. In Proceedings of the IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010.
18. Han, J.; Kim, J.; Son, N.S. Persistent automatic tracking of multiple surface vessels by fusing radar and lidar. In Proceedings of the IEEE Oceans Aberdeen, Aberdeen, UK, 19–22 June 2017.
19. Blanc, C.; Trassoudaine, L.; Gallice, J. EKF and particle filter track to track fusion a quantitative comparison from radar lidar obstacle tracks. In Proceedings of the IEEE International Conference on Information Fusion, Philadelphia, PA, USA, 25–28 July 2005.
20. Yang, Y.; Que, Y.; Huang, S.; Lin, P. Multimodal sensor medical image fusion based on type-2 fuzzy logic in NSCT domain. *IEEE Sens. J.* **2016**, *16*, 3735–3745. [CrossRef]
21. Jo, K.; Lee, M.; Sunwoo, M. Road slope aided vehicle position estimation system based on sensor fusion of GPS and automotive onboard sensors. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 250–263. [CrossRef]
22. Zhao, G.; Xiao, X.; Yuan, J. Fusion of Velodyne and camera data for scene parsing. In Proceedings of the IEEE International Conference of information Fusion, Singapore, 9–12 July 2012.
23. Gao, H.; Cheng, B.; Wang, J.; Li, K.; Zhao, J.; Li, D. Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE T Ind Inform* **2018**, *14*, 4224–4231. [CrossRef]
24. Zhou, Y.; Tuzel, O. VoxelNet: End-to-end learning for point cloud based 3D object detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
25. Engelcke, M.; Rao, D.; Wang, D.Z.; Tong, C.H.; Posner, I. Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. In Proceedings of the Conference on Robotics and Automation, Marina Bay Sands, Singapore, 29 May–3 June 2017.
26. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D object detection network for autonomous driving. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
27. Behl, A.; Paschalidou, D.; Andreas, S.D.; Geiger, A. PointFlowNet: Learning representations for rigid motion estimation from point clouds. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.

28. Kwon, S.S.; Park, T.H. Polar-View based Object Detection Algorithm using 3D Low-Channel Lidar. *J. Inst. Control* **2019**, *25*, 55–62.
29. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum PointNets for 3D object detection from RGB-D data. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018.
30. Wang, J.; Xu, K. Shape Detection from Raw Lidar Data with Subspace Modeling. *IEEE T. Vis. Comput. Gr.* **2016**, *23*, 2137–2150. [CrossRef]
31. Wehr, A.; Lohr, U. Airborne Laser Scanning—an Introduction and Overview. *ISPRS J. Photogramm.* **1999**, *54*, 68–82. [CrossRef]
32. Felguera-Martín, D.; González-Partida, J.T.; Almorox-González, P.; Burgos-García, M. Vehicular Traffic Surveillance and Road Lane Detection using Radar Interferometry. *IEEE T. Veh. Technol.* **2012**, *61*, 959–970. [CrossRef]
33. Li, Y.; Peng, Z.; Pal, R.; Li, C. Potential Active Shooter Detection Based on Radar Micro-Doppler and Range-Doppler Analysis using Artificial Neural Network. *IEEE Sens. J.* **2018**, *19*, 1052–1063. [CrossRef]
34. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
35. Shin, K.; Kwon, Y.; Tomizuka, M. Roarnet. A robust 3D object detection based on region approximation refinement. *arXiv* **2018**, arXiv:1811.03818.
36. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3D object detection. *arXiv* **2019**, arXiv:1912.13192.
37. Karunasekera, H.; Wang, H.; Zhang, H. Multiple Object Tracking with Attention to Appearance, Structure, Motion and Size. *IEEE Access* **2019**, *7*, 104423–104434. [CrossRef]
38. Bae, S.H.; Yoon, K.J. Robust Online Multiobject Tracking with Data Association and Track Management. *IEEE T. Image Process.* **2014**, *23*, 2820–2833.
39. Simon, D. Kalman Filtering. Embedded. Available online: https://www.embedded.com/kalman-filtering/ (accessed on 16 July 2020).
40. Kim, T.L.; Lee, J.S.; Park, T.H. Fusing lidar, radar, and camera using extended Kalman filter for estimating the forward position of vehicles. In Proceedings of the International Conference on Cybernetics and Intelligent Systems (CIS) and Conference on Robotics, Automation and Mechatronics (RAM), Bangkok, Thailand, 18–20 November 2019.
41. Sugimoto, S.; Tateda, H.; Takahashi, H.; Okutomi, M. Obstacle detection using millimeter-wave radar and its visualization on image sequence. In Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, 26–26 August 2004.
42. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Eurasip. J. Image Vide* **2008**, 246309. [CrossRef]