


Article

Fast Online Coordinate Correction of a Multi-Sensor for Object Identification in Autonomous Vehicles

Wooyoung Lee ¹, Minchul Lee ², Myoungcho Sunwoo ² and Kichun Jo ^{3,*}

¹ Autonomous Driving Platform Team, Hyundai Motor Company, Seoul 06797, Korea; ericlee0829@hyundai.com

² Department of Automotive Engineering, Hanyang University, Seoul 04763, Korea; minchul.baker.lee@gmail.com (M.L.); msunwoo@hanyang.ac.kr (M.S.)

³ Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, Korea

* Correspondence: kichun.jo@konkuk.ac.kr; Tel.: +82-2-2049-6265

Received: 14 March 2019; Accepted: 26 April 2019; Published: 29 April 2019



Abstract: Multi-sensor perception systems may have mismatched coordinates between each sensor even if the sensor coordinates are converted to a common coordinate. This discrepancy can be due to the sensor noise, deformation of the sensor mount, and other factors. These mismatched coordinates can seriously affect the estimation of a distant object's position and this error can result in problems with object identification. To overcome these problems, numerous coordinate correction methods have been studied to minimize coordinate mismatching, such as off-line sensor error modeling and real-time error estimation methods. The first approach, off-line sensor error modeling, cannot cope with the occurrence of a mismatched coordinate in real-time. The second approach, using real-time error estimation methods, has high computational complexity due to the singular value decomposition. Therefore, we present a fast online coordinate correction method based on a reduced sensor position error model with dominant parameters and estimate the parameters by using rapid math operations. By applying the fast coordinate correction method, we can reduce the computational effort within the necessary tolerance of the estimation error. By experiments, the computational effort was improved by up to 99.7% compared to the previous study, and regarding the object's radar the identification problems were improved by 94.8%. We conclude that the proposed method provides sufficient correcting performance for autonomous driving applications when the multi-sensor coordinates are mismatched.

Keywords: multi-sensor coordinate matching; data association; object identification; online parameter estimation; multi-sensor object convergence; autonomous vehicle

1. Introduction

In autonomous driving systems, one of the most important factors to plan and control a vehicle's motion is the understanding of the autonomous vehicle's environment, especially when moving objects approach it. To obtain this information, various perception sensors such as the camera [1–3], lidar [4–6], and radar [7,8] have been developed. However, single sensor-based perception systems have performance constraints for recognizing surrounding moving objects because of sensor limitations. For example, the camera can make a better estimation of the lateral position and classification of the moving object than radar and lidar can. On the contrary, the radar and lidar can make a better estimation of the longitudinal distance of a moving object whilst have a longer detection range than the camera. To combine the advantages of each sensor, multiple sensor-based perception systems are widely used in autonomous driving systems [9–14].

To obtain the moving object's information from the multi-sensor, converting the coordinates of each sensor into a common coordinate must be preceded, as each sensor has a different installation position, type of coordinate, and measuring units. In this paper, we define the coordinate conversion process as rotating and translating the sensor coordinate for matching into a common coordinate based on origin, orientation difference between sensors and a common coordinate. Even if the coordinate conversion process is completed, the sensor coordinate may slightly mismatch, due to the sensor's noise effect, the sensor's mount deformation due to aging, and the dynamic movements of the vehicle. The slightly mismatched coordinate causes the position error of moving objects from each sensor. Moreover, as a target object moves further, the position error becomes larger. For example, if the azimuth orientation does not match 1-degree, a 0.17 m lateral position error can occur for an object 10 m away, however, a 1.7 m lateral position error occurs for an object 100 m away, as shown in Figure 1. In other words, even if multi-sensor detects the same distant object, it is difficult to determine as the same object due to lateral position error due to the coordinate mismatch. We name this an object identification problem.

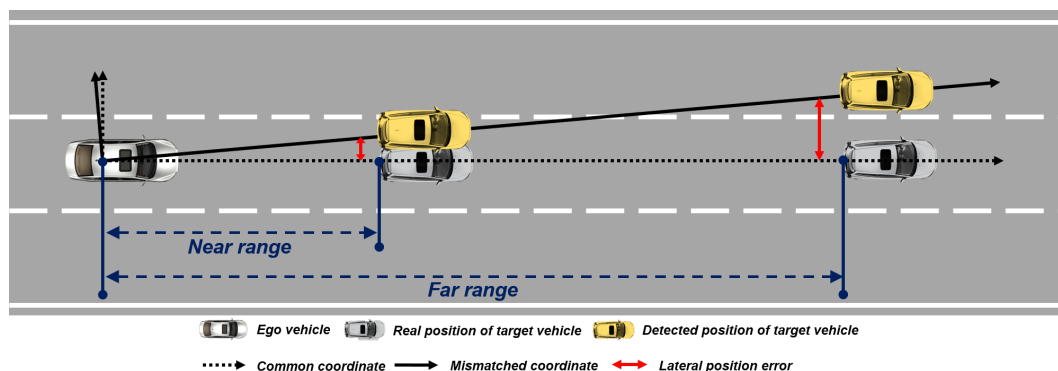


Figure 1. The serious effect of the mismatched coordinate to the distant object.

The object identification problem can cause malfunction of the autonomous driving system, such as collision avoidance and lane change applications, because the system can possibly determine that many objects exist in the frontal area and there is no place to avoid or move, as shown in Figure 2a. Thus, obtaining reliable moving object information by correcting the mismatched coordinate is very important to the autonomous driving system, as shown in Figure 2b. In addition, since the sensor is installed in the sensor mount, coordinate misalignment can occur in real-time by deformation on the sensor mount with the time-lapse and light collision. Furthermore, variation in the pitch and the roll of the vehicle arising from the dynamic driving environment can occur in the slight coordinate mismatching between sensors in real-time. Therefore, mismatched coordinate between multi-sensor must be corrected in real-time to obtain the reliable moving object information from multiple sensors.

There are many approaches concerning coordinate misalignment. One of the approaches uses off-line sensor error modeling based on RTK-GPS [15,16]. The off-line sensor error modeling attempts to solve the misaligned coordinate by modeling the position error of sensor measurements and RTK-GPS information. This method has the advantage to obtain accurate reference positions from RTK-GPS. Another advantage is that the corrected object information could be obtained without an additional computation effort to estimate the model parameters since the model parameters are already estimated off-line. However, this method requires an expensive equipment, the RTK-GPS, to estimate the object information error and could not cope with the coordinate mismatch occurring in real-time. Another approach is minimizing the position error between sensor measurements based on the error estimation methods, such as iterative closest point (ICP), least square method (LSM), amongst other methods [17–21]. One of the previous studies in this approach uses lidar, which has a position accuracy of less than 0.1m, as the reference sensor and correcting the other sensors with ICP algorithm. This method has an ability and advantage to handle the online coordinate mismatching problem. Although

this method is effective, this method requires a high computational effort to calculate the singular value decomposition (SVD), which, is used in ICP. Moreover, additional methods are needed to reduce the sensor noise effect.

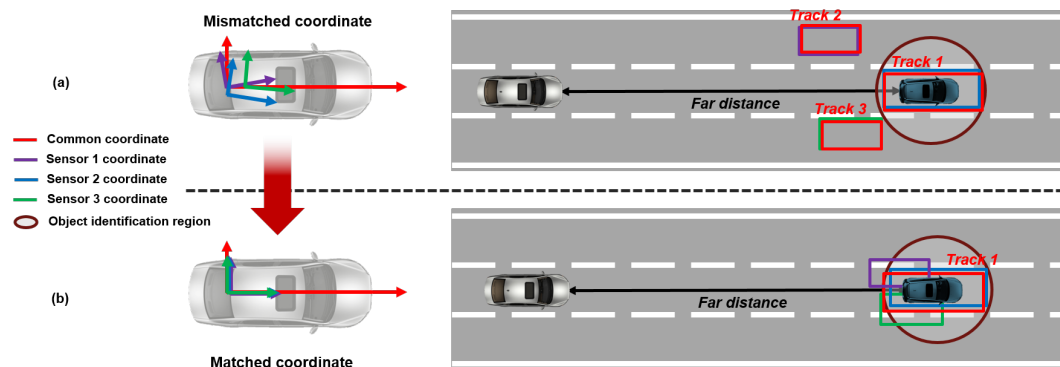


Figure 2. The concept of coordinate correction to solve the object identification problem. (a) The object identification problem caused by coordinate mismatching. (b) The object identified by coordinate correction.

Therefore, we propose a fast online coordinate correction method with the reduced computational effort for matching the mismatched sensor coordinates. The proposed method simplifies a sensor position error model and model parameter estimation based on using rapid mathematical operations. In this paper, the rapid mathematical operations represent the arithmetic operations and inverse trigonometric functions which take a small computational burden. The sensor position error model is designed only with the key factors of position error to reduce the model's complexity within the necessary tolerance of model accuracy. The model parameter estimation method is used as a projection approach with recursive least square to reduce the computation complexity and noise effect within necessary tolerance of correction error.

The fast online coordinate correction method is explained in the following four sections. Section 2 describes the design for sensor position error model. This section contains the dominant parameters selection method based on performance impact analysis of coordinate mismatching factors. In Section 3, the projection approach with recursive least squares is proposed for online model parameter estimation and this section explains the synergy effects of using projection approach and recursive least squares. The proposed method is verified through vehicle experiments in Section 4. Finally, the advantages of the fast online coordinate correction method are summarized in Section 5.

2. Sensor Position Error Model

The coordinate conversion process is translating and rotating the sensor coordinates to match the common coordinate based on the errors of six degrees-of-freedom (DOF) parameters. Since sensors are installed in the mount, the six DOF parameters can be defined as the coordinate origin position (x_o, y_o, z_o) of the sensor mount and the orientation ($\theta_{pitch}, \theta_{roll}, \theta_{yaw}$) of coordinate of the sensor mount. This setup means that the sensor position error model should be designed with errors of six DOF parameters. However, the sensor position error model does not need to consider all six DOF parameters because the object identification problem is closely related to the longitudinal, and lateral positional errors between measurements. In other words, the parameters, which do not affect the longitudinal and lateral position of a moving object, are not necessary to consider in the model's design. Thus, the proposed sensor position error model was designed with only the dominant parameters which influence the longitudinal and lateral position changes of moving objects among the six DOF parameters. Moreover, simplifying the sensor position error model has the ability to reduce the model's complexity.

The dominant parameters were selected through performance impact analysis, which determines the parameters that have a large influence on the longitudinal and lateral position of the moving object, by applying the offset to six DOF parameters. In the performance impact analysis, a 1 m origin offsets were given to the x , y , and z -axis, and 1-degree orientation offsets were given to the θ_{pitch} , θ_{roll} , and θ_{yaw} , respectively. Equations for identifying the longitudinal and lateral direction position changes of a moving object by applying an offset to the six DOF parameters are as follows [22]:

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} x_{offset} \\ y_{offset} \\ z_{offset} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} \cos\theta_{pitch} & 0 & \sin\theta_{pitch} \\ 0 & 1 & 0 \\ -\sin\theta_{pitch} & 0 & \cos\theta_{pitch} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{roll} & -\sin\theta_{roll} \\ 0 & \sin\theta_{roll} & \cos\theta_{roll} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} \cos\theta_{yaw} & \sin\theta_{yaw} & 0 \\ -\sin\theta_{yaw} & \cos\theta_{yaw} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \quad (4)$$

In the above equation, $X_m[x_m, y_m, z_m]$ is the position information changed according to the offset given to the six DOF parameters. Also, the other parameters, $X_o[x_o, y_o, z_o]$ is the object's position obtained from the sensor, $X_{offset}[x_{offset}, y_{offset}, z_{offset}]$ is the translation of the object in three perpendicular directions along the axis, and where $\theta_{offset}[\theta_{pitch}, \theta_{roll}, \theta_{yaw}]$ is the perpendicular rotation of the object along the axis.

The results of the misaligned sensor performance analysis through the above equations are shown in Figure 3. In the case of the axis offsets, the x and y -axis offsets affect to the longitudinal or lateral position. In contrast to x , y -offsets, the z -offset didn't affect to the longitudinal and lateral position, as shown in the above part of Figure 3. These axis offsets could be represented by the translation matrix and (1). If the coordinate rotates 1-degree with respect to x -axis (the roll angle), roll angle offset affected 0.02 m to lateral position. The rotation matrix for roll angle can be represented by (3). When the coordinate rotates 1-degree based on y -axis (the pitch angle), it changed the 0.03 m to longitudinal position. The rotation matrix for pitch angle can be represented by (2). In the case of rotating 1-degree of z -axis (the yaw angle), the yaw angle was affected by up to 4 m to lateral position at 200 m away. The rotation matrix for yaw angle can be represented by (4). Excluding the z -axis offset, the offsets given to the parameters ($x, y, \theta_{pitch}, \theta_{roll}, \theta_{yaw}$) affected the longitudinal and lateral position of the moving object acquired from the sensor. The effects of the offsets given to x, y , and θ_{yaw} on the longitudinal and lateral position of the moving object were relatively larger than the effect of the offsets given to the θ_{pitch} and θ_{roll} . Therefore, the dominant parameters that had a large effect on the longitudinal and lateral position error of the moving object among the six degrees-of-freedom parameters were the x and y -axis origin position error and θ_{yaw} orientation error.

The sensor position error model should be designed to reflect the dominant parameters x and y -axis origin position error and θ_{yaw} error. The x and y -axis origin position errors could be represented by the translation of three perpendicular axes, and the θ_{yaw} error can be represented by the rotation of the z -axis. Therefore, the sensor position error model could be summarized as the rotation matrix and translation matrix as shown in (5).

$$\begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} = \begin{bmatrix} \cos\theta_{err} & -\sin\theta_{err} \\ \sin\theta_{err} & \cos\theta_{err} \end{bmatrix} \begin{bmatrix} x_{meas} \\ y_{meas} \end{bmatrix} + \begin{bmatrix} x_{err} \\ y_{err} \end{bmatrix} \quad (5)$$

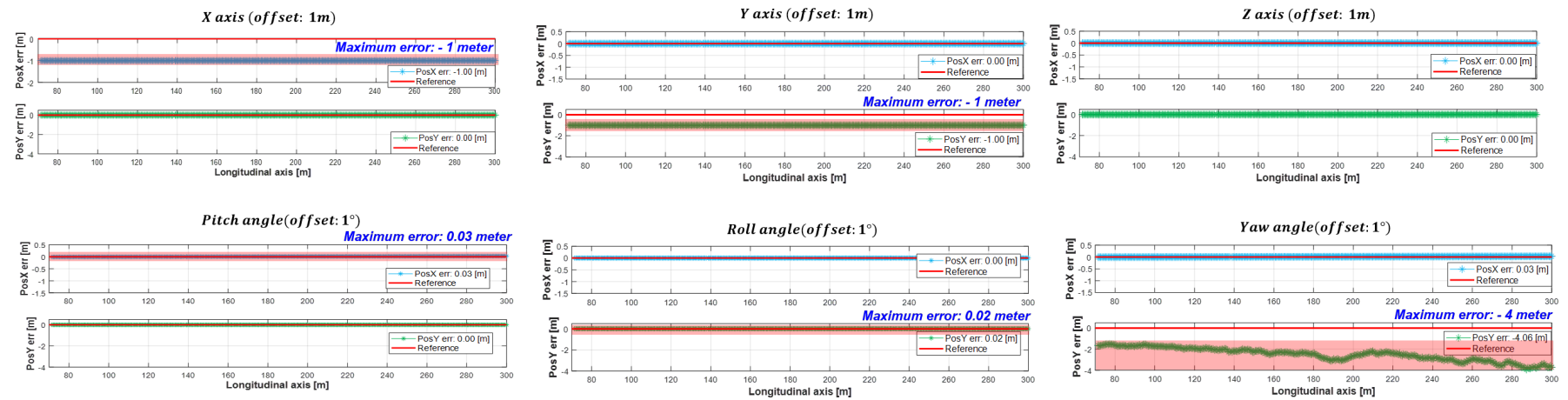


Figure 3. The results of the dominant factors selection by performance impact analysis.

In the above equation, $X_{ref}[x_{ref}, y_{ref}]$ is the reference position on the moving object, $X_{meas}[x_{meas}, y_{meas}]$ is the object position obtained from the sensor, $X_{err}[x_{err}, y_{err}]$ is the origin position error in the longitudinal and lateral, and θ_{err} indicates the θ_{yaw} orientation error.

3. Online Estimation of Model Parameters

As described in the introduction, parameter estimation methods such as ICP, LSM, and other methods use pseudo inverse operation to estimate the parameters of under-determined systems. The pseudo inverse matrix can be obtained by calculating the singular vector matrix with the modified singular value matrix which is inverse and transposed. Singular value decomposition(SVD) is widely used to obtain the singular value matrix and singular vector matrix from the matrix because SVD can decompose the matrix into a singular value matrix and singular vector matrix, even if the matrix is not a square matrix [23]. However, estimation methods using SVD have the property of increasing the accuracy of estimating parameters as the amount of data increases, whilst increasing the computational complexity. These methods correct the mismatched coordinates by collecting the data for a certain period of time to guarantee the reliability of estimated parameters. Thus, these methods are difficult in terms of solving coordinate mismatching problems due to the change of the position of the sensor mount, according to the vehicle dynamics occurring in real-time and require large computational resources. To overcome this problem, we proposed the projection approach with recursive least squares method, that estimates the model parameters for each time step.

3.1. Projection Approach with Recursive Least Squares

The projection approach with recursive least squares method rapidly calculates the orientation and origin position errors between the reference position of the moving object and the position of the sensor measurements through the projection approach. Since sensor measurements are generated under the assumption of Gaussian white noise, the states that represent the origin position and orientation error include uncertainty due to the sensor's noise. These state uncertainty are corrected through probability-based modeling by recursive least squares.

3.1.1. The Reference Object Position Selection Based on Multi-Sensor Object Convergence

Reference position selection is important when attempting to match the misaligned coordinates. In a previous research, RTK-GPS was used to obtain the reference position, however, this approach could not adjust correction parameters online because most of the objects in the real world do not equip RTK-GPS. In the other approach, lidar data were used as the reference position. However, lidar's coordinate can also mismatch to the common coordinate. In this case, even if lidar and other sensor are matched into the same coordinate, this coordinate may still have mismatching with the common coordinate. In other words, the reference position is difficult to define with a specific sensor, such as lidar and other sensors. To overcome these limitations, we define the reference position as the position of the converged track, estimated through multi-sensor object convergence, whereby the tracks represent the result of multi-sensor object convergence. To apply the converged track as the reference position, we assumed that converged track in the near range had no difference with converged track position which the sensor coordinates are matched. This is because, in the case of the object located in close distance, misaligned coordinate affects small position error within the tolerance of sensor position accuracy. Moreover, since multi-sensor object convergence estimates the track position based on combining each sensor's advantages, the converged track information is more reliable than track information from single sensor-based object tracking.

The details of the multi-sensor object convergence method are follows; the object data that were obtained from the multiple sensors have a position and velocity, amongst other information. Generally, the object information came from different sensors are asynchronized to a common timer and do not provide correlation information from each sensor's internal tracking method. It is for these reasons that make it difficult to converge each sensor's information. In order to improve the accuracy and

reliability of information, the methods, nearest neighbor filter (NNF) [24] and cross covariance method (CCM) [25] based multi-sensor object convergence are applied, as shown in Figure 4. These methods are composed of track correction and track merge briefly. In the track correction step, the object data from multiple sensors are used as measurements. The NNF corrects the states and the covariance of the track by calculating the nearest measurement based on the Mahalanobis distance. In addition, the measurement must be located inside of an object identification's region. The object identification's region is the region for identifying the same object and generally the size of the region is obtained from inverse chi-square distribution at 0.95 or 0.99 [26]. If the measurement is not associated with the existing track, the measurement generates a new track. In the track merge step, the CCM converges tracks. Which tracks are identified as the same object, is based on the existing tracks' status and covariance. The track management manages the status of the track and outputs the tracks which are confirmed.

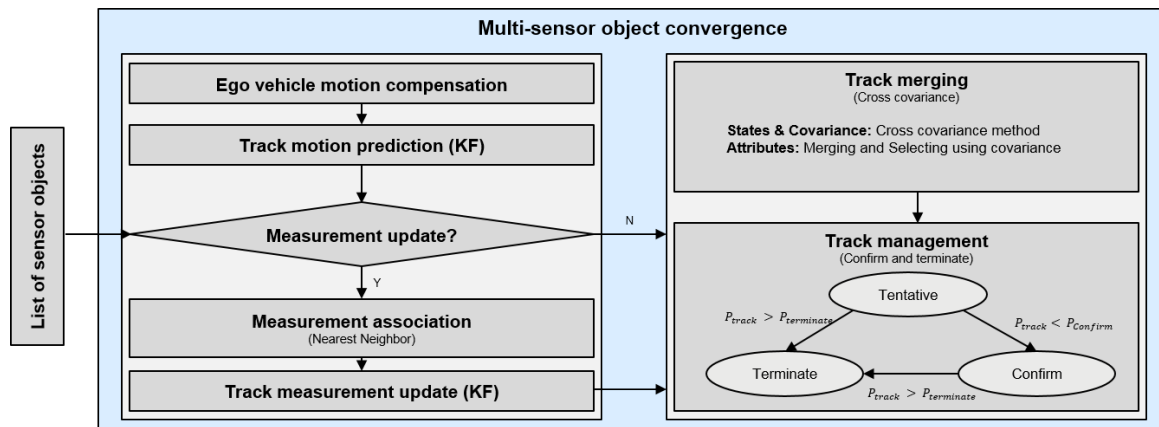


Figure 4. The system architecture of multi-sensor object convergence.

3.1.2. The Error Calculation Based on the Projection Approach

The orientation and origin position error are calculated by projection approach. The orientation error can be calculated by applying the difference between the reference and the measurement position to the inverse trigonometric function, as shown in (6) and Figure 5. As shown in (7) and Figure 6, the origin position error is the position difference between the measurement and the reference when the measurement position, including the orientation error, is projected on the line connecting the reference coordinate origin and the converged object.

$$\Delta\theta_k = \tan^{-1}\left(\frac{y_{meas} - y_{k-1}}{x_{meas} - x_{k-1}}\right) - \tan^{-1}\left(\frac{y_{ref}}{x_{ref}}\right) - \theta_{k-1} \quad (6)$$

$$\begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} - \begin{bmatrix} \cos \Delta\theta_k & -\sin \Delta\theta_k \\ \sin \Delta\theta_k & \cos \Delta\theta_k \end{bmatrix} \begin{bmatrix} x_{meas} - x_{k-1} \\ y_{meas} - y_{k-1} \end{bmatrix} \quad (7)$$

In (6) and (7), $\Delta\theta_k$, Δx_k , Δy_k are the orientation, longitudinal, and lateral position errors between the reference and measurement at time step k ; θ_{k-1} , x_{k-1} , y_{k-1} are the orientation, longitudinal and lateral position errors that have accumulated up to time step $k-1$; x_{meas} , y_{meas} are the longitudinal and lateral position coordinates of the measurement; and x_{ref} , y_{ref} are the longitudinal and lateral position coordinates of the reference.

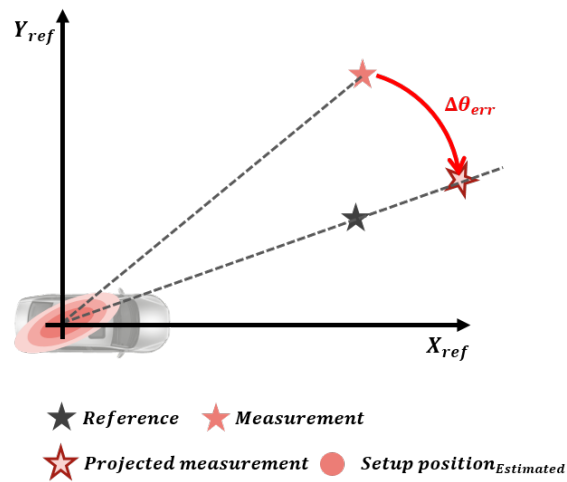


Figure 5. Orientation error estimated by the projection approach.

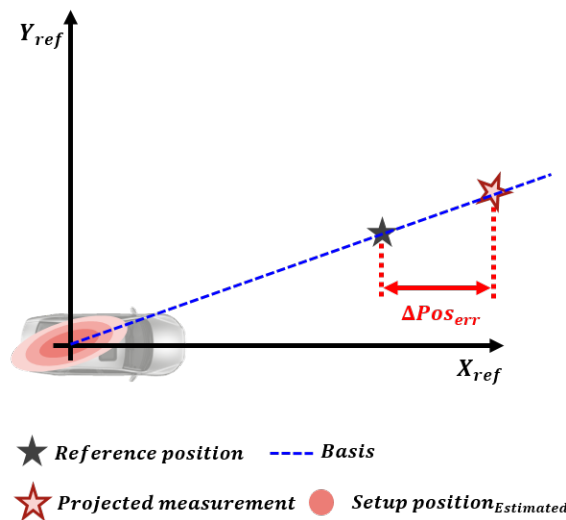


Figure 6. Origin position error estimated by the projection approach.

The accumulated orientation and origin position error up to time step k can be summarized as follows:

$$\theta_k = \tan^{-1} \left(\frac{y_{meas} - y_{k-1}}{x_{meas} - x_{k-1}} \right) - \tan^{-1} \left(\frac{y_{ref}}{x_{ref}} \right) \quad (8)$$

$$x_k = (x_{ref} + x_{k-1}) + (x_{meas} - x_{k-1}) \cos \theta_k - (y_{meas} - y_{k-1}) \sin \theta_k \quad (9)$$

$$y_k = (y_{ref} + y_{k-1}) + (x_{meas} - x_{k-1}) \sin \theta_k + (y_{meas} - y_{k-1}) \cos \theta_k \quad (10)$$

After substituting (8) into (9) and (10), the following equations are derived by applying the trigonometric addition method.

$$\frac{x_{ref} + x_{k-1} - x_k}{\sqrt{(x_{meas} - x_{k-1})^2 + (y_{meas} - y_{k-1})^2}} = \cos \left(\tan^{-1} \frac{y_{ref}}{x_{ref}} \right) \quad (11)$$

$$\frac{y_{ref} + y_{k-1} - y_k}{\sqrt{(x_{meas} - x_{k-1})^2 + (y_{meas} - y_{k-1})^2}} = \sin \left(\tan^{-1} \frac{y_{ref}}{x_{ref}} \right) \quad (12)$$

The equation of the circle in (13) is derived by applying the trigonometric identities to (11) and (12) and taking the limit of a trigonometric function.

$$(x_{meas} - x_k)^2 + (y_{meas} - y_k)^2 = x_{ref}^2 + y_{ref}^2 \quad (13)$$

In (13) and Figure 7, the origin position error $[x_k, y_k]$ must be located at the boundary of a circle which centered at the measurement position $[x_{meas}, y_{meas}]$ and with radius $\sqrt{x_{ref}^2 + y_{ref}^2}$. Since the projected measurement exists on a straight line that passed the sensor coordinates origin and reference position, the intersection point of this straight line and the circle is the origin position error $[x_k, y_k]$. Therefore, the straight line passing through the sensor coordinate origin and reference position is the basis on which the position error can be determined. If the basis has various positions sequentially, the states converge to the constant value.

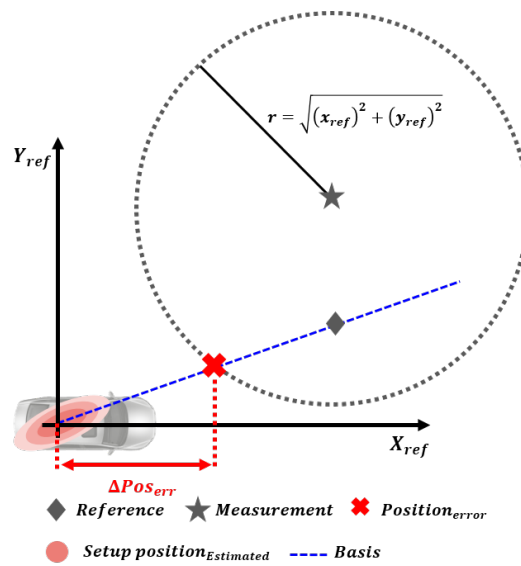


Figure 7. The intersection point between basis and the equation of circle.

3.1.3. The Error Correction Based on Recursive Least Squares

The states are calculated by (6) and (7), which are $\Delta\theta_k, \Delta x_k, \Delta y_k$, from the measurements. However, we want to update estimated states with new measurements which are acquired sequentially and can reduce the sensor noise effect. We applied the recursive least squares method (RLS) for the state correction. If the sensor noise is considered as the constant value, RLS has no difference with the low-pass filter. However, the sensors in the autonomous vehicle, which are the camera, radar, and lidar, have performance differences depending on the detected region, such as, whether the target object is at the center of sensor's field of view or at its boundary, it can also differ whether or not the target object is close by or far away in the distance. This performance difference can exist due to distortion and a smaller number of point clouds, and it can exist for various amount of other reasons. Therefore, we determined the RLS rather than other simple filtering methods for the noise effect reduction and optimal state estimation, by setting the large sensor noise at the boundary and far away information. The details of how we set the sensor noise differently, are based on region as described in the Appendix A. There are two advantages of using the RLS briefly. First of all, the RLS is widely used to estimate the reliable states from sequential data by stochastically modeling of the sensor noise, which can be assumed by Gaussian distribution. Secondly, RLS requires a low computational burden to estimate the states because RLS only requires the information in time step $k - 1$ and k [27,28]. One of the cautions to implement the RLS is outlier rejection. The outlier is the measurement that is distant to other measurements. In the case of probabilistic-based estimation,

the outlier causes a decrement of estimation performance. In order to remove the outlier in the sensor measurement sets, we pre-processed the sensor measurement and applied a measurement selection. The pre-processing filters measurements based on several factors included in the acquired sensor measurement information, such as reliability value, quality level, track status, and other factors. In the measurement selection, the converged track determines the measurement based on object identification region and history of converged sensor ID. The equations for the RLS are as follows:

$$K_k = P_{k-1} H_K^T \left(H_K P_{k-1} H_K^T + R_k \right)^{-1} \quad (14)$$

$$x_k^* = x_{k-1}^* + K_k (y_k - H_k x_{k-1}^*) \quad (15)$$

$$P_k = (I - K_k H_k) P_{k-1} \quad (16)$$

In (14)–(16), an estimator gain parameter represented by K , estimated states (estimated origin position and orientation errors) by $x^* = [\theta_{err}^*, x_{err}^*, y_{err}^*]^T$; calculated states (calculated errors from projection approach) by $y = [\Delta\theta_{err}, \Delta x_{err}, \Delta y_{err}]^T$; P is 3 by 3 matrix with uncertainty of estimated states; I and H are the 3 by 3 identity matrices; and R is noise covariance. The subscript k represents time step k .

The procedure of RLS is as follows: When new states from the projection approach are updated, RLS calculates the appropriate gain K which can be minimize the cost functions ($J = x_{ref} - x^*$). The gain K has a strong dependency on sensor noise. Refer to (14) to (16), the sensor noise affects to adjust the K and P . For instance, when the sensor noise increased, K become smaller and P become bigger than before. In other word, If the K is a small value, the RLS sets more weight to accumulated estimation states than the calculated states from projection approach. On the other hand, if the K is a big value, the RLS sets more weight to calculated errors (x_k) from projection approach than accumulated estimation errors (x_{k-1}^*). Therefore, the proposed method estimates the reliable states by obtaining the new states through the projection approach and reducing the state uncertainty through RLS sequentially.

As a results, the proposed method estimates the origin position error (x_{err}^*, y_{err}^*) and orientation error (θ_{err}^*) between the reference and sensor coordinates, as shown in Figure 8.

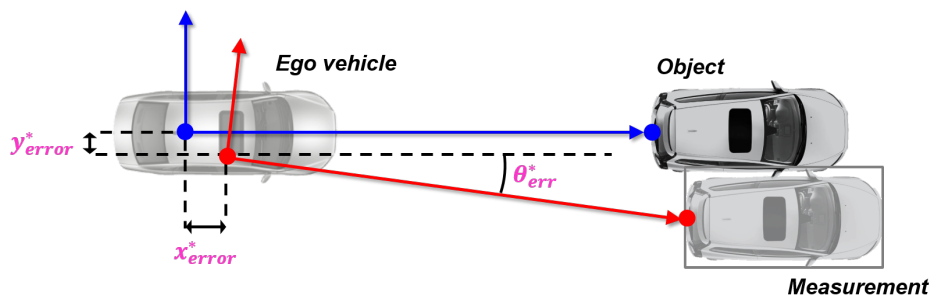


Figure 8. Parameters of sensor position error model.

3.2. Advantages of Projection Approach with Recursive Least Squares

In this paper, we proposed the method of estimating model parameters with the reference and measurement acquired for each time step by iteratively using the projection approach and the recursive least squares. The proposed method has the advantages of reducing computational complexity, and noise effects for estimating the parameters of the under-determined system. The details are as follows. First of all, the computational complexity can be reduced by estimating the parameters of the under-determined system through the rapid mathematical operations which are arithmetic operations, and trigonometric functions. In previous studies, such as ICP, SVD is used to estimate the parameters of the under-determined system. To estimate the parameters of the under-determined

system composed of n equations with m parameters through SVD, at least $(m + 1)$ datasets are needed. Thus, SVD has a computational complexity of $O(n(m + 1)^2)$ at a minimum. In other words, ICP increase the estimation performance and the computational complexity as the amount of datasets increases. However, the proposed method has a computational complexity of $O(n^2)$ because our method is performed with rapid mathematical operations. Therefore, the proposed method can estimate the parameters of the under-determined system even with the less computational resources and computation time than the previous methods. Secondly, the estimated uncertainty due to sensor noise can be reduced. Since the previous method cannot consider information uncertainty due to noise, additional filtering algorithms are necessary in order to estimate the reliable information. On the other hand, the proposed method estimates the reliable parameters by stochastically modeling the noise that has Gaussian distribution assumption through the recursive least squares method. Therefore, the influence of uncertainty, such as noise, is less than that of the previous methods.

4. Experimental Results

4.1. Experimental Environment

The sensors, as shown in Figure 9, are mounted in front of the test vehicle used to verify the proposed online multi-sensor coordinate correction method. The lidar, radar, and camera were mounted on the frontal area of the test vehicle. All the sensors were conducted the coordinate conversion with respect to the vehicle coordinate which origin at the rear axle and follows the ISO coordinate.

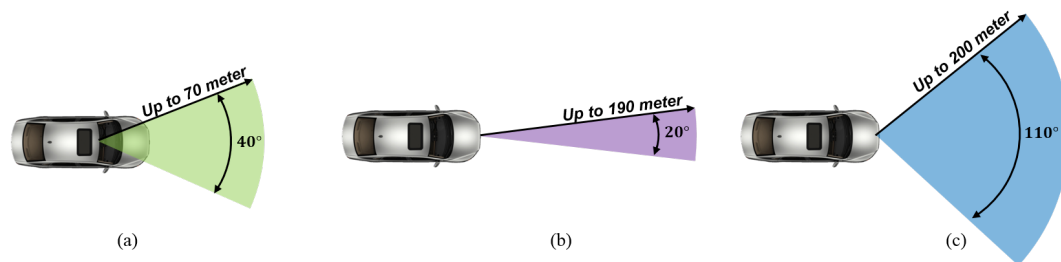


Figure 9. Coverage of sensors on the test vehicle: (a) Camera. (b) Radar. (c) Lidar.

4.2. Comparison between Proposed Method and ICP

To evaluate the performance of the proposed method, the ICP and the proposed method among the estimation algorithms used in a previous study [19] was compared. ICP has the following characteristics: first, the ICP is widely used for three-dimensional motion alignment. Second, the ICP can be applied to estimate the parameters of the proposed sensor position error model because ICP method estimates the rotation and translation parameters that minimizes the position error between the reference and measurement, as shown in (14).

$$e = \sum_i ((Rp_i + T - q_i) \times n_i)^2 \quad (17)$$

In (14), R is the rotation matrix; T is the translation matrix; n_i is the normal vector; p_i is the measurements; and q_i is the reference.

The estimation accuracy and computation time of the ICP algorithm depend on the number of datasets. In order to analyze the trade-off relationship according to estimation accuracy, the computation time and the number of datasets, we performed on the ICP with 4 datasets, 50 datasets, and 100 datasets as shown in Figure 10. The estimation accuracy of the ICP was compared with ground truth. Since the mismatch of the sensors mounted on the test vehicle could not be verified, the error between the converged track and the RTK-GPS was used as the ground truth. If the ICP estimates the parameters with 4 datasets, estimated parameters were seriously affected by sensor's noise. From this,

we can conclude that 4 datasets are not sufficient enough to use for model parameters estimation, even if, these 4 datasets have a has fast computation time of 0.47 ms. In case of 50 datasets, the ICP took 12.11 ms to estimate the parameters, the parameters' RMS error was 0.25 m in the longitudinal direction, 0.1 m in the lateral direction, and 0.19-degrees in orientation. When performed with 100 datasets, the computation time was 60.1 ms, and the parameters' RMS error was 0.3 m in longitudinal direction, 0.08 m in lateral direction, and 0.05-degrees in orientation. Therefore, the greater number of datasets, the slower the computation time and the better the estimation accuracy. Conversely, the smaller number of datasets, the faster the computation time and the lower the accuracy. In addition, when comparing the estimation accuracy of 50 datasets and 100 datasets, the estimation accuracy does not improve in proportion to the increase of the datasets.

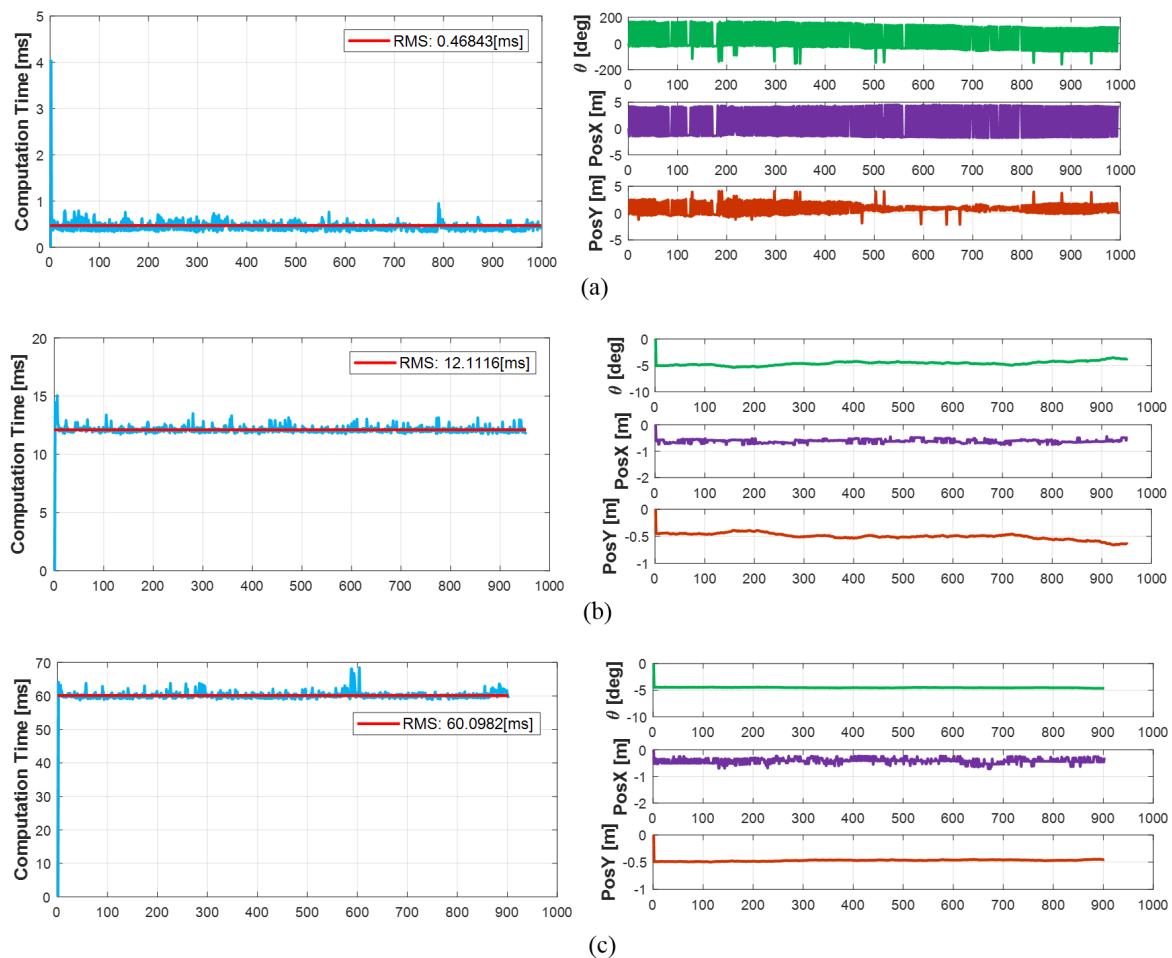


Figure 10. Characteristics of ICP with various datasets: (a) 4 datasets. (b) 50 datasets. (c) 100 data sets.

4.2.1. Computation Time Comparison

To evaluate the computation time reduction using the proposed method, we compared the computation time of the proposed method and the ICP in a PC environment. We used the PC environment for comparing the computation time consisted of an Intel i5-4690 at 3.5 GHz and 8 GB memory. For the computation time comparison, the ICP was performed with 4 datasets (minimum datasets) and 50 datasets. Both methods performed a total of 900 iterations. As shown in Figure 11, the computation time of the ICP showed variation. This is because the ICP internally iterates until the error between the reference and the measurement becomes less than a certain level, thereby changing the number of iterations per dataset. As shown in Figure 11, the ICP took 0.47 ms with the 4 datasets to estimate the parameters, and 11.72 ms with the 50 datasets. On the other hand, the proposed method

took 0.04 ms, therefore, the proposed method reduced the computation time at least 90.8% and up to 99.7% compared to ICP.

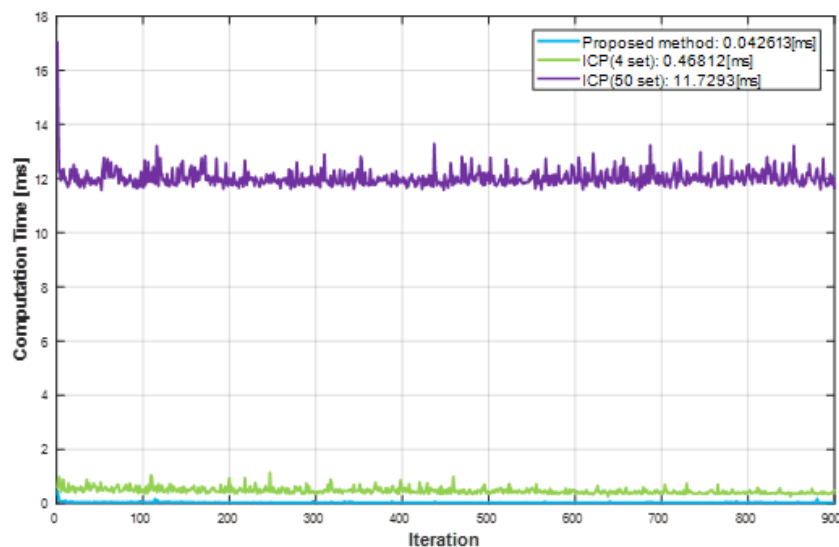


Figure 11. Result of computation time analysis.

4.2.2. Estimation Accuracy Comparison

To evaluate the estimation accuracy of the model parameters, the proposed method and the ICP were compared with ground truth. At this time, the ICP was performed with 50 datasets which have similar estimation accuracy with the proposed method. 100 datasets weren't compared because the computation time of 100 datasets is over the sensor data update period which are among 40 ms to 50 ms as shown in Figure 10. Namely, when we applied the ICP with 100 datasets, the information delay can reduce the reliability of the object's information, as the information is contaminated and missing. Thus, we did not compare 100 datasets performance with the proposed method. The two estimation methods performed a total of 900 iterations.

As shown in Figure 12, the proposed method estimated the state $[\theta_{err}, x_{err}, y_{err}]$ similarly to the ground truth. Additionally, the ICP showed variation in the estimated parameters due to sensor noise, but the proposed method was less affected by noise.

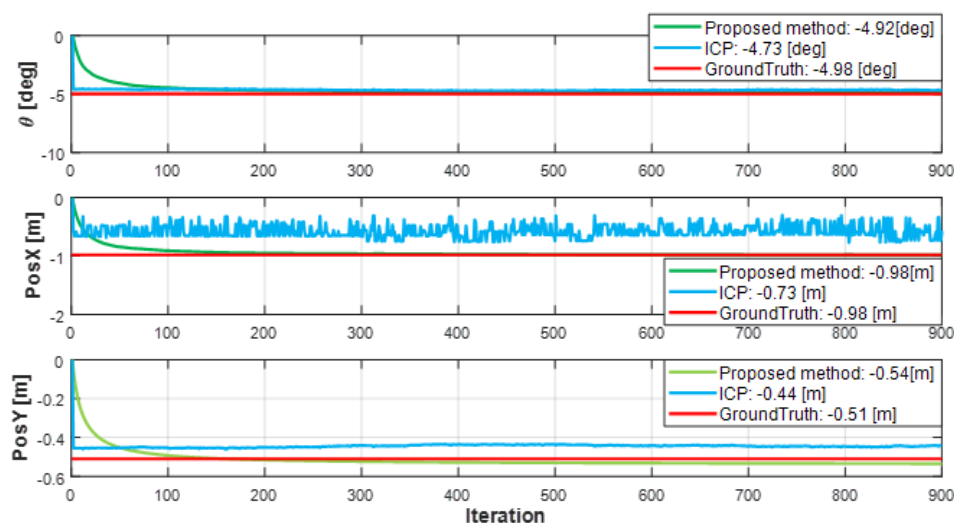


Figure 12. Results of model parameters estimation.

4.3. Effect of Online Multi-Sensor Coordinate Correction

The multi-sensor coordinate mismatch was caused by an object identification problem due to the large position error for the distant moving object. To solve this problem, the performance of the proposed multi-sensor coordinate correction method was evaluated by the position error between converged objects and sensor measurements and the status of the object identification of multi-sensor measurements. For the experiment, the scenario was composed of the target vehicle moving straight forward at 30 km/h in front of the ego vehicle to achieve information from a near to a far distance within the multi-sensors' overlapping region of interest, as shown in Figure 13.

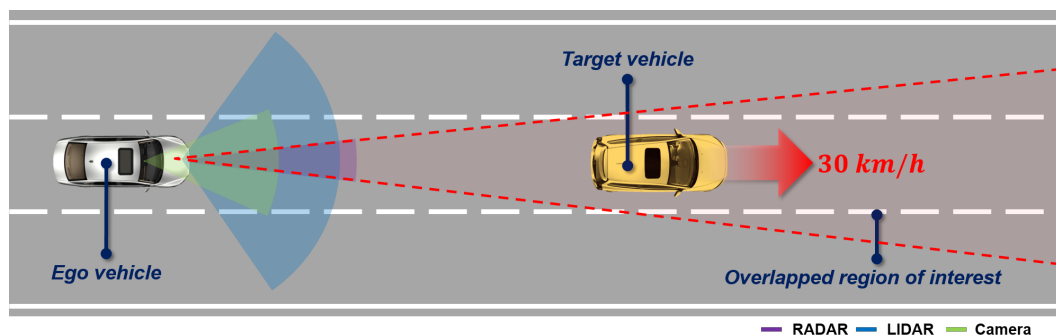


Figure 13. Scenario for evaluating the online multi-sensor coordinate correction.

4.3.1. Position Error between Converged Tracks and Sensor Measurements

To evaluate the performance of the proposed method, we compared the position errors between the converged tracks and their measurements when the proposed method was applied and then when was not applied. As shown in Figure 14, when the proposed method was not used, all of the sensor's object has the position error with the converged object increased with the distance from the target vehicle. If the object identification region was set to a 1 m radius, this position error could be determined as a different object from the converged tracks. However, as shown in Figure 15, the use of the proposed method significantly reduced the position error to the converged object. In the case of the radar, the distance error to the converged object was reduced within 1 m up to 190 m away; also, the camera had a distance error within 1 m from the converged object up to 70 m away; and lidar had a position error less than 0.5 m in all range. Therefore, the proposed method overcame the problem that the position error increased as the distance increased due to the slightly mismatched coordinate.

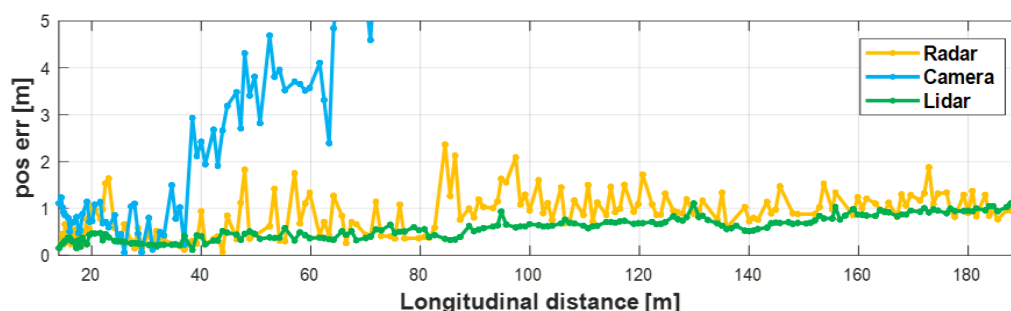


Figure 14. Position error when not using the proposed method.

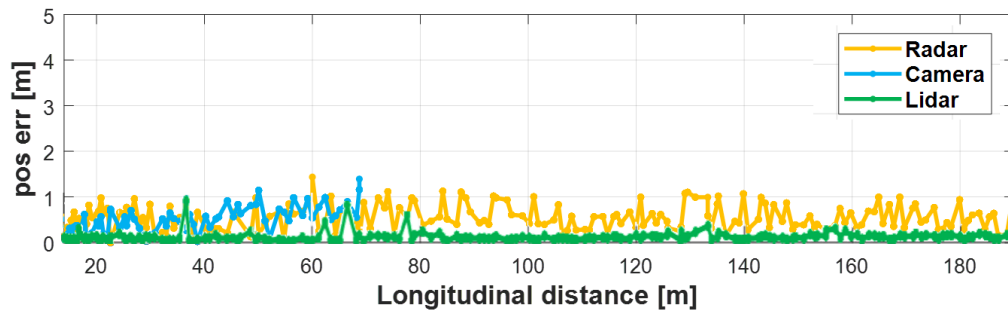


Figure 15. Position error when using the proposed method.

4.3.2. Object Identification Status of Multi-Sensor

To evaluate the performance of the proposed method, the object identification status was compared when the proposed method was used and when the proposed method was not used. The rate of unidentified measurements was used as an evaluation index for the object identification status as shown in (18).

$$\frac{\text{The number of unidentified measurements}}{\text{Total number of detections}} \times 100 (\%) \quad (18)$$

As can be seen in Figure 16a, 7.7% of radar measurements were misidentified when the proposed method was not applied. Likewise, as shown in Figure 17a, the measurements from camera were misidentified by 43.7% of the measurements. In the case of lidar, 0.04% of measurements were unidentified, as shown in Figure 18a. The reason for the lowest unidentified rate of lidar is that the reliability of lidar object's position is the most reliable among the sensors, thus, the multi-sensor object convergence more depends on lidar than other sensors when the converged tracks are estimated. When the proposed method applied to these sensor measurements, the unidentified rates became significantly reduced. 0.04% of radar measurements were misidentified; 23.2% of camera measurements were misidentified; and 0% of lidar measurements are misidentified, this can be seen Figures 16b, 17b, and 18b. As a result, the multi-sensor coordinate correction method reduced 94.8% of misidentification in radar measurements, 46.9% of misidentification in camera measurements, and 100% of misidentification in lidar measurements.

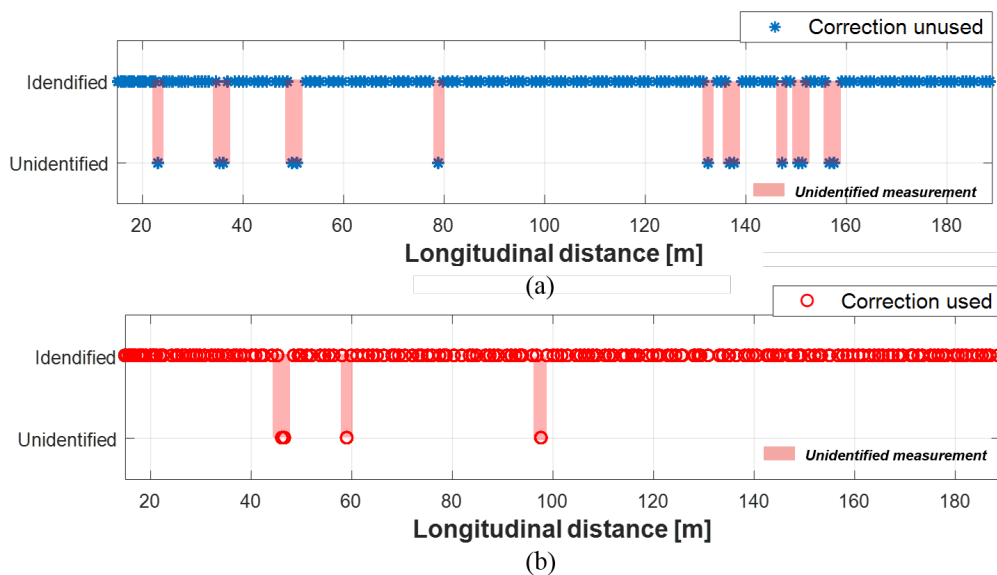


Figure 16. Object identification status of radar: (a) Without correction. (b) With correction.

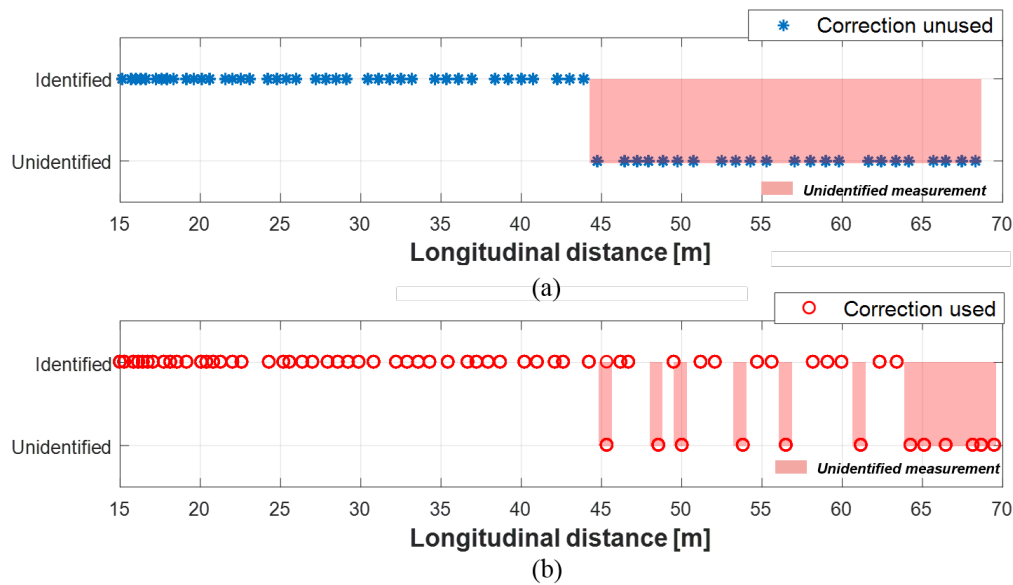


Figure 17. Object identification status of camera: (a) Without correction. (b) With correction.

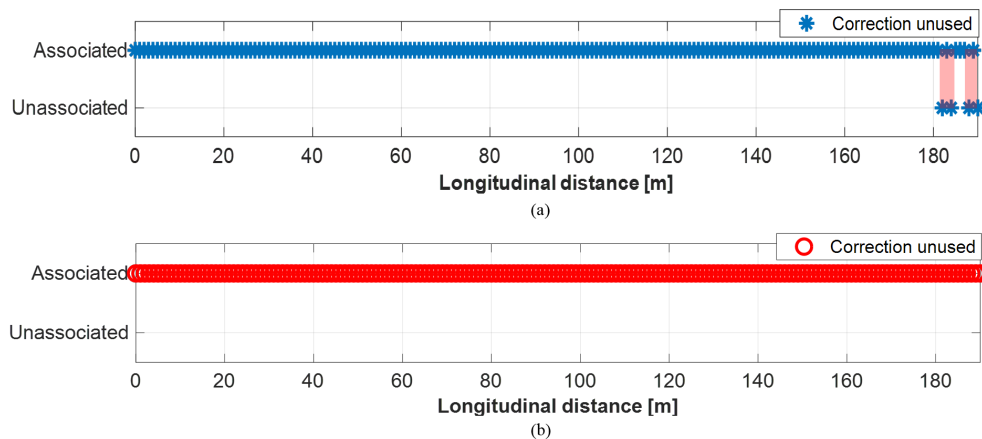


Figure 18. Object identification status of lidar: (a) Without correction. (b) With correction.

5. Conclusions

This paper proposed the fast online coordinate correction method with reduced computational effort for matching the slight mismatched sensor coordinates. The advantages of this study can be summarized as follows:

The proposed method has lower computational complexity than the conventional methods, such as ICP, LSM, and other methods, because the proposed method can estimate the parameters of the under-determined system by using rapid mathematical operations, unlike previous studies that require the pseudo inverse operation or SVD. According to the experimental results, the proposed method reduced the computation time by up to 99.7% (at least 90.8%) compared with the ICP and the estimated parameters were almost the same as the ground truth. In addition, the proposed method was less affected by sensor noise than the ICP because it could probabilistically model the uncertainty of the dataset.

Since the proposed method estimates the parameters of an under-determined system using only simple mathematical operations without the pseudo inverse and SVD, the presented method can be adopted in an embedded system that does not support a floating point operation. Furthermore, a system using ICP, such as map matching, can apply the proposed method to estimate the rotation and translation parameters with faster and higher accuracy. Additionally, the proposed method can perform as an auto-calibration of heterogeneous sensors in the driver-assistance system

and autonomous vehicles to maintain the reliability of estimated moving object's information using a multi-sensor based perception system.

Although the proposed method improves the object's identification performance of multi-sensor object convergence, it does not guarantee performance in complex environments such as crowded vehicles, which is due to the reason that the object's identification region is set to a constant size. In order to guarantee the object's identification performance in various environments, the authors plan to research an adaptive object identification region resizing method based on mismatched coordinate parameters. The authors also plan to integrate the proposed method into the embedded system for the feasibility test of mass production.

Author Contributions: Conceptualization, W.L. and M.L.; methodology, W.L. and M.L.; software, W.L.; validation, W.L. and M.L.; formal analysis, M.L.; writing—review and editing, M.S. and K.C.; visualization, W.L.; supervision, K.C.; project administration, M.L.; funding acquisition, M.S.

Funding: This work was financially supported by the BK21 plus program (22A20130000045) under the Ministry of Education, Republic of Korea, the Industrial Strategy Technology Development Program (No. 10039673, 10060068, 10042633, 10079961, 10080284), the International Collaborative Research and Development Program (N0001992) under the Ministry of Trade, Industry and Energy (MOTIE Korea), and National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2011-0017495).

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

Appendix A

Sensors have different performances depending on the detected region, therefore, we set the sensor noise differently based on detected region. To set the noise of each sensor appropriately, we performed the sensor evaluation based on grid cells. The grid cells-based sensor evaluation stacks a large amount of measurement sets to obtain reliable information; in our case, we stacked 100 measurement sets which covered the whole field of view of each sensor. After gathering the measurement sets, we divided the measurement sets into a certain size of grid cells and calculated the mean position error and variance as shown in Figure A1. The position error is the difference between reference (RTK-GPS) and sensor.

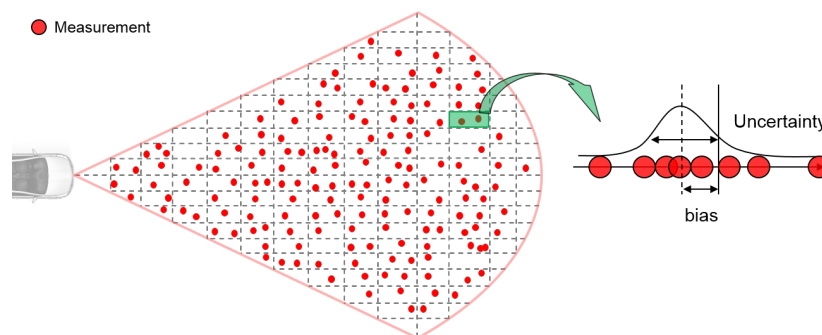


Figure A1. The concept of grid cells based sensor evaluation.

The results of grid cells based sensor evaluation, show that the sensors had a different position accuracy and variance depending on whether the target object was at the center of the field of view or at the boundary, or whether the target object was near to or far from it. In Figure A2, X is the longitudinal position, Y is lateral position, ellipsed with the green line is a variance of position error, and the red circle is mean position error. In Figure A2, all of the sensors have a small position error and variance when the target object is at the center and near in distance. If the target object went to the boundary of the field of view or far away, the mean position error and variance become larger.

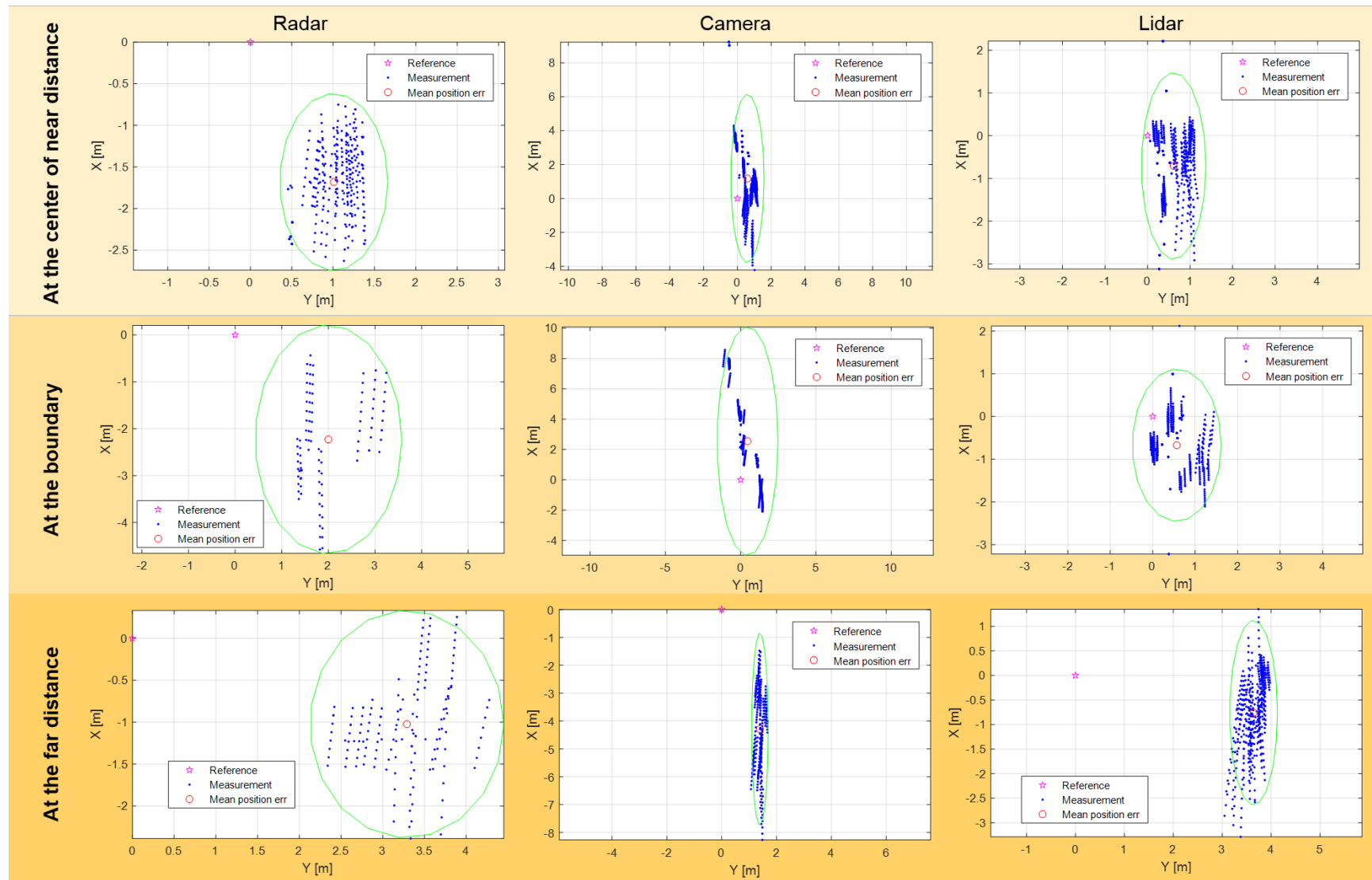


Figure A2. The mean position error and variance of the radar, camera, lidar at center of field of view, boundary, and far distance.

References

1. Geronimo, D.; Lopez, A.M.; Sappa, A.D.; Graf, T. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1239–1258. [[CrossRef](#)] [[PubMed](#)]
2. Franke, U.; Pfeiffer, D.; Rabe, C.; Knoeppel, C.; Enzweiler, M.; Stein, F.; Herrtwich, R. Making bertha see. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 1–8 Decembver 2013; pp. 214–221.
3. Jang, C.; Cho, S.; Jeong, S.; Suhr, J.K.; Jung, H.G.; Sunwoo, M. Traffic light recognition exploiting map and localization at every stage. *Expert Syst. Appl.* **2017**, *88*, 290–304. [[CrossRef](#)]
4. Han, J.; Kim, D.; Lee, M.; Sunwoo, M. Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor. *IEEE Trans. Veh. Technol.* **2012**, *61*, 971–985. [[CrossRef](#)]
5. MacLachlan, R. *Tracking Moving Objects From a Moving Vehicle Using a Laser Scanner*; Technology Report CMU-RI-TR-05-07; Robotics Institute: Pittsburgh, PA, USA, 2005.
6. Fayad, F.; Cherfaoui, V. Tracking objects using a laser scanner in driving situation based on modeling target shape. In Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 44–49.
7. Reina, G.; Underwood, J.; Brooker, G.; Durrant-Whyte, H. Radar-based perception for autonomous outdoor vehicles. *J. Field Rob.* **2011**, *28*, 894–913. [[CrossRef](#)]
8. Dickmann, J.; Appenrodt, N.; Bloecher, H.L.; Brenk, C.; Hackbarth, T.; Hahn, M.; Klappstein, J.; Muntzinger, M.; Sailer, A. Radar contribution to highly automated driving. In Proceedings of the 2014 44th European Microwave Conference (EuMC), Rome, Italy, 6–9 October 2014; pp. 1715–1718.
9. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, 13. [[CrossRef](#)]
10. Darms, M.; Rybski, P.; Urmson, C. Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 1197–1202.
11. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
12. Ziegler, J.; Bender, P.; Schreiber, M.; Latégahn, H.; Strauss, T.; Stiller, C.; Dang, T.; Franke, U.; Appenrodt, N.; Keller, C.G.; et al. Making Bertha Drive-An Autonomous Journey on a Historic Route. *IEEE Intell. Transport. Syst. Mag.* **2014**, *6*, 8–20. [[CrossRef](#)]
13. Cho, H.; Seo, Y.W.; Kumar, B.V.; Rajkumar, R.R. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1836–1843.
14. Kunz, F.; Nuss, D.; Wiest, J.; Deusch, H.; Reuter, S.; Gritschneider, F.; Scheel, A.; Stübler, M.; Bach, M.; Hatzelmann, P.; et al. Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 666–673.
15. Li, X.; Wee, W.G. Sensor error modeling and compensation for range images captured by a 3D ranging system. *Meas. Sci. Technol.* **2008**, *19*, 125502. [[CrossRef](#)]
16. Underwood, J.P.; Hill, A.; Peynot, T.; Scheduling, S.J. Error modeling and calibration of exteroceptive sensors for accurate mapping applications. *J. Field Robot.* **2010**, *27*, 2–20. [[CrossRef](#)]
17. Flanagan, B.P.; Bell, K.L. Array self-calibration with large sensor position errors. *Signal Process.* **2001**, *81*, 2201–2214. [[CrossRef](#)]
18. Whitehouse, K.; Culler, D. Calibration as parameter estimation in sensor networks. In Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, USA, 28 September 2002; pp. 59–67.
19. Budge, S.E. A calibration-and-error correction method for improved texel (fused ladar/digital camera) images. *Proc. SPIE* **2012**, *8379*, 83790S.
20. Hirsenkorn, N.; Hanke, T.; Rauch, A.; Dehlink, B.; Rasshofer, R.; Biebl, E. A non-parametric approach for modeling sensor behavior. In Proceedings of the 2015 16th International Radar Symposium (IRS), Dresden, Germany, 24–26 June 2015; pp. 131–136.
21. Hirsenkorn, N.; Hanke, T.; Rauch, A.; Dehlink, B.; Rasshofer, R.; Biebl, E. Virtual sensor models for real-time applications. *Adv. Radio Sci.* **2016**, *14*, 31–37. [[CrossRef](#)]

22. Craig, J.J. *Introduction to Robotics: Mechanics and Control*, 3/E; Pearson Education: Karnataka, India, 2009.
23. Anton, H.; Busby, R.C. *Contemporary Linear Algebra*; Wiley: New York, NY, USA, 2003.
24. Li, X.R.; Bar-Shalom, Y. Tracking in clutter with nearest neighbor filters: Analysis and performance. *IEEE Trans. Aerosp. Electron. Syst.* **1996**, *32*, 995–1010.
25. Bar-Shalom, Y. On the track-to-track correlation problem. *IEEE Trans. Autom. Control* **1981**, *26*, 571–572. [[CrossRef](#)]
26. Mušicki, D.; Morelande, M. Gate volume estimation for target tracking. In Proceedings of the 7th International Conference on Information Fusion (Fusion 2004), Stockholm, Sweden, 28 June–1 July 2004.
27. Ljung, S.; Ljung, L. Error propagation properties of recursive least-squares adaptation algorithms. *Automatica* **1985**, *21*, 157–167. [[CrossRef](#)]
28. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: New York, NY, USA, 2006.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).