Lane-Level Localization Using an AVM Camera for an Automated Driving Vehicle in Urban Environments

Dongwook Kim, Beomjun Kim, Taeyoung Chung, and Kyongsu Yi, *Member, IEEE*

***Abstract*—This paper describes a lane-level localization algorithm based on a map-matching method for application to automated driving in urban environments. The lane-level localization implies localizing the vehicle with centimeter-level accuracy. In order to achieve a satisfactory level of position accuracy with a low-cost GPS, a sensor fusion approach is essential for lane-level localization. The proposed sensor fusion approach for the lane-level localization of a vehicle uses an around view monitoring (AVM) module and vehicle sensors. The proposed algorithm consists of three parts: lane detection, position correction, and localization filter. In order to detect lanes, a commercialized AVM module is used. Since this module can acquire an image around the vehicle, it is possible to obtain accurate position information of the lanes. With this information, the vehicle position can be corrected by the iterative closest point (ICP) algorithm. This algorithm estimates the rigid transformation between the lane map and lanes obtained by AVM in real-time. The vehicle position corrected by this transformation is fused with the information of vehicle sensors based on an extended Kalman filter. For higher accuracy, the covariance of the ICP is estimated using Haralick's method. The performance of the proposed localization algorithm is verified via vehicle tests on a proving ground. Test results show that the proposed method can achieve localization centimeter-level accuracy. The proposed algorithm will be useful in the implementation of automated driving control.**

***Index Terms*—Around view monitoring, automated vehicle, extended Kalman filter-iterative closest point (EKF-ICP), lane-level localization.**

Manuscript received August 17, 2015; revised November 27, 2015 and January 30, 2016; accepted February 8, 2016. Date of publication March 14, 2016; date of current version February 14, 2017. Recommended by Technical Editor X. Chen. This work was supported by the Hyundai Mobis, the BK21 program, SNU-IAMD, and the National Research Foundation of Korea grant funded by the Ministry of Science, ICT, and Future Planning 2009–0083495.

D. Kim and B. Kim are with the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-744, Korea (e-mail: mechar04@snu.ac.kr; boemjun@snu.ac.kr).

T. Chung is with Hyundai Mobis, Uiwang-si 437-718, Korea (e-mail: inxskick@mobis.co.kr).

K. Yi is with the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-744, Korea (e-mail: kyi@snu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2016.2533635

I. Introduction

Recently, automated driving has been widely regarded as mainstream in the automotive industry since it offers increased comfort and safety. Many motor vehicle manufacturers aim to commercialize self-driving cars by 2020 and are spurring intelligent vehicle research to realize this. Among these research activities, localization has emerged recently as one of the hottest issues in the development of autonomous vehicles. For vehicle navigation, accurate and robust localization is required. Much research of vehicle localization has been done over the last decade. A comprehensive overview is given in [1]. Currently, there is much interest in the vehicle localization based on high-definition map [2]–[4]. This is because the latest sensors and their processors still do not reach a satisfying level of development in terms of robustness and availability at various environments [5]. A digital map is built that includes sufficient information to determine the current position of the autonomous vehicle relative to this map. Thus, the digital map is used as a powerful additional sensor to improve the performance of the vehicle localization.

In this paper, three research issues are considered in map-aided vehicle localization: features for map-matching, method of correcting the vehicle position, and a localization filter. The digital maps used for localization can be mainly classified according to features for map-matching. The most commonly used feature is a lane marking. Lane markings are in standard use and exist on almost all roads. Thus, many researches [2], [3], [6], and [7] use lane marking data to build their digital map. The second most used feature is a curb [2], [8]. Curbs usually appear at the borders between streets and sidewalks. They are another important feature to determine the drivable area. Besides these two features, three-dimensional features [9], keypoints [10], visual features [11], and GPS shadow [12] are also considered to generate the digital map for the map-aided localization. Based on the works above, it is apparent that map with these features enhances the performance of vehicle localization. However, features other than lane markings still have problems with the standardization of the feature map. This is because lane markings have a consistent appearance and are usually painted based on rules given by the government. In contrast, other features have many exceptions that make it difficult to build a large map. To detect other features, high-cost sensors such as a stereo camera [2] or a LiDAR [8]–[10] are required. So, in our study, we only use lane markings as features for map-matching and analyze the performance of localization based on this feature.

To correct the vehicle position using a map and feature data obtained by sensors in real-time, a map-matching algorithm is needed. The most famous method of map-matching is the iterative closest point (ICP) algorithm initially proposed in [13]. Given two point clouds, the ICP algorithm estimates the rigid transformation between them. The ICP algorithm sets up correspondences between the source and the target point clouds. Then, it finds a transformation that minimizes an error metric function and transforms the source point cloud based on this transformation. It iterates over these steps until the residual error between the source and the target point clouds is smaller than a certain threshold. There are many variants of ICP [14]. Among these variants of ICP, the point-to-plane approach is known to be very accurate and fast [15], so we apply this approach to our algorithm. In spite of an enormous amount of research effort to improve the accuracy of ICP [16], these variants still can result in inaccurate alignment due to the problems of noise and outliers. It is therefore essential to fuse the vehicle position derived from the map-matching algorithm with other sensors. To implement it properly, a precise estimate of ICP's covariance is needed. The covariance estimator of minimization algorithms such as ICP was initially proposed in [17], and has come to be referred to as Haralick's method [18]. Haralick's method is based on the Hessian of the cost function with respect to the estimated displacement and the derivative of the Jacobian of the cost function with respect to the measurements. In [19], this method was adapted to ICP for the first time. In our study, we adjust this method to apply to our matching algorithm.

There is a body of work in the field of localization filters (nonlinear filter): mono-model approaches (EKF, UKF, DD1, DD2: the first-order and second-order divided difference filter [20]–[22]), multimodel [23], [24], and particle filter [25]. As a result of improvements in estimation performance, the complexity and computational load of developed localization filters are increasing. Although estimation performance is an important consideration for localization filters, there are other practical issues related to the filter selection. One of the most important practical considerations is the computational load of the filter, especially for real-time applications. Based on both theoretical and empirical analyses of a particular application, the extended Kalman filter (EKF) is well known for its computational efficiency compared to other nonlinear filters [26]. As long as the process or measurement model is not strongly nonlinear, a nonlinear filtering algorithm is less likely to be useful [27]. Because of this, our localization filter is designed based on an EKF.

There are two main differences between the proposed localization approach and the previous works. The main difference is sensor configuration to detect map-matching features. Conventional researches use front cameras or LiDARs. These approaches have disadvantage in a heavy traffic situation because its field of view (FOV) is disturbed by other vehicles. In order to overcome this issue, an around view monitoring (AVM) camera was used in our approach. This is described in more detail in Section II-A. The other difference is consideration of an error caused by mismatching. In previous work, the error variances were only determined by sensors noises. By incorporating the error caused by mismatching as well as the sensor noise in the proposed algorithm, the proposed localization performance is more enhanced.

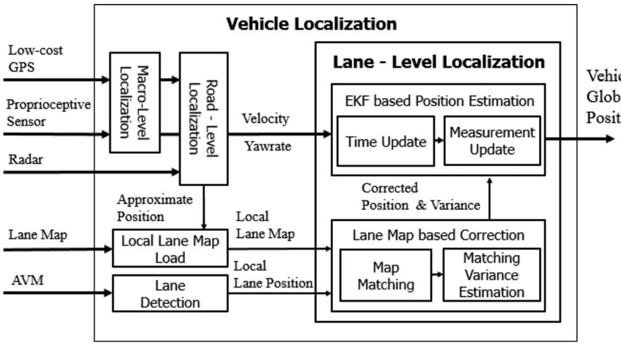


Fig. 1. Structure of vehicle localization.

The remainder of this paper is organized as follows: Section II gives an overview of our localization algorithm. Section III describes the process of detecting the lane that is to be used based on map-matching features. Section IV describes the method of map-matching and estimation of ICP's covariance. Section V describes the lane-level localization solver based on an EKF. Section VI includes results from a series of experiments using a test vehicle in a proving ground. Finally, Section VII concludes this paper.

II. Localization System Overview

A. Vehicle Localization Architecture

The vehicle localization is performed using three steps: macro-level, road-level, and lane-level localization. These steps are depicted in Fig. 1.

Macro-level localization has already been widely used to set up car navigation systems based on low-cost GPS and proprioceptive sensors. The accuracy of these systems is in the order of 10 m [28]. The objective of the road-level localization is to determine the lane in which the ego-vehicle is located on a multilane road. The determined lane is used as the initial guess for the lane-level localization. The key idea is to recognize surrounding vehicles and road boundaries by radar under the assumption that the maximum number of lanes is known by the GPS and the lane map information. The detailed description of road-level localization was presented in [29]. However, road-level accuracy of localization is not enough to control the automated vehicle on urban roads. The required precision for automated driving is within a few centimeters [2], i.e., lane-level localization is essential.

Three research issues are considered in lane-level localization: lane detection using AVM, position correction based on map-matching, and a localization filter. For the lane detection, we use AVM cameras enabling one to obtain a top-view image around the vehicle. There are four reasons for using an AVM module for the lane-level localization. First, the top-view image obtained by AVM enables direct calculation of the lateral offset without any model for lane tracking. The tracking models may sometimes result in inaccurate lateral offset due to road conditions, especially a road slope. Second, it is barely affected

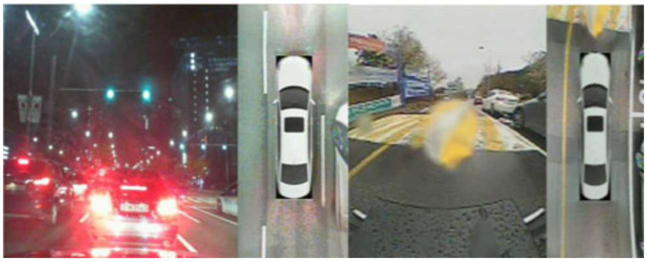


Fig. 2. Comparing front camera and AVM: left (heavy traffic condition) and right (rainy condition).

by invisible lane images caused by neighboring vehicles even in heavy traffic. As shown in Fig. 2, the lane markings may be occluded by other vehicles in front when using the front camera. However, there are few neighboring vehicles that invade AVM's FOV in the same situation.

Furthermore, the whole straight lane marking at the side can be observed, which makes it easier and more robust to detect the lane marking [42]. Third, AVM cameras are more robust to weather and illumination conditions than front cameras, because AVM cameras are mounted toward the ground. Finally, AVM cameras are pervasively used in driver assistance systems in mid-class vehicles. For these reasons, we think that an AVM camera is the most suitable sensor for lane-level localization. However, AVM cannot be used alone due to its small FOV. Thus, map-matching is required to localize the vehicle relative to the digital map. For this, the ICP algorithm is used. The ICP algorithm is widely used in spatial and geometric alignment. In this paper, we applied the point-to-plane matching method to correct the vehicle position. To solve the problem of false matching, we estimate the covariance of the ICP algorithm and set up a validation gate in the localization filter. Finally, the localization filter is designed based on an EKF. The corrected vehicle position obtained by map-matching is used as an observation inside a Kalman filter framework. The main contribution of this paper is the application of existing techniques to develop a low-cost system of lane-level localization and its experimental validation. Detailed explanations of each algorithm are given in the following sections.

B. Test Vehicle Configuration

Our test vehicle is equipped with close-to-production sensors and a referencing system. We have a radar and two single-layer LiDARs mounted on the front bumper. For lane detection, a monocular vision system was mounted on the windshield and AVM cameras were mounted on each side of the vehicle. A low-cost GPS was also mounted for localization, as well as a RTK-GPS receiver for the mobile mapping process and ground truth. The RTK-DGPS is completely independent of the GPS input to the system. The low-cost GPS and DGPS have accuracies of about 2.5 and 0.02 m circular error probable, respectively. The actuator module contains steering, throttle, and brake actuators. These systems are interfaced using the control area network bus. The command signals are transmitted digitally. The controller consists of a computer and microautobox. The complete sensor, actuator, and controller setup is shown in Fig. 3. To power this equipment, an additional sub-battery is installed in the trunk space, and the sub-battery has a mechanism to be charged by an alternator. Moreover, a 220-V, 2000-W inverter system is also installed. For lane-level localization, we use proprioceptive sensors (velocity, yaw-rate) and AVM cameras.

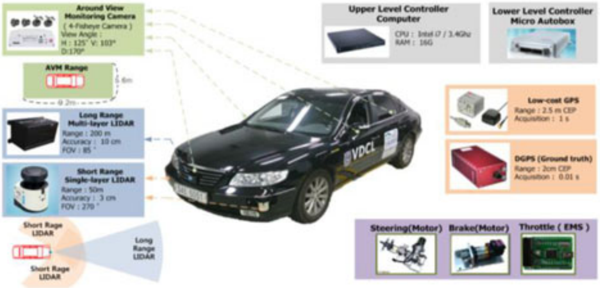


Fig. 3. Test vehicle configuration.

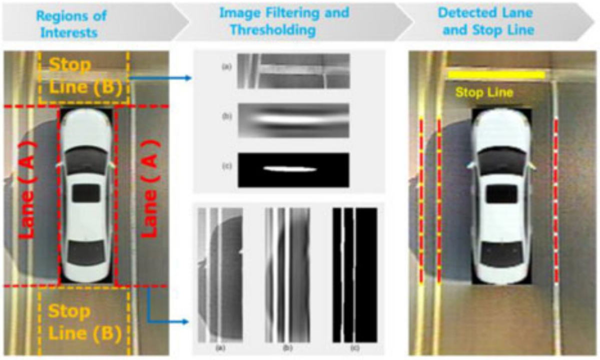


Fig. 4. Schematic diagram of the lane and stop line detection: (a) grayscale image, (b) image after filtering, and (c) image after thresholding.

III. Lane Detection Using an AVM Camera

For detecting the lane, we use a commercial AVM module [41]. The advantage of the AVM has already been described in Section II. This module is composed of four fish-eye cameras mounted on each side of the vehicle. Usually, there are many steps involved in the image processing of AVM cameras to obtain the top-view image around the vehicle: camera calibration, warping distortion rectification, view-point conversion, image stitching, and so on [30]. However, a detailed description of these processes is skipped here, because we use a commercial AVM module that provides top-view images.

To improve the computational efficiency, regions of interest (ROI) were set as shown on the left of Fig. 4. The ROI reduce the searching area by assigning the line type (lane or stop line, vertical or horizontal line) detected in each ROI.

The red- and orange-dashed boxes (A and B) in Fig. 4 indicate the ROI in order to detect lanes and stop lines. Images outside the ROI are not used due to the possibility of image distortion. Then, each region's images are filtered by a two-dimensional Gaussian kernel [31]. The vertical direction of image A and the horizontal direction of image B are smoothing Gaussians, whose $\sigma_{y,A}, \sigma_{x,B}$ are adjusted according to the required height of the lane and stop lines to be detected

$$f_{v,A}(y_A) = \exp\left(-\frac{y_A^2}{2\sigma_{y,A}^2}\right) \quad (1)$$

$$f_{h,B}(x_B) = \exp\left(-\frac{x_B^2}{2\sigma_{x,B}^2}\right), \tag{2}$$

where *f* is a Gaussian function for calculating the transformation to apply to each pixel in the image. x is the distance from the origin in the horizontal axis and *y* is the distance from the origin in the vertical axis. Subscripts A/B and *x/y* represent images A and B and horizontal/vertical axis, respectively. The horizontal direction of image A and the vertical direction of image B are the second derivatives of Gaussians, whose $\sigma_{x,A}, \sigma_{y,B}$ are adjusted according to the expected width of the lanes and stop lines

$$f_{h,A}(x_A) = \frac{1}{\sigma_{x,A}^2}\exp\left(-\frac{x_A^2}{2\sigma_{x,A}^2}\right)\left(1-\frac{x_A^2}{2\sigma_{x,A}^2}\right) \tag{3}$$

$$f_{v,B}(x_B) = \frac{1}{\sigma_{y,B}^2}\exp\left(-\frac{y_B^2}{2\sigma_{y,B}^2}\right)\left(1-\frac{y_B^2}{2\sigma_{y,B}^2}\right). \tag{4}$$

Each filter A and B is adjusted for vertical (lane) and horizontal (stop line) bright lines of a specific width on a dark background. Fig. 4(b) show the images filtered by these Gaussian kernels. As shown in Fig. 4(b), areas where lanes or stop lines exist have high response. These filters can also handle quasi-vertical and quasi-horizontal lines, which produce considerable output after the thresholding process. The threshold value was determined by selecting the *q*% quantile value from the filtered image. The filtered image was then binarized by this threshold value. In this paper, q is set to 95%. Fig. 4(c) shows the result after thresholding.

Next, the pixels exceeding the threshold value are reprojected in the vehicle coordinate system. To find a polyline expression that best approximates the lane data, the RANSAC (random sample consensus) [32] algorithm was used. Lanes and stop lines in the AVM images can be sufficiently represented by a second-order polynomial. So, we use a second-order polyline for expressing the line. Two thresholds are necessary for the RANSAC algorithm; the first one indicates the expected outliers rate in the points set, which is directly related to the iterations number, and the second one specifies the distance above which a point is considered an outlier. In this paper, we set the iterations number and the threshold distance to determine outliers to 50 and 0.05 m, respectively. The results of lane and stop line detection are shown in Fig. 5. Red-dashed lines represent the extracted lane marker and yellow solid lines represent the detected stop lines. It can be seen that the various lane markers are well extracted. Final outputs of the lane detection algorithm are points of the lane approximated by a second-order polynomial. These points are maintained at 10-cm intervals.

IV. Position Correction

A. Map-Matching Based on ICP

Position correction can be achieved by matching the digital map with the lane data obtained by AVM. The digital map used for our approach has been presented in [35]. This map includes geo-localized lane markings. For real-time purposes, a limited number of parameters must represent the map line segments.

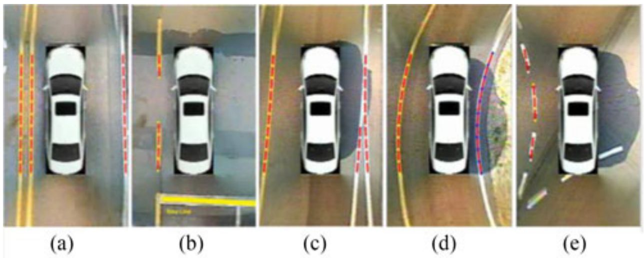


Fig. 5. Result of various lines detection: (a) straight double and single line, (b) dashed line and stop line, (c) merged line, (d) curved line, and (e) curved and dashed line.

As mentioned in Section I, we use a two-dimensional point-to-plane ICP algorithm for map-matching. The point-to-plane ICP was initially introduced by Chen and Medioni [36] and has come into widespread use as a faster and more accurate variant of standard ICP. This algorithm improves performance by using surface normal information. In common with standard ICP, the point-to-plane ICP algorithm finds the best transformation between two point clouds, iteratively repeating the following two steps until the alignment error is smaller than a set threshold:

1) compute correspondences between the two point clouds;
2) compute a transformation which minimizes the error metric function between corresponding points.

The only difference is the error metric function. Instead of minimizing the Euclidean distance between corresponding points, the point-to-plane algorithm minimizes error along the surface normal. This error metric function can be written as

$$J = \sum_{i=1}^{N} ||\eta_i \cdot (R \cdot p_i + T - q_i)||^2 \tag{5}$$

where $p_i = [p_{i,x}, p_{i,y}]^T$ and $q_i = [q_{i,x}, q_{i,y}]^T$ are the N correspondences used in the iteration of ICP; and η_i is the surface normal at q_i. In our study, p_i represents the points obtained by the lane detection algorithm and q_i represents the points of the lane map with respect to ego-vehicle coordinates. *R* and *T* are the rotation matrix and translation vector estimated by the ICP algorithm. Because we deal with two-dimensional ICP in this paper, *R* and *T* can be written as

$$R = \begin{bmatrix} \cos(r) & -\sin(r) \\ \sin(r) & \cos(r) \end{bmatrix}, \quad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{6}$$

where r, t_x, t_y are the matching result (amount of angle and position correction). This result can be used to correct the vehicle position. Let (X_V, Ψ) express the current vehicle position. The corrected vehicle position (X_V^*, Ψ^*) can be derived [37] as

$$\begin{aligned} X_V^* &= R_\Psi \cdot T + X_V \\ \Psi^* &= r + \Psi \\ \text{where } R_\Psi &= \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) \\ \sin(\Psi) & \cos(\Psi) \end{bmatrix}. \end{aligned} \tag{7}$$

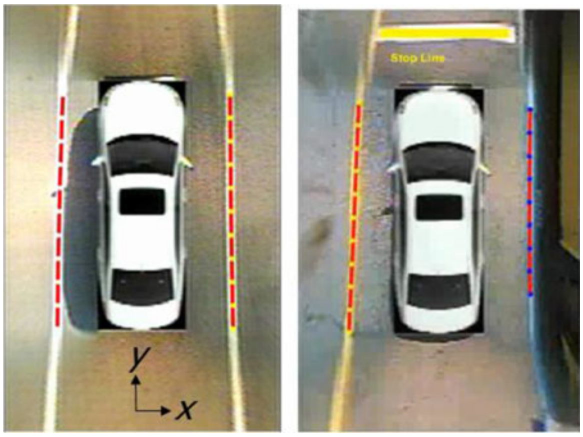


Fig. 6. Different shape of matching data: corridor (left) and U-shape (right).

B. Matching Covariance Estimation

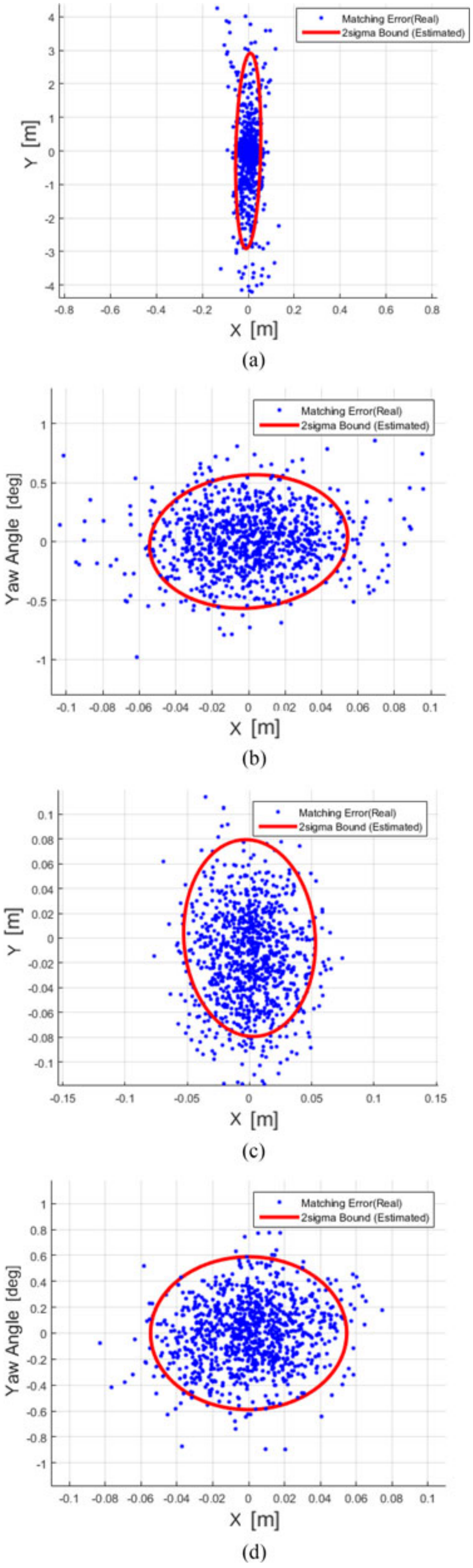
Although the ICP algorithm provides a very good estimate for correcting the vehicle position, it does not consider its uncertainty. Calculating the covariance of ICP is essential when it has to be fused with other measurements in a stochastic localization framework. For estimating the ICP's covariance, we use Haralick's method [17]. This method is summarized in the following proposition.

Proposition 1: Let Z be the input/measurements and χ be the output of an algorithm A that operates on minimizing an objective function J, i.e., $\chi^* = A(Z) = \operatorname{argmin}_\chi J(Z, \chi)$. Then, the approximate value of the covariance of χ will be

$$\operatorname{cov}(\chi^*) \approx \left(\frac{\partial^2 J}{\partial \chi^2}\right)^{-1} \frac{\partial^2 J}{\partial Z \partial \chi} \operatorname{cov}(Z) \frac{\partial^2 J}{\partial Z \partial \chi}^T \left(\frac{\partial^2 J}{\partial \chi^2}\right)^{-1}. \tag{8}$$

In this paper, χ corresponds to the matching result(r, t_x, t_y) and Z corresponds to the points p_i obtained by the lane detection algorithm and the points q_i of the lane map. Thus, cov (Z) is to be set in accordance with the noise of the AVM camera and the lane map. The detailed equation for $\operatorname{cov}(\chi^*)$ can be found in [38].

Monte Carlo simulations were conducted to verify the performance of Haralick's method for different shapes of matching data, as shown in Fig. 6. The true χ is fixed and a noise-corrupted Z is created to reflect noise characteristics of an AVM and a digital map. Using the noise-corrupted Z, our map-matching algorithm estimates χ^* and the true error of matching $(\chi^* - \chi)$ is calculated. True error samples obtained by repeating the above process 500 times are represented by blue dots in Fig. 7. Red solid lines in Fig. 7 show the two-sigma bound of $\operatorname{cov}(\chi^*)$ estimated by Haralick's method. When the ICP uses the corridor shape data, it can only correct the errors in *x* and Ψ, and not in *y*. For this reason, the variance in *y* should theoretically be infinite. The Haralick's method captures this well and shows a large uncertainty in *y* as shown in Fig. 7(a). In the case of matching U-shape data, it can fully determine translation and rotation. It can also be said that the uncertainty in *x* and *y* should be similar.



(a)

(b)

(c)

(d)

Fig. 7. Estimated standard deviation of matching error (red line) and true error obtained by Monte Carlo simulation (blue dot): (a) X-Y error (corridor shape matching), (b) X-Yaw error (corridor shape matching), (c) X-Y error (U-shape matching), and (d) X-Yaw error (U-shape matching).

TABLE I
COMPARING STANDARD DEVIATION OF MAP-MATCHING

	Corridor		U-shape	
Matching Shape	Haralick	True	Haralick	True
$\sigma_x\,(m)$	0.026	0.039	0.027	0.023
$\sigma_y\,(m)$	1.588	1.337	0.036	0.039
$\sigma_\Psi\,(\,^\circ\,)$	0.296	0.271	0.294	0.258

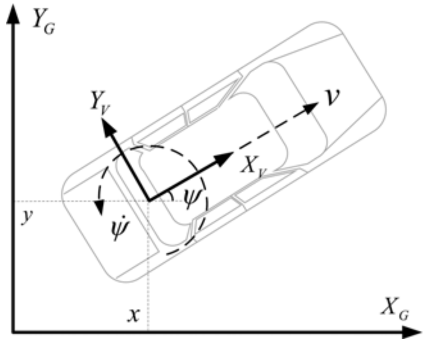


Fig. 8. Prediction vehicle model.

This method also captures this well, as shown in Fig. 7(c). The standard deviations obtained by the true error and Haralick's method are summarized below in Table I.

V. LOCALIZATION FILTER

In this section, we present the key stages of the localization system. As show in Section IV-B, the result of map-matching inevitably contains some errors, due to the shape of data used for matching as well as the noise of sensors. In order to get higher accuracy, the map-matching result should be fused with proprioceptive sensor data [39]. The measurement of the vehicle position calculated by ICP is used as an observation inside a Kalman filter framework. We also set up a validation gate in the localization filter to solve the problem of false matching.

A. Extended Kalman Filter

The ego-vehicle is described by means of a point position (x, y) and orientation (Ψ) in the global coordinate system, shown in Fig. 8.

The state vector is then given by

$$X = [x \;\; y \;\; \Psi]^T. \tag{9}$$

The basic framework for the EKF involves estimation of the state of a discrete-time nonlinear dynamic system, shown below

$$\begin{aligned} X_k &= f(X_{k-1}, \mathrm{U}_k) + w_k \\ Z_k &= h(X_{k-1}) + v_k \end{aligned} \tag{10}$$

where U_k is the known external input (velocity and yaw-rate) and Z_k is the corrected vehicle position by the map-matching process. The process noise and measurement noise are given by w_k and v_k, respectively. The measurement noise is estimated by the method described in Section IV-B. The process noise is associated with proprioceptive sensors. It is assumed to have zero mean and a Gaussian distribution. The specifications of the proprioceptive sensors are given in Table II.

TABLE II
SPECIFICATION OF PROPRIOCEPTIVE SENSORS

Sensor	Range	Resolution	Noise (RMS)	Unit
Yaw rate	±120	0.0625	0.5	deg/s
Wheel speed	0−130	0.035	0.3	m/s

By integrating using the Euler approximation and assuming that the control signals, the velocity and yaw-rate are approximately constant over the sample period, the nominal discrete process model equations can be written as

$$\bar{X}_{k|k-1} = \begin{bmatrix} \hat{x}_{k-1|k-1} + v_k \cdot \Delta t \cdot \cos(\hat{\Psi}_{k-1|k-1} + \triangle\, t \cdot \dot{\Psi}_k) \\ \hat{y}_{k-1|k-1} + v_k \cdot \Delta t \cdot \sin(\hat{\Psi}_{k-1|k-1} + \triangle\, t \cdot \dot{\Psi}_k) \\ \hat{\Psi}_{k-1|k-1} + \Delta t \cdot \dot{\Psi}_k \end{bmatrix} \tag{11}$$

where $\hat{x}_{k-1|k-1}$, $\hat{y}_{k-1|k-1}$, $\hat{\Psi}_{k-1|k-1}$ are estimated states from the previous time step, and *k* is the time index of the discrete model. The covariance of the predicted state is described as

$$\begin{aligned} &P_{k|k-1} = F_k P_{k-1|k-1} F_k + G_k Q G_k \\ &\text{where} \;\; F_k = \frac{\partial f}{\partial x}\Big|_{\hat{X}_{k-1|k-1}, u_k}, G_k = \frac{\partial f}{\partial u}\Big|_{\hat{X}_{k-1|k-1}, u_k}, \end{aligned} \tag{12}$$

where *Q* describes the covariance matrix related to the proprioceptive sensor's noise.

Vehicle positions are corrected by a measurement update of the EKF as follows:

$$\begin{aligned} &\hat{X}_{k|k} = \bar{X}_{k|k-1} + K_k \cdot \left(Z_k - H \cdot \bar{X}_{k|k-1}\right) \\ &\begin{aligned} K_k &= P_{k|k-1} H^T \cdot S_k^{-1} \\ S_k &= H P_{k|k-1} H^T + R_k \end{aligned} \quad \text{where } H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{13}$$

where R_k describes the covariance matrix of the measurement.

The covariance of the estimated state is described as

$$P_{k|k} = (I - k_k H) \cdot P_{k|k-1}. \tag{14}$$

The rate of time and measurement update is 10 Hz.

B. Validation Gate

A validation gate is set up to prevent fusing the result of false matching. The validation gate represents a threshold that is associated with the acceptability of the measurements. Only measurements inside of the validation gate are used to update the filter [40]. The validation gate can be obtained as follows:

$$\begin{aligned} e^2 &= (Z_k - H\bar{X}_{k|k-1})^T S_k^{-1} (Z_k - H\bar{X}_{k|k-1}) \\ V_k &= \left\{Z : e^2 < g^2\right\}, \end{aligned} \tag{15}$$

where g^2 is chosen as a confidence level. A confidence level is generally chosen between 1 and 3. In this paper, we set this value to 3. A three-sigma gate is commonly used, ensuring that

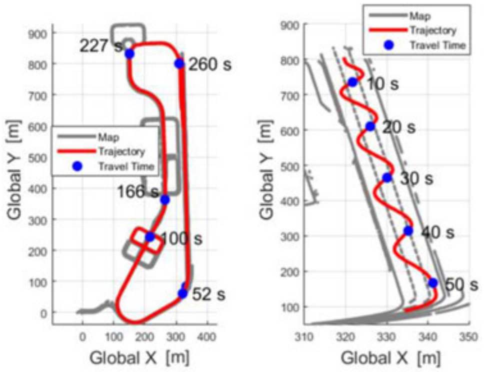
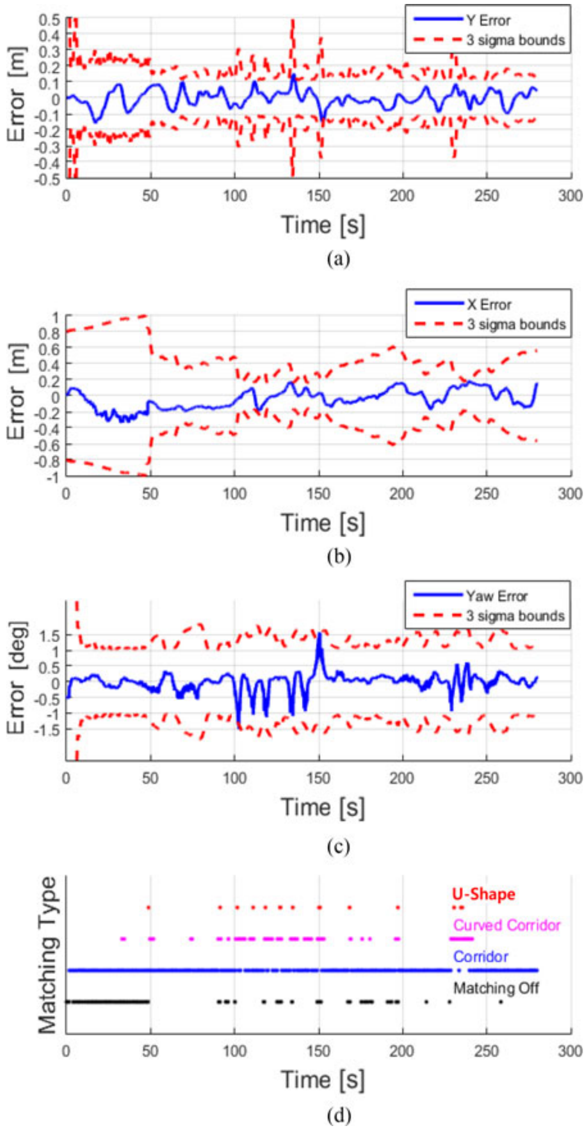


Fig. 9. Detailed trajectory and travel time of two typical scenarios.



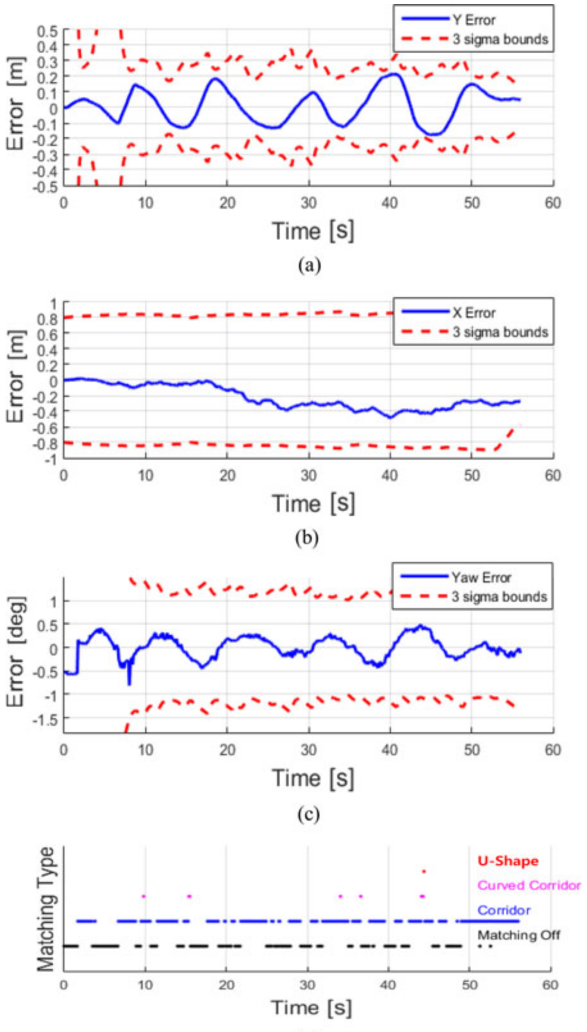
(a)

(b)

(c)

(d)

Fig. 10. Localization result of first scenario: (a) lateral position error, (b) longitudinal position error, (c) Yaw angle error, and (d) map-matching history.



(a)

(b)

(c)

(d)

Fig. 11. Localization result of second scenario: (a) lateral position error, (b) longitudinal position error, (c) Yaw angle error, and (d) map-matching history.

the measurement will fall in the gate with a probability of 0.998 under the Gaussian assumption. The normalized error e^2 varies as a Chi-squared distribution with the number of measurement degrees of freedom

VI. Test Results

In order to validate the proposed algorithm, tests have been carried out at the Korea Automobile Testing and Research Institute proving ground. The test track replicates a real urban environment that includes intersecting streets, pedestrian crosswalks, and traffic signals as shown in Fig. 9. Since the lane-level map was generated using RTK-DGPS, the accuracy of the digital road map in an absolute coordinate system was in the order of centimeters. For real-time implementation, a lane-level localization algorithm was built with the LabVIEW software installed on the computer. A test driver manually drove the test vehicle along 15 different trajectories. In keeping with general urban driving conditions, the speed and acceleration of the test vehicle were restricted to 60 km/h and 2 m/s^2. The total distance and

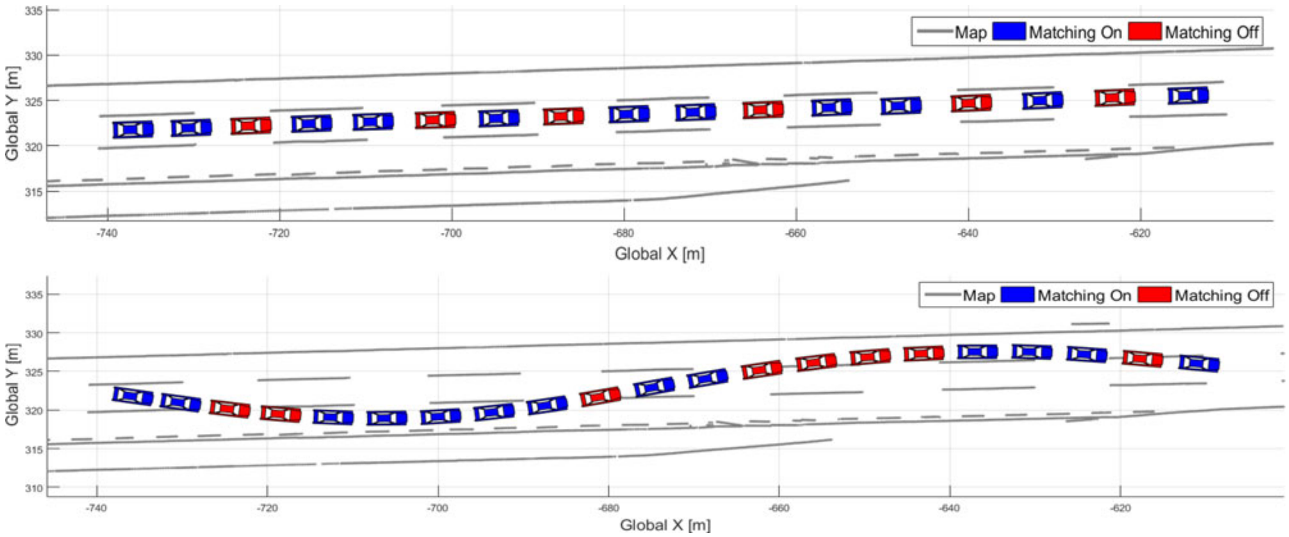


Fig. 12. Operation of map-matching according to the driving maneuvers.

driving time of the tests were about 16.1 km and 32 min, respectively. All datasets have in common that ground truth data are obtained by RTK-DGPS.

Figs. 10 and 11 represent the result of two typical test scenarios. The first scenario is lane keeping in the various environments such as intersection, crosswalk, roundabout, and merging/splitting roads. The second scenario is consecutive lane changing. Most previous works validated their localization algorithm only under the lane keeping condition. However, lane changes occur frequently on urban roads. Therefore, it is necessary to validate the proposed algorithm under the lane change condition. Through the results of the two tests, we can verify that the proposed localization algorithm is suitable for automated driving in urban environments. The detailed trajectory and travel time of the two tests are represented in Fig. 9.

Figs. 10 and 11(a), (b), and (c) show changes of localization errors with respect to the vehicle coordinates. To show how the filter is well tuned, $\pm 3\sigma$ bounds are plotted as red-dashed lines. In both cases, the lateral error is smaller than the longitudinal error because the most frequently used shape for map-matching is a corridor shape. Figs. 10 and 11(d) represent the history of shapes used to matching the map. As mentioned in Section IV-B, the covariance of map-matching in the longitudinal direction is much larger than that in the lateral direction in the case of corridor shape matching. The confidence of the lateral position is much higher than that of the longitudinal position in the overall tests.

Actually, the longitudinal error is not important to control an automated vehicle for keeping the lane on small-curvature roads. However, for roads with large curvature (especially, in intersections), the longitudinal error should be decreased to the same level as that of the lateral error. In most cases, there are stop lines or merge lines that enable a U-shape matching before entering a road with large curvature. The longitudinal error and variance significantly decrease whenever a U-shape matching occurs. Thus, it can be said that the proposed algorithm has a sufficient localization performance to control an automated vehicle on the proving ground. Fig. 13 shows the situations where U-shape matchings occurred during the first scenario.

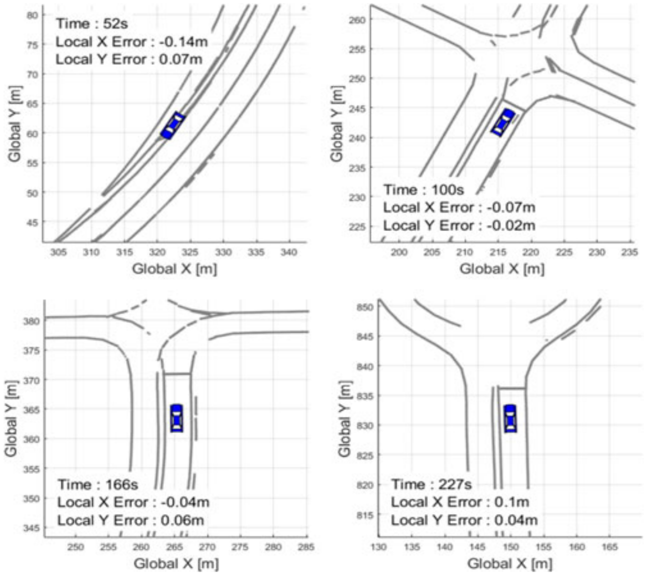
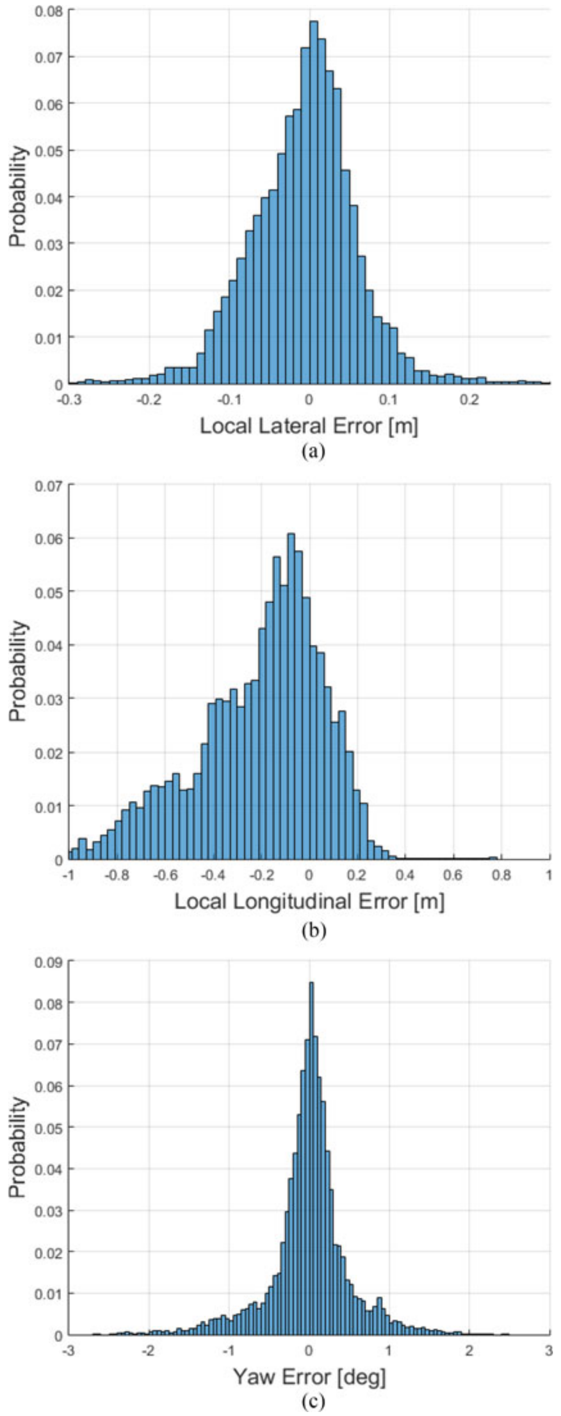


Fig. 13. Localization result of second scenario (dataset 4).

The lateral error of the second scenario is slightly bigger than that of the first. This is caused by the lack of visual measurement updates. Since an AVM camera has a small FOV, lane data used for map-matching are not sufficient when the test vehicle changes lane. This is depicted in Fig. 12. However, in spite of consecutive lane changing, the mean errors of the lateral direction are also small (less than 20 cm) enough to control the automated vehicle.

Fig. 14 shows the histogram of the measurement residuals over all datasets. The medians of the lateral position and yaw error are, respectively, less than 20 cm and 1°. The mean errors in the longitudinal direction are much larger than that in the lateral. However, as mentioned earlier, the longitudinal position error



(a)

(b)

(c)

Fig. 14. Histogram of local longitudinal/lateral position and yaw angle error of total data: (a) lateral error distribution, (b) lateral error distribution, and (c) lateral error distribution.

decreases to the same level as that of the lateral error before entering a road with large curvature. So, it does not matter to the control of the automated vehicle. From these results, it was confirmed that our localization method has applicability to automated driving in an urban environment.

TABLE III
ERROR STATICS

	Lateral PE (m)		Longitudinal PE (m)	
	I	II	I	II
mean	0.26	0.072	0.39	0.26
std. dev	0.34	0.067	0.39	0.23

(PE: positioning error; I: front camera-based localization; II: AVM-based localization).

We compared our algorithm with another localization method based on the front camera [3]. Table III gives performance metrics of both localization methods. Since the two results were not obtained under the same conditions, the results cannot be compared directly. Even then, it is obvious that lateral accuracy is significantly improved by the proposed algorithm. The major factor that brings about these results is the AVM's characteristics mentioned in Section II-B. AVM images enable direct calculation of the lateral offset and make it possible to improve the lateral accuracy through map-matching.

VII. Conclusion

A novel method for lane-level localization using AVM cameras with a lane map for application to automated driving has been presented. The proposed algorithm consists of three parts: lane detection, position correction, and localization filter. In lane detection, a well-known lane detection algorithm is adapted to be applicable to the commercial AVM module. To correct the vehicle position using detected lane and a digital map, an ICP-based map-matching algorithm is used, and an EKF is applied to fuse the map-matching result with the vehicle sensors' data. In order to improve the reliability of the proposed localization algorithm, a covariance estimator for ICP and a validation gate are designed in the localization filter. The lane-level localization algorithm has been successfully implemented on a test vehicle. Tests have been conducted on a proving ground. Vehicle test results have revealed that a precision within a few centimeters can be achieved, which is sufficient for automated vehicle control. However, there still remain challenging driving situations in terms of lane-level localization on urban roads. Such situations include low-visibility conditions of the lane, lack of longitudinal position correction over a long period, increasing nonlinearity due to unusual maneuvering, and so on. Enhancing and verifying the proposed algorithm to achieve good accuracy for automated vehicle control in challenging driving situations on urban roads are the topics of our future research.

References

[1] I. Skog and P. Händel, "In-car positioning and navigation technologies—A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 4–21, Feb. 2009.

[2] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2013, pp. 449–454.

[3] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Lane marking aided vehicle localization," in *Proc. IEEE 16th Int. Conf. Intell. Transp. Syst.*, 2013, pp. 1509–1515.

[4] D. Gruyer, R. Belaroussi, and M. Revilloud, "Map-aided localization with lateral perception," in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 674–680.

[5] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, “Recent progress in road and lane detection: A survey,” *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, 2014.

[6] N. Suganuma and T. Uozumi, “Precise position estimation of autonomous vehicle based on map-matching,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, 2011, pp. 296–301.

[7] D. Cui, J. Xue, S. Du, and N. Zheng “Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 4958–4964.

[8] A. Y. Hata, F. S. Osorio, and D. F. Wolf, “Robust curb detection and vehicle localization in urban environments,” in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 1257–1262.

[9] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, “Synthetic 2D LIDAR for precise vehicle localization in 3D urban environment,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1554–1559.

[10] R. Zlot and M. Bosse, “Place recognition using keypoint similarities in 2D lidar maps,” in *Experimental Robotics*. Berlin, Germany: Springer, 2009.

[11] K. Wang, Y.-H. Liu, and L. Li, “A simple and parallel algorithm for real-time robot localization by fusing monocular vision and odometry/AHRS sensors,” *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1447–1457, Jan. 2014.

[12] L. Wang, P. D. Groves, and M. K. Ziebart, “GNSS shadow matching: Improving urban positioning accuracy using a 3D city model with optimized visibility scoring scheme,” *Navigation*, vol. 60, no. 3, pp. 195–207, 2013.

[13] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” in *Robot.-DL Tentative. Int. Soc. Opt. Photon.*, pp. 586–606, Apr. 1992.

[14] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” presented at the *3rd Int. Conf. 3-D Digital Imag. Model.*, Quebec City, QC, Canada, 2001.

[15] S.-Y. Park and M. Subbarao, “An accurate and fast point-to-plane registration technique,” *Pattern Recog. Lett.*, vol. 24, no. 16, pp. 2967–2976, 2003.

[16] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing ICP variants on real-world data sets,” *Auton. Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[17] R. Haralick, “Propagating covariance in computer vision,” in *Performance Characterization in Computer Vision*, vol. 1, New York, NY, USA: Springer, 1998, pp. 95–115.

[18] A. Mallios, P. Ridao, D. Ribas, and E. Hernández, “Scan matching SLAM in underwater environments,” *Auton. Robots*, vol. 36, no. 3, pp. 181–198, 2014.

[19] A. Censi, “An accurate closed-form estimate of ICP’s covariance,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3167–3172.

[20] L. Marin, M. Valles, A. Soriano, A. Valera, and P. Albertos, “Event-based localization in Ackermann steering limited resource mobile robots,” *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1171–1182, Aug. 2013.

[21] T. Lefebvre, H. Bruyninckx, and J. D. Schutter, “Kalman filters for nonlinear systems: A comparison of performance,” *Int. J. Control*, vol. 77, no. 7, pp. 639–653, 2004.

[22] M. Norgaard, N. K. Poulsen, and O. Ravn, “Advances in derivative-free state estimation for nonlinear systems,” Tech. Univ. Denmark, Kongens Lyngby, Denmark, Tech. Rep. IMM-REP-1998-15, 2000.

[23] R. Toledo-Moreo, M. Zamora-Izquierdo, B. Ubeda-Miarro, and A. Gomez-Skarmeta, “High-integrity IMM-EKF-based road vehicle navigation with low-cost GPS/SBAS/INS,” *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 491–511, Sep. 2007.

[24] A. N. Ndjeng, D. Gruyer, S. Glaser, and A. Lambert, “Low cost IMU-odometer-GPS ego localization with IMM for unusual maneuvers,” *Inf. Fusion*, vol. 12, no. 4, pp. 264–274, 2011.

[25] S.-B. Han, J.-H. Kim, and H. Myung, “Landmark-based particle localization algorithm for mobile robots with a fish-eye vision system,” *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1745–1756, Aug. 2012.

[26] M. Rhudy and Y. Gu, “Understanding nonlinear Kalman filters part 1: Selection of EKF or UKF,” in *Interactive Robot. Lett.*, West Virginia University: Morgantown, WV, USA, 2013, pp. 1–9.

[27] F. Martinelli, “Robot localization: Comparable performance of EKF and UKF in some interesting indoor settings,” in *Proc. IEEE 16th Mediterranean Conf. Control Autom.*, 2008, pp. 499–504.

[28] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, “Mining GPS traces for map refinement,” *Data Mining Knowledge Discovery*, vol. 9, no. 1, pp. 59–87, 2004.

[29] K. Heong-Tae, S. Bongsob, and K. Yeonsik, “Lane-level localization based on situational awareness on highway,” in *Proc. IEEE 6th Int. Conf. Ubiquitous Future Netw.*, 2014, pp. 3–6.

[30] Y.-L. Chang, L.-Y. Hsu, and O. T-C. Chen, “Auto-calibration around-view monitoring system,” in *Proc. IEEE 78th Veh. Technol. Conf.*, 2013, pp. 1–5.

[31] M. Aly, “Real time detection of lane markers in urban streets,” in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 7–12.

[32] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Englewood Cliffs, NJ, USA: Prentice Hall, 2002.

[33] L. Xu *et al.*, “Shadow detection and removal in real images: A survey,” Comput. Vis. Lab, Dept. Comput. Sci. Eng., Shanghai Jiao Tong Univ., Shanghai, China, 2006.

[34] K. K. Singh, K. Pal, and M. J. Nigam, “Shadow detection and removal from remote sensing images using NDI and morphological operators,” *Int. J. Comput. Appl.*, vol. 42, no. 10, pp. 37–40, 2012.

[35] D. Kim, T. Chung, and K. Yi, “Lane map building and localization for automated driving using 2D laser rangefinder,” presented at the Intell. Veh. Symp., Seoul, Korea, Jun. 2015, pp. 680–685.

[36] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 1991, pp. 2724–2729.

[37] H. Fang, M. Yang, and R. Yang, “Ground texture matching based global localization for intelligent vehicles in urban environment,” in *Proc. IEEE Intell. Veh. Symp.*, 2007, pp. 105–110.

[38] P. S. Manoj *et al.*, “A closed-form estimate of 3D ICP covariance,” in *Proc. 14th IAPR Int. Conf. Mach. Vis. Appl.*, 2015, pp. 526–529.

[39] R. Madhavan, M. W. M. G. Dissanayake, and H. F. Durrant-Whyte, “Autonomous underground navigation of an LHD using a combined ICP-EKF approach,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, vol. 4, pp. 3703–3708.

[40] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York, NY, USA: Wiley, 1987.

[41] J. Seo and K. Yi, “Robust mode predictive control for lane change of automated driving vehicles,” SAE Tech. Paper 2015–01–0317, 2015.

[42] S. Li and Y. Shimomura, “Lane marking detection by side fisheye camera,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 606–611.

[43] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, pp. 23–27, 1975.

[44] M. St-Pierre and D. Gingras, “Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system,” in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 831–835.

[45] [Online]. Available: http:\\www.omnivue.co.kr\

Dongwook Kim received the B.S. degree in mechanical and aerospace engineering from Seoul National University, Seoul, Korea, in 2008, where he is currently working toward the Ph.D. degree in mechanical and aerospace engineering.

His research interests include control systems, driver assistant systems, and autonomous vehicle control.

Beomjun Kim received the B.S. degree in mechanical and aerospace engineering from Seoul National University, Seoul, Korea, in 2011, where he is currently working toward the Ph.D. degree in mechanical and aerospace engineering.

His research interests include control systems, driver assistant systems, autonomous vehicle control, and active safety systems of a ground vehicle.

Taeyoung Chung received the B.S. degree in mechanical engineering from Kookmin University, Seoul, Korea, in 1996, and the M.S. and Ph.D. degrees in automotive engineering from Hanyang University, Seoul, in 1998 and 2005, respectively.

Since September 2005, he has been a Postdoctoral Researcher with the University of California at Berkeley. He is also with the Smart Car Research Team, Hyundai Mobis, Uiwang-si, Korea. His research interests include lateral and roll stability control, autonomous vehicle control, and active safety systems of a ground vehicle.

Kyongsu Yi (M'12) received the B.S. and M.S. degrees in mechanical engineering from Seoul National University, Seoul, Korea, in 1985 and 1987, respectively, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, Berkeley, CA, USA, in 1992.

He is a currently a Professor at the School of Mechanical and Aerospace Engineering, Seoul National University. He currently serves as a member of the editorial boards of the Korean Society of Mechanical Engineers, International Journal of Automotive Technology, and Institute of Control, Robotics and Systems journals. His research interests include control systems, driver assistant systems, and active safety systems of a ground vehicle.