

Probabilistic lane detection and lane tracking for autonomous vehicles using a cascade particle filter

Minchae Lee¹, Chulhoon Jang² and Myoungho Sunwoo³

Abstract

This paper proposes a robust lane detection algorithm with a cascade particle filter that incorporates a model decomposition approach. Despite the sophisticated tracking mechanism of a particle filter, the conventional particle-filter-based lane detection system suffers from an estimation accuracy problem and a high computational load. In order to improve the robustness and the computation time for lane detection systems, the proposed cascade particle filter decomposes a lane model into two submodels: a straight model and a curve model. By dividing the lane model, not only can the computation time be decreased, but also the accuracy of the lane state estimation system can be increased. The proposed lane detection algorithm and the cascade particle filter were evaluated on various roads and environmental conditions with the autonomous vehicle A1, which was the winner of the 2010 and 2012 Autonomous Vehicle Competition in the Republic of Korea organized by the Hyundai motor group. The proposed algorithm proved to be sufficiently robust and fast to be applied to autonomous vehicles as well as to intelligent vehicles for improving the vehicle safety.

Keywords

Lane detection, cascade particle filtering, tracking, autonomous vehicle

Date received: 25 January 2014; accepted: 11 November 2014

Introduction

According to research by the National Highway Traffic Safety Administration (NHTSA), most traffic accidents are caused by human faults such as distractions, drowsiness or excessive speed.¹ To protect against human faults, advanced driver assistance systems (ADASs) have been increasingly researched for intelligent vehicles. Furthermore, as perception technologies have been improved, autonomous vehicles, which are the ultimate goal of intelligent vehicles, are expected to be realized in the near future. For autonomous driving systems, perception technologies play a significant role. In particular, vision systems which use cameras have been widely studied for positioning and object recognition such as vehicles, lanes, traffic signs and pedestrians. In terms of the positioning system, lane detection is a type of local positioning system. Although advanced Global Positioning Systems (GPSs) such as the Differential Global Positioning System and the Real-Time Kinematic Global Positioning System provide a precise global position, GPSs are still not sufficiently robust in urban areas owing to satellite outage conditions and multi-path problems.² Consequently,

lane detection is an essential system for ego-vehicle positioning. Such systems have proven to be sufficiently reliable for highways.^{3–6} For ADASs, lane detection systems have already been commercialized such as lane departure warning systems and lane-keeping assistance systems.⁷ For autonomous vehicles, however, lane detection systems are not yet robust, especially in complex urban environments.^{8,9} Although the vehicle position and the heading can be acquired by using only the camera and existing infrastructures, it is limited to structured roads. In urban areas, lane detection systems must tackle many different scenarios such as parked

¹Global R&D Center, Mando, Seongnam-si, Gyeonggi-do, Republic of Korea

²Department of Automotive Engineering, Graduate School, Hanyang University, Seoul, Republic of Korea

³Department of Automotive Engineering, Hanyang University, Seoul, Republic of Korea

Corresponding author:

Myoungho Sunwoo, Department of Automotive Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 133-791, Republic of Korea.

Email: msunwoo@hanyang.ac.kr

and moving vehicles, higher-curvature roads, intersections, pedestrians and crosswalks.

Lane detection systems mainly utilize two research areas: image processing and state estimation. Image-processing techniques are used initially to detect lane markings. Then, using this information, the lateral position of the vehicle is estimated through filtering algorithms. To detect lane markings, Bertozzi and Broggi¹⁰ presented a lane detection system based on a horizontal dark–light–dark (DLD) pattern analysis and an inverse perspective mapping (IPM) method. IPM is performed to remove the perspective effect of a camera under a flat-road assumption. Single-lane and double-lane marking features are detected using the DLD and the dark–light–dark–light–dark patterns. The DLD-pattern-based lane detection system is implemented into the General Obstacle and Lane Detection (GOLD) framework^{10,11} and has been validated through the Defense Advanced Research Projects Agency Urban Challenge,¹² the VisLab Intercontinental Autonomous Challenge (VIAC)⁶ and Hyundai's Autonomous Vehicle Competition.¹³ However, it has been argued that the detected lane position is not sufficiently reliable in the case of disturbances of the road environment. In particular, difficulties arise if lane markings are poor or missing. To increase the robustness of the estimated position, Sehestedt et al.¹⁴ and Jiang et al.¹⁵ advanced the DLD-pattern-based and IPM-based lane detection methods by applying filtering algorithms.

In the context of feature detection, lane markings are simple to detect in a highway environment. However, because of the uncertainty in the road environment, robustness is difficult to achieve without a filtering algorithm. Accordingly, filtering is essential for estimating the lane states from noisy measurements. There are several filtering algorithms for state estimation including the Kalman filter,¹⁶ the extended Kalman filter,¹⁷ the random sample consensus (RANSAC)¹⁸ and the particle filter.¹⁹ A linear filter such as the Kalman filter estimates the system states by using measurements and linear dynamic models under Gaussian assumptions. However, because of the non-linear and non-Gaussian noise of an image, the Kalman filter cannot be easily applied to a vision system.²⁰ In general, two-dimensional image data cannot be directly used for the Kalman filter owing to its measurement structure. To relate the image data to the measurement structure of the Kalman filter, the region-of-interest (ROI) tracking approach is employed for the Kalman-filter-based lane detection system.²¹ The ROI tracking approach has the advantage of reducing the image-processing region and eliminating the irrelevant features. However, ROI initialization requires the initial position of the lane. If the road is not continuous, ROI initialization should be repeated to recover the track loss. In addition, ROI initialization requires a relatively high computational load to meet the robustness of the initial estimation. As a result, the initialization process must spend several image frames and use

complex image-processing algorithms to ensure robustness.

In order to overcome the non-linear and non-Gaussian noise of an image, non-linear filtering algorithms can be applied to lane detection systems. In most vision applications, however, the optimal non-linear filters are difficult to implement practically owing to the complexity and the computational load. Alternatively, suboptimal non-linear filters, such as the extended Kalman filter, the unscented Kalman filter and the particle filter, which are approximations of non-linear functions in the state dynamic and measurement models, are solutions. In particular, particle filters have been proven to be robust for lane detection and lane-tracking systems compared with other suboptimal filters.²⁰

In this study, a cascade particle filter (CPF) is proposed on the basis of a particle filter to achieve robust lane detection in difficult scenarios. This paper contains two contributions. First, we propose a CPF for state estimation and tracking applications. The proposed CPF combines the advantage of multiple models and conventional particle filters. Second, we develop a new lane detection system using the CPF. We show that the proposed architecture of the CPF and the lane detection algorithm increase the robustness and the accuracy of the lane state estimation as shown by various experimental conditions.

The rest of this paper is organized as follows. In the second section, the background information of the CPF is described. In the third section, the CPF-based lane detection and lane-tracking algorithms are explained. In the fourth section, the experimental results are presented for highway, urban and country road conditions. Conclusions are discussed in the fifth section.

Cascade particle filtering

Particle filtering

Particle filters are suboptimal filters that perform Bayesian estimation based on a set of random states that represent the probability densities.^{22,23} Particle filters do not rely on a fixed functional form to describe the probability densities. Instead, they approximate the probability densities by a set of random samples which are called particles. The sample representation approach of particle filters originates from Monte Carlo (MC) integration, which is a numerical method for an approximate evaluation using random samples. MC integration uses an importance sampling method to decrease the required number of samples and estimation error by applying random samples of reduced variance.

In particle filtering, a state vector at time t is denoted as a set of particles according to

$$\mathbf{x}_t = x_t^1, x_t^2, \dots, x_t^N \quad (1)$$

Each particle x_t^k (with $1 \leq k \leq N$) is a type of hypothesis for representing the probability densities. N denotes the number of particles of a state x_t . The number of particles plays a significant role in determining the performance of particle filters. In the case of the Kalman filter, the number N of particles is due to the Kalman filter for which the estimates are based on only one state using a single input and a single output. However, for particle filters, calculations are made for N possible states using multiple hypotheses. As a result, for an increasing order of the system, particle filters suffer from increasing computation because of the increasing required number of particles. Moreover, the number of particles can be limited by considering the hardware performance. Consequently, the tracking and state estimation performance can be limited by the number of particles in order to implement importance sampling for real-time systems.

Cascade particle filtering

In this study, a CPF is proposed to solve real-time and implementation problems by applying a model decomposition method. The CPF is designed to reduce the required number of particles to increase the computational efficiency while preserving the state estimation performance for real-time applications.

The approximate system of the proposed CPF models by decomposing the states. There are several approaches to solving model approximation problems such as an interacting multiple-model estimator and a multiple-model-based state estimator. In particular, a multiple-model-based particle filter (MMPF) method is the base model of the CPF. The MMPF improves the estimation accuracy by augmenting continuous target state vectors and discrete regime variables.²² A regime variable r_k augments the state vector y_k to specify the distinct model for each particle as

$$y_k = [x_k^T \quad r_k]^T \quad (2)$$

where $r_k \in S = \{1, 2, \dots, s\}$. The regime variables are designed by a time-homogeneous Markov chain with transitional probabilities according to

$$\pi_{i,j} = P\{r_k = j | r_{k-1} = i\}d \quad (3)$$

where $i, j \in S = \{1, 2, \dots, s\}$. However, as the states of the multiple models augment the state vector, the dimension of the state space is increased. Sample impoverishment, which is a classical problem of the particle filter, can be made worse by the regime variable. The sample impoverishment is a phenomenon of duplicated sampling and the lack of a number of samples. If the dimension of the state space is increased, the total number of cases to be examined should be increased. Consequently, the samples can be biased and overlapped owing to the implementation limitations on real-time hardware. Moreover, measurement

likelihoods can be contaminated by other measurement noises.

The concept of the CPF is that it is a series of particle filters. From the viewpoint of the state-space equations, the CPF is structured by dividing the independent states of a model into multiple subsets of states. Commonly, particle filters use states x_k and importance weights w_k . In this study, π_k and Π_k are used to represent multiple sets of states. As shown in

$$\{x_k, w_k\} = \pi_k \quad (4)$$

π_k represents a set of states and importance weights by combining x_k and w_k . In general, conventional particle filters use x_k and w_k . However, the CPF decomposes π_k as

$$\begin{aligned} \{\pi_k^1, \pi_k^2, \dots, \pi_k^S\} &= \{\pi_k^i\}_{i=1}^S \\ &= \Pi_k^{1,S} \end{aligned} \quad (5)$$

With the set of states and importance weights π_k at time k , multiple subset of states can be defined as

$$\Pi_k^{1,S} = \Pi_k^{1,s-1} + \Pi_k^{s,S} \quad (6)$$

$$\Pi_k^{1,s-1} = \{\pi_k^i\}_{i=1}^{s-1} \quad (7)$$

$$\Pi_k^{s,S} = \{\pi_k^i\}_{i=s}^S \quad (8)$$

With multiple subsets of states Π_k , the CPF can be represented as follows in algorithm 1. The major difference between the particle filter and the CPF is the sequential importance sampling (SIS) which is computed independently with a given subset of each stage.

As shown in the CPF algorithm, the CPF uses SIS for calculating the states and its weights. Through SIS, probability calculation is carried out in multiple stages. The idea is to divide the state variables into multiple stages. Consequently, the parameters can be estimated independently, and stage-to-stage parameter passing is required. The following algorithm 2 describes the two-stage SIS algorithm.

The design approach, namely the two-stage SIS algorithm, significantly improves the computational complexity such as an inversion of the matrix. As a result, the order of complexity can be significantly decreased according to

$$O(m_1^k) + O(m_2^k) + \dots + O(m_x^k) < O(n^k) \quad (9)$$

Algorithm 1. The CPF algorithm is as follows.

[$\mathbf{x}_k, \mathbf{w}_k$] = Cascade Particle Filter[$\mathbf{x}_{k-1}, \mathbf{w}_{k-1}, \mathbf{z}_k$]

1. For $s = 1 : S$,
 2. $\Pi_k^s = \text{SIS}[\Pi_k^{1,s-1}, \Pi_{k-1}^{s,S}, \mathbf{z}_k]$.
 3. Calculate \hat{N}_{eff} .
 4. If $\hat{N}_{\text{eff}} < N_{\text{thr}}$,
 5. $\Pi_k^s = \text{resample}[\Pi_k^s]$.
 6. Endif.
 7. Endfor.
-

Algorithm 2. The SIS algorithm is as follows.

$$\Pi_k^s = \text{SIS}[\Pi_k^{1:s-1}, \Pi_{k-1}^{s,S}, z_k]$$

First stage

1. For $i = 1 : s - 1$,
2. $P(y_k^i)$ = generate the probability of y_k^i using x_k^i and system dynamics.
3. If $P(y_k^i) < P_{thr}$,
4. store x_k^i as $x_{k|k}^i$ and increment j .
5. Endif.
6. Endfor.
7. Calculate w_k using $P(y_k^{1:s-1})$.
8. Estimate states $(x_k^{1:s-1})$ with $w_k^{1:s-1}$.

Second stage

9. For $i = s : S$,
10. $P(y_k^i)$ = generate the probability of y_k^i using x_k^i and system dynamics.
11. If $P(y_k^i) < P_{thr}$,
12. store x_k^i as $x_{k|k}^i$ and increment j .
13. Endif.
14. Endfor.
15. Calculate w_k using $P(y_k^{s:S})$.
16. Generate $\Pi_k^{s,S}$ using stored x_k^i and w_k^i .
17. Estimate states and weights (Π_k^s) with $x_k^{1:s-1}$ and $(x_k^{s:S}, w_k^{s:S})$.

where

$$\sum_{i=1}^s m_i^k = S$$

In equation (9), m_i^k is the number of states in the i th stage and S is the total number of states.

To construct the CPF, a system model should be decomposed into several simplified submodels. Figure 1 shows the concept of N -stage model decomposition. Depending on the independence of states, a model can be decomposed into several states. As an example of model decomposition, Figure 2 shows the concept of two-stage model decomposition which is used for lane detection and lane tracking. In this example, if more specific estimations are required, a residual model can also be divided into two submodels: a second-stage approximated model and a second-stage residual model. In this decomposition methodology, an approximated model and a residual model do not share states. In other words, when the probability evaluation of particles is required, the approximated model and the residual model should be merged to describe the entire systems.

A measurement model is the most important design factor in a filtering algorithm because it can be configured to various forms according to applications. Through the resampling step, the particles are evaluated with the importance weights which are generated by the measurement model. For this reason, the measurement model plays a crucial role in designing the particle filter. Likewise, the performance of the CPF is also strongly affected by the measurements. In order to design the CPF, individual measurement models have to be provided for each particle filter, as shown in Figure 3. It

is possible to share a measurement model with multiple particle filters. However, undoubtedly, the individual measurement models for each subsystem model have better performances than a shared measurement model has. In this study, each particle filter has its own measurement model and the submodels, which are the approximated model and the residual model. To increase the robustness of the tracking performance, the proceeding particle filter uses the approximated model. To track the object roughly, the proceeding particle filter can be designed with core state variables such as a lateral offset and a heading angle in lane-tracking systems. After estimating the core state variables, the following particle filter can estimate more precise state variables such as the curvature and the curvature derivative with given core state variables.

Unfortunately, the variation in the state estimation results is normally larger than other non-probabilistic approaches such as the curve-fitting-based lane detection system. The probabilistic diffusion of particles makes smoothed estimates difficult. Another problem is that the measurement quality and accuracy of lane positions decrease along the longitudinal axis of a vehicle owing to the perspective effect of the camera. Consequently, the reliable distance of the lane position measurement is limited for maintaining the robustness and the accuracy of a particle filter.

Lane detection and tracking

System architecture of the cascade particle filtering

Figure 4 shows the system architecture of the proposed lane detection and lane-tracking system. The system

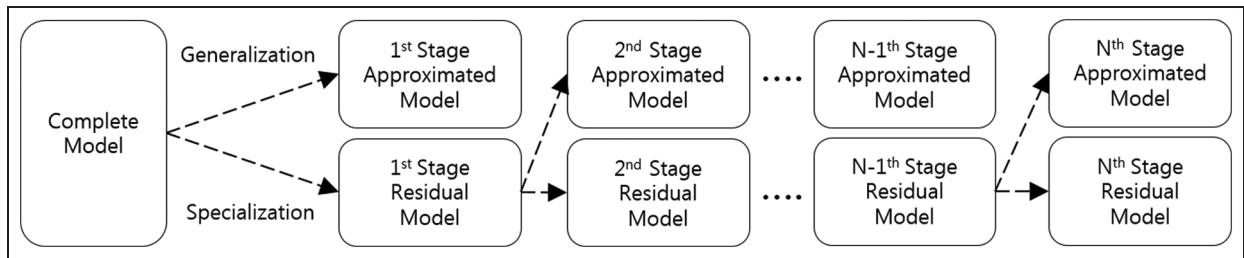


Figure 1. The concept of the N -stage model decomposition for a CPF.

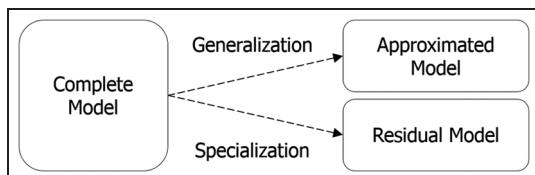


Figure 2. The concept of two-stage model decomposition for a CPF.

consists of three parts: feature extraction, the CPF and the Kalman filter. The feature extraction consists of DLD pattern detection and edge detection. The DLD pattern is used for measurement of the first-stage CPF, and the edge information is used for measurement of the second-stage CPF. The CPF is implemented into two stages, which are the approximated tracking stage and the precise tracking stage. The first stage of the particle filter estimates the lane parameters (the lateral offset, the heading angle, the lane width and the variation in the lane width) using the approximated lane model. The second stage of the particle filter estimates the precise lane parameters (the curvature and the curve transition point) using the estimated states in the first stage. Finally, the Kalman filter is used to increase the smoothness and the robustness of the state estimation.

The CPF can be implemented in several stages. In our system, the lane model is partitioned into the first-order model used in the approximated tracking stage and the curvature model used in the precise tracking stage. In the approximated tracking stage, the first-order lane model is designed to ensure the tracking performance. Although the first-order lane model does not represent the curvature of the road, the estimation results of the lateral offset and the heading angle are more stable than those in the complicated lane models. As shown in Table 1, the comparison result shows that the variance of the estimated state increases as the order of the model increases. The larger variance of the higher-order model is due to the effect of sampling methodology. Because of the number of limited samples and the hardware constraints, estimating more states with the same number of samples can easily cause a lack of accuracy.

Clothoid lane model

Although most of the road can be represented with a straight line and a circle model, a curvature discontinuity exists at the point of transition between the straight line and the circle. To ensure road safety, a clothoid is applied for the highway design to solve the track transition problem. As shown in Figure 5, a clothoid is a curve where the radius of the curvature decreases linearly as a function of the arc length to remove the curvature discontinuity. In this study, a clothoid-based lane model is used to solve the curvature discontinuity problem. To derive a clothoid in Cartesian coordinates, the arc length s of the curve is defined as

$$\frac{1}{R} = \frac{d\theta}{ds} \propto s \quad (10)$$

$$\frac{d\theta}{ds} = \frac{s}{R_c s_o} = 2a^2 s \quad (11)$$

where R_c is the radius of the circular curve at the end of the spiral, s_o is the length of the spiral curve, a is the gain factor clothoid which is given by

$$a = \frac{1}{\sqrt{2R_c s_o}} \quad (12)$$

$$\theta = (as)^2 \quad (13)$$

Thus, we obtain

$$x = \int_0^L \cos[(as)^2] ds \quad (14)$$

$$y = \int_0^L \sin[(as)^2] ds \quad (15)$$

If $a = 1$, which is the case of a normalized clothoid, the coordinates are given by Fresnel integration as

$$C(L) = \int_0^L \cos s^2 ds \quad (16)$$

$$S(L) = \int_0^L \sin s^2 ds \quad (17)$$

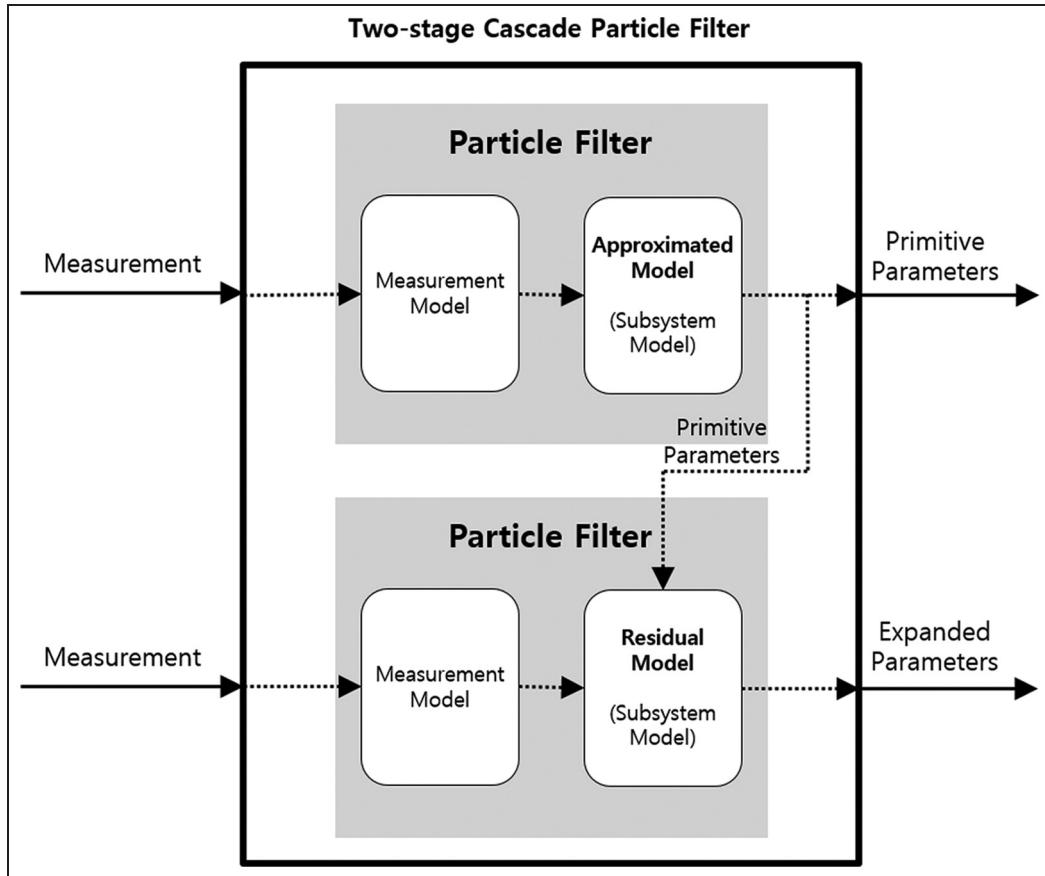


Figure 3. Two-stage CPF.

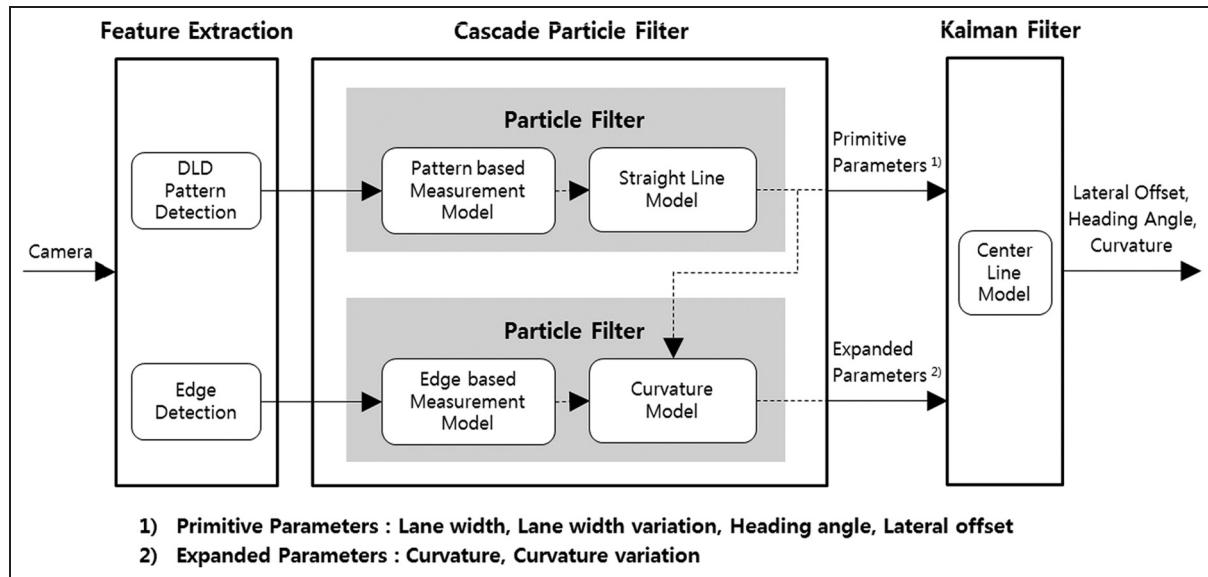


Figure 4. System architecture of lane detection and lane tracking.
 DLD: dark-light-dark.

Finally, the power series expansion of a cosine and a sine can be applied to equations (14) and (15) to decrease the computation time. The clothoid-lane model increases the smoothness of the transition.

However, the clothoid also increases the complexity of the lane model. In particular, if the clothoid model is applied to a particle filter, the main drawback of the complicated model is the large number of required

Table I. Comparison of the differences between the first-order lane model and the second-order lane model.

Lane model	Lateral offset (m)	Heading angle (deg)
First-order model	0.004 482	0.072 599
Second-order model	0.007 829	0.307 876

particles to span all possible states. To reduce the computational load, model decomposition can be applied to lane models. The complexity of a lane model can be reduced by dividing a lane model into submodels. In this study, a road model is decomposed into two submodels using physical model decomposition. The system states are categorized into two groups such as straight-line elements (the lateral offset and the heading angle) and curve elements (the curvature and the curvature derivative) for a curved-lane model. Although the physical model decomposition tends to be archived heuristically without a mathematical background, it can be directly related to systems architecture. Another possible solution is mathematical model decomposition. However, mathematical decomposition is difficult to apply to complex non-linear system models. Moreover, it is not easy to consider the system architecture and the physical meaning.

Lane particles and model decomposition

Because of the city traffic and the distance of the camera with respect to visibility, the complex lane model suffers from a loss of stability. Moreover, the computation time increases with increasing state variables as well. On the other hand, the decomposed state variables and the particle filters simplify problems such as the stability, the robustness and the computation time.

The clothoid-lane model requires six state variables: the heading angle ψ , the lateral offset y_o , the lane width w , the variation w' in the lane width, the transition point distance c and the curvature δ . However, a transition point and a curvature are not required for the straight-lane model. By separating the state variables, the CPF increases the traceability of the lane detection system and reduces the computation time which is its most significant advantage. Consequently, the particles of the first-stage particle filter are designed as

$$\mathbf{x}_t^k = [\psi \ y_o \ w \ w']^T \quad (18)$$

where \mathbf{x}_t^k is the k th particle at time t . Using these particles, the first-order lane model is described to evaluate the measurement. In the same way, the particles of the second stage are designed as

$$x_t^k = [c \ \delta]^T \quad (19)$$

In the above equation, the second stage does not include particles of the first stage because the estimated state variables are provided by the first stage, as shown in Figure 4.

Prediction model

State variables of the lane model are changed by depending on the road conditions and the vehicle motions. In particular, because the road geometry varies according to the road environment, the motion of the lane targets in local coordinates cannot be easily defined. Thus, in this study, a free-motion model is used to represent the lane model, as

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \quad (20)$$

where \mathbf{A} is the process model and \mathbf{w}_t is the Gaussian noise variable with the respective covariance matrix \mathbf{Q} . The process model \mathbf{A} and the Gaussian noise \mathbf{w}_t are

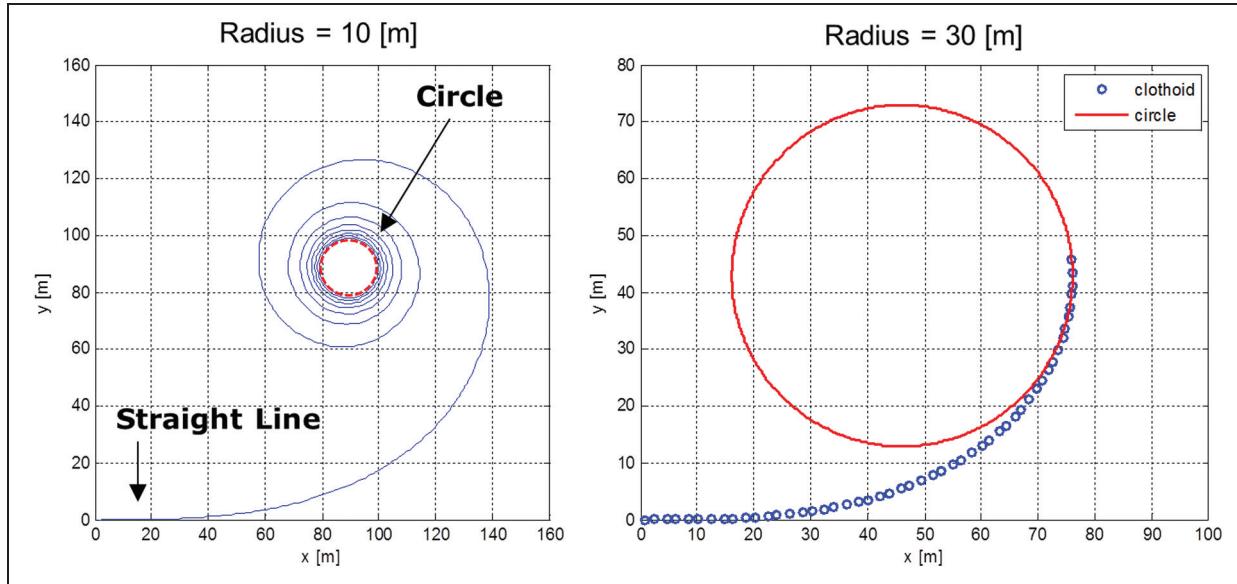


Figure 5. A clothoid which is an intermediate curve to connect a straight line and a circle.

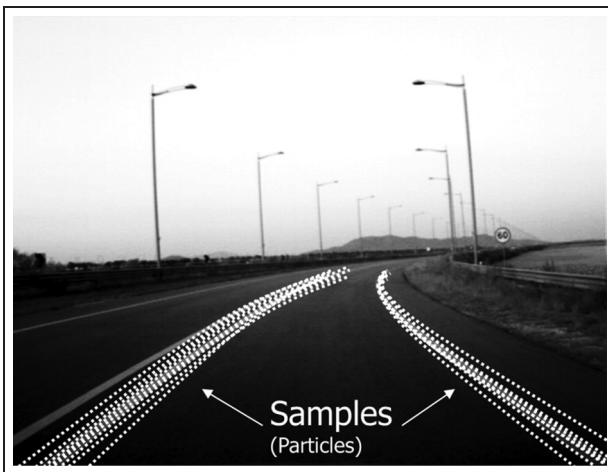


Figure 6. Representation of the probability density for lane state estimation using a set of samples.

used for the deterministic drift and the stochastic diffusion respectively. Because of the unpredictable characteristics of the lane, the process model of the lane parameters is designed as an identity matrix. The Gaussian noise w_t is calibrated through the lane parameter estimation using ground-truth lane-marking points. By using the probabilistic representation of the particle filtering, the lane states are represented by the set of samples, as shown in Figure 6.

Measurement model

In this study, two measurement models were designed for each stage, as shown in Figure 3: the approximated measurement model and the residual measurement model. Distinct types of feature extraction method were used at each stage of the particle filter: DLD pattern detection for the first stage, and Canny edge detection for the second stage. The DLD-pattern-matching method, which is a robust lane detection method, is adequate for near-range measurements. In particular, both the robustness and the real-time performance are satisfied by simplification of the detection algorithm. The performance was proved by TerraMax vision system which applied the DLD pattern detection method at the Urban Challenge 2007. On the other hand, the Canny edge detection method is suitable for the far-range measurements owing to the detailed and precise edge information.

Approximated measurement model (first stage). The approximated measurement model consists of DLD pattern detection and a distance transform. The edge information of the lane markings is generated by the DLD pattern detection method. Then, the edges are transformed to probabilistic representation by the distance transform method. The DLD pattern detection can be implemented by calculating the horizontal difference as

$$I_d(u, v) = m_r(u, v) - m_l(u, v) \quad (21)$$

where

$$m_l(u, v) = \frac{1}{N} \sum_{k=1}^N I(u-k, v) \quad (22)$$

$$m_r(u, v) = \frac{1}{N} \sum_{k=1}^N I(u+k, v) \quad (23)$$

In the above equations, u and v are the image coordinate axes and I is the two-dimensional image vector. The horizontal difference I_d is the difference between the left average m_l and the right average m_r . The left and right averages are calculated within an averaging window N . If I_d is larger than the threshold value γ , the horizontal coordinate u is considered as the rising edge e_r . Similarly, if I_d is smaller than the threshold value $-\gamma$, the horizontal coordinate u is considered as the falling edge e_f as in the equations

$$E(u, v) = \begin{cases} e_r & \text{if } I_d(u, v) > \gamma \\ e_f & \text{if } I_d(u, v) < -\gamma \\ \text{none} & \text{otherwise} \end{cases} \quad (24)$$

where $E(u, v)$ is the edge map. Then, IPM is performed with a homography matrix \mathbf{H} to consider the lane-marking width in the world coordinates as

$$\mathbf{H}\mathbf{u} = \mathbf{x} \quad (25)$$

where $\mathbf{u} = [u \ v]^T$ is the position vector in the image coordinates and $\mathbf{x} = [x \ y]^T$ is the position vector in the world coordinates. Accordingly, the world coordinates of u_r and u_f are calculated as $\mathbf{H}u_r = x_r$ and $\mathbf{H}u_f = x_f$ where $\mathbf{u}_r = [u_r \ v_r]^T$ and $\mathbf{u}_f = [u_f \ v_f]^T$ are the rising position and the falling position respectively in the image. Consequently, $x_r = [x_r \ y_r]^T$ and $x_f = [x_f \ y_f]^T$ are the world coordinates of the rising edge and the falling edge respectively which are converted through IPM. Finally, the pattern length w is calculated as

$$w = |x_f - x_r| \quad (26)$$

If the pattern length w is less than the predefined threshold value, the pattern is considered to be a candidate for lane markings. Then, the candidates are marked in a two-dimensional matrix. In this matrix, the marked positions are the exact positions of the lane markings. Therefore, the distance transform from the binary map to the probabilistic distribution is required to apply particle filtering.

Residual measurement model (second stage). The second measurement model is generated by using the Canny edge detector. The Canny edge detector is widely used for lane-marking detection systems. In particular, straight-line segments can be easily detected by using the Canny edge detector and the Hough transform.

However, in this proposed measurement model, the Canny edge detector is used only for the curved-lane model. As the near-range lane estimation has already been performed in the first-stage particle filter, a Hough transform is not required. Moreover, because the far-range lane estimation requires high-resolution measurements, detailed edge detection is needed to provide distinguishable likelihoods to the second particle filter. However, the distance transform is not used for Canny edge images in order to preserve accurate measurement for the long range.

The Canny edge detector is the most dependable edge detector, which uses difference-of-Gaussian and connectivity analysis. In terms of the far-range measurement, the advantages of the Canny edge detector is that detailed edge information can be acquired by configuring the double threshold values, which are the design parameters. In this study, using experimental data, these design parameters are carefully chosen to ensure the robustness of the far-range measurement.

Another candidate method for the far-range measurement is the gradient map. The gradient map contains the magnitudes and the directions by applying the Sobel operator. In particular, directional information can be used for eliminating inaccurate measurements. In spite of the potential ability of the gradient map, these advantages can be met only by advance information of the road curvature. In future work, precise measurement data can be obtained by applying the ADAS map with the GPS.

Measurement evaluation. The positions of lane markings and the inside of the lane markings are calculated to evaluate the particles as

$$y_L(x) = y_o + x \tan \psi + \frac{w_0 + w_1 x}{2} \quad (27)$$

$$y_R(x) = y_o + x \tan \psi - \frac{w_0 + w_1 x}{2} \quad (28)$$

$$y_{LN}(x) = y_L - \tau \quad (29)$$

$$y_{RN}(x) = y_R + \tau \quad (30)$$

where $y_L(x)$, $y_R(x)$, $y_{LN}(x)$ and $y_{RN}(x)$ are the lateral positions of the evaluation point when the longitudinal position x is given. In particular, the inside of the lane

is evaluated using $y_{LN}(x)$ and $y_{RN}(x)$ with τ , which is the distance from the lane markings. The likelihoods are calculated for each particle using the distance transform of the DLD pattern as

$$\begin{aligned} p_L(x) &= p(y_L(x)|z_t) \\ &= \mathbf{D}(x, y_L(x)) \end{aligned} \quad (31)$$

Similarly, $p_R(x)$, $p_{LN}(x)$ and $p_{RN}(x)$ can be calculated by using the distance transform. To summarize the evaluation, the likelihoods are weighted by α and β as

$$q = \sum_{x=1}^M \alpha[p_L(x) + p_R(x)] + \beta[p_{LN}(x) + p_{RN}(x)] \quad (32)$$

Finally, the likelihoods of the particles are computed using the Gaussian distribution to determine the probability of each particle.

$$p(\mathbf{x}_t^k | z_t) = \mathcal{N}(q; \mu, \sigma^2) \quad (33)$$

Experimental results

Experimental environment and conditions

The proposed lane detection algorithms were applied to the autonomous vehicle A1, which won the Autonomous Vehicle Competition organized by the Hyundai Motor Group in the Republic of Korea.^{2,24,25} A high-dynamic-range (HDR) complementary metal-oxide-semiconductor (CMOS) camera is installed behind the windshield. The HDR CMOS camera has a resolution of 768 by 576 pixels and an HDR of 120 dB. The proposed algorithm is implemented in MATLAB and C++ language and evaluated on an Intel Core 2 Duo central processing unit with 2.1 GHz.

In order to evaluate the robustness of the lane detection algorithm, various road and environment conditions were considered. Table 2 shows the test roads and the environmental conditions. During the highway driving, lane keeping, lane changing, lane splitting and lane merging were considered. Moreover, during the urban driving, various disturbance conditions were included such as crossroads, weak lane-marking conditions and

Table 2. Test roads and conditions.

Road (city to city)	Length (km)	Driving time (h:min)
<i>Highway</i>		
Seoul Ring Expressway (Songpa-Hanam)	10	17:28–17:31
Jungbu Expressway (Hanam-Hobeop)	41	17:31–18:05
Yeongdong Expressway (Hobeop-Yeoju)	15	18:05–18:15
Jungbu Naeryuk Expressway (Yeoju-Chungju)	42	18:34–19:02
<i>Urban</i>		
Seoul (Seongdong)	1	13:17–13:19
Gyeonggi Province (Guri)	1	17:45–17:47
Incheon (Yeongjong Island)	3	13:21–17:26

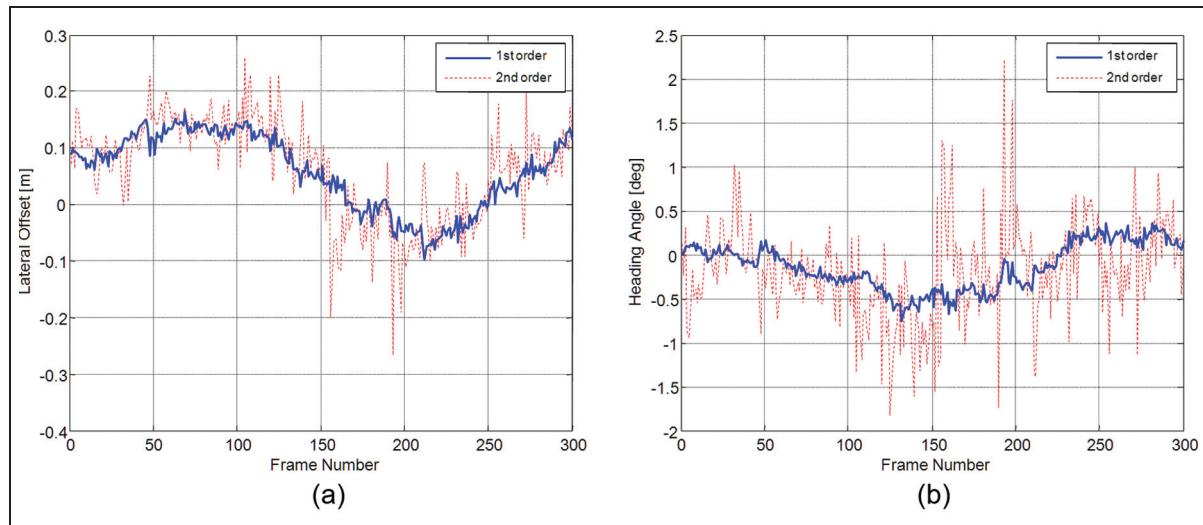


Figure 7. Variations in the states of the lane models.

limited visibility conditions due to a leading vehicle. The detection results have been given by Lee.²⁶

Variations in the estimation error in particle filtering by considering the model order

Lane models have been improved to describe lanes accurately. Simplified lane models, such as the polynomial lane model, are adequate for highway driving. Control systems of highway applications, such as a lane-keeping assistance system, require a limited number of lane states. Moreover, simplified lane models are more robust against noise and require a short computation time. For urban driving, however, a more complicated lane model is required to represent a complex road environment. In addition, geometric assumptions for the coordinate transformation can be violated by disturbances such as a non-uniform road surface and a road slope. For these reasons, numerous modelling parameters are included to describe lanes in various driving conditions. However, the robustness and the accuracy of state estimation are decreased in comparison with simplified lane models owing to the increased degrees of freedom. In particular, a lane detection system cannot directly measure the lateral offset and the heading angle of the vehicle at the centre of gravity, which is a non-viable area of the forward-looking camera. Accordingly, state estimation relies on extrapolation approaches. In the case of extrapolation, higher-order models show more unstable estimation results.

To analyse the correlation between the order of the lane model and the variations in the estimation results, a lane model analysis system was developed. The analysis system excludes the effect of lane-marking detection error by determining the positions of the lane markings manually. The analysis is performed for 300 frames of highway images. As shown in Figure 7, the mean values of the estimated lateral offset and the heading angle of

the lane are similar. However, the variance of the second-order lane model is twice to four times larger than that of the first-order lane model. The instability is caused by the higher degrees of freedom of the second-order lane model. Consequently, the order of the lane model should be carefully determined to ensure stable estimation results.

Performance evaluation

Figure 8 and Figure 9 show the experimental results for the highway and the urban areas respectively. The vehicle speed was 50–110 km/h during the experiment. The highways have a lane width of 3.7 m, with more than two lanes. The performance of the lane detection system is measured using the number of valid frame numbers during the lane keeping and the lane changing. Generally, because the ground truth of the lane-marking positions cannot be easily acquired using a vision system, qualitative evaluation must be performed. The number of valid frames was 43,460 of the total 45,804 frames during experiments for 25 min 27 s. Consequently, the proposed algorithm showed a detection rate of 94.88%.

The urban areas have a variable road width and kerb. The vehicle speed was 0–60 km/h during the tests. As shown in Figure 9, despite absent or worn lane markings in urban areas, the proposed lane detection algorithm was able to maintain tracking reliably. The number of valid frames was 4883 of the total 6831 frames during experiments for 3 min in Gyeonggi Province. Consequently, the proposed algorithm showed a detection rate of 71.48% in urban areas.

Figure 10 and Figure 11 show the state estimation results for the highway and for the urban areas respectively. In the highway scenario, five lane changes can be found using the discontinuity of the lateral offset. The estimation results show that the CPF keeps track of the

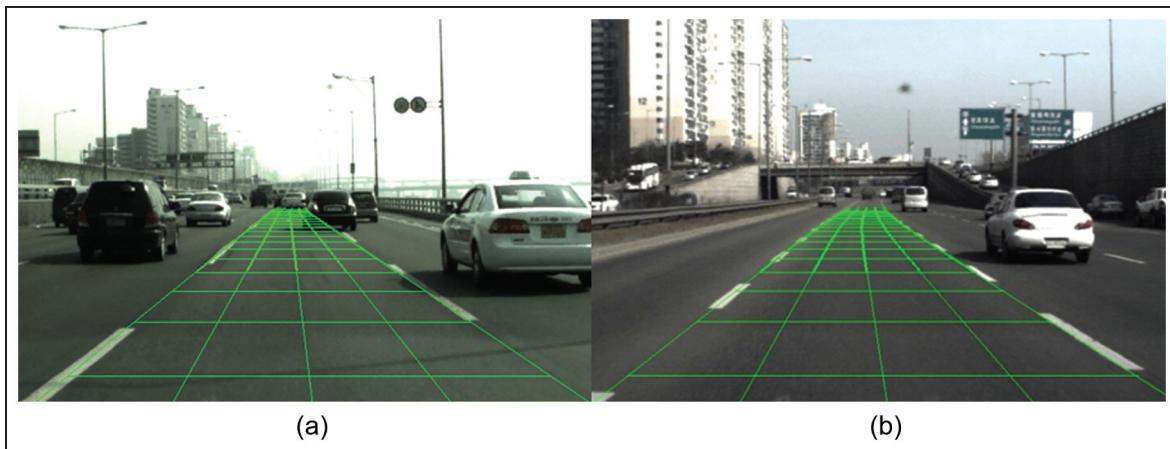


Figure 8. Experimental results for the highway.



Figure 9. Experimental results for the urban areas.

lane during high-speed driving. On the other hand, in the urban scenario, CPF loses track of the lane owing to the weak measurement conditions at frame number 1500 (a red box). In the fourth plot, the index is used as the unit of curvature owing to the implementation limitations of the clothoid. The range of the index values is from 0 to 400 and the corresponding radius of the curve is in the range from 24 m to infinity.

Figure 12 shows the experimental results of the local roads. In particular, the lane markings of the local roads are severely worn or covered with dirt. Figure 13 is the experimental results of night driving conditions. During the night driving conditions, illumination, such as the headlight of cars driving in the opposite direction, disturb the visibility of the road as recorded by the cameras. Thus, the tracking algorithm is important to reduce the influence of the changes in the illumination.

Figure 14 and Figure 15 show comparisons between the results for the CPF and the conventional particle filter. The results show the estimation results of the lateral offset and heading angle. Although the quantitative results significantly depend on the environmental conditions, the mean error for the CPF was 0.007 621, and the mean error of the third-order particle filter was

0.116 971. The variances for the CPF and the third-order particle filter were 0.010 036 and 0.011 961 respectively. As shown in Figure 14(a) and (b), the CPF shows better estimation results than the conventional particle filter does with a second-order lane model. The results in Figure 14(c) are similar to those in Figure 14(b) because of the higher order of the lane model.

Figure 15 shows the heading-angle estimation results for the particle filters. The heading angle estimation for the CPF is much more accurate than for the other conventional particle filters. Because the heading angle is sensitive to the order of the lane model, the effect of the approximated tracking of the CPF shows a significant improvement in the estimation stability.

Computational aspects of the cascade particle filter

Since particle filters are expensive in terms of the computational costs, we should optimize the number of particles. Generally, increasing the number of particles and states increases the computation time exponentially. In this paper, however, the complexity of the lane model decreased from the order $O(n^2)$ to the order

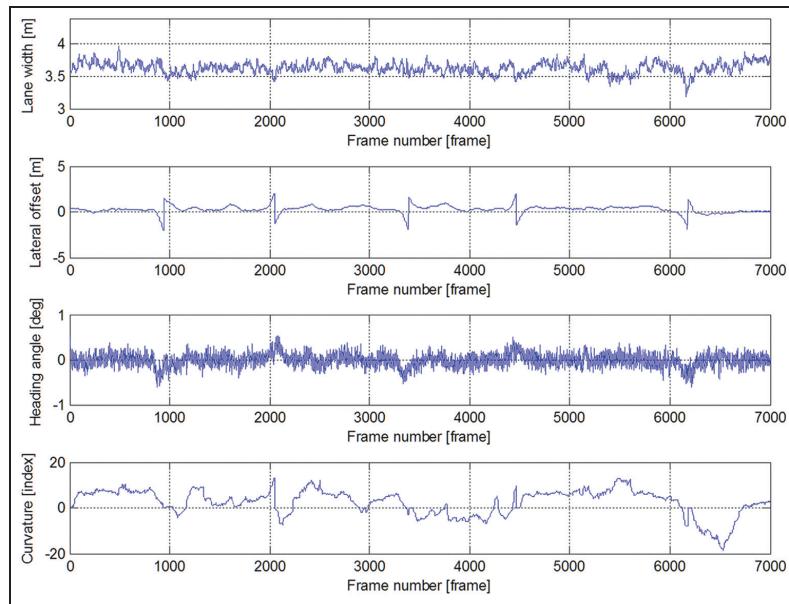


Figure 10. State estimation results for a highway.

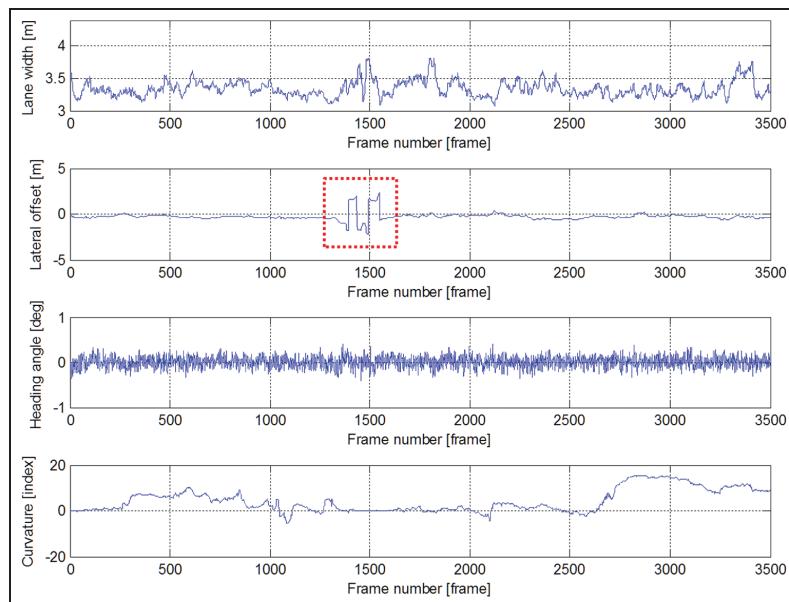


Figure 11. State estimation results for urban areas.

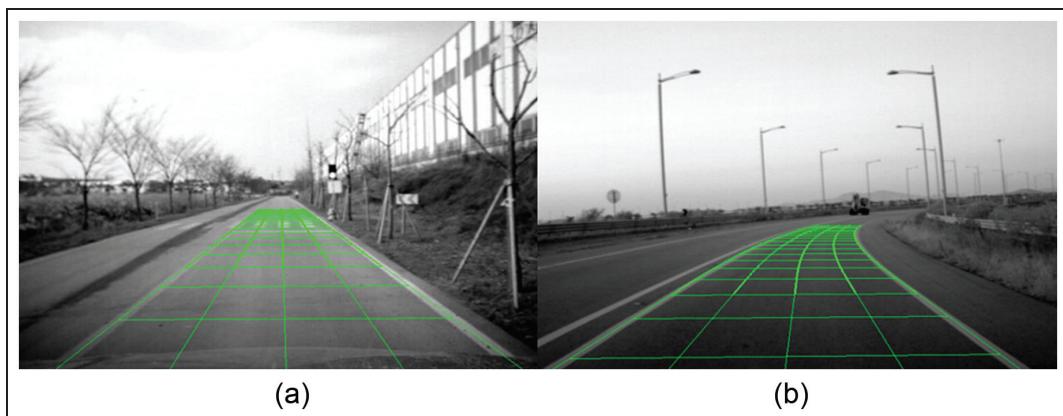


Figure 12. Experimental results for local roads.



Figure 13. Experimental results for night driving conditions.

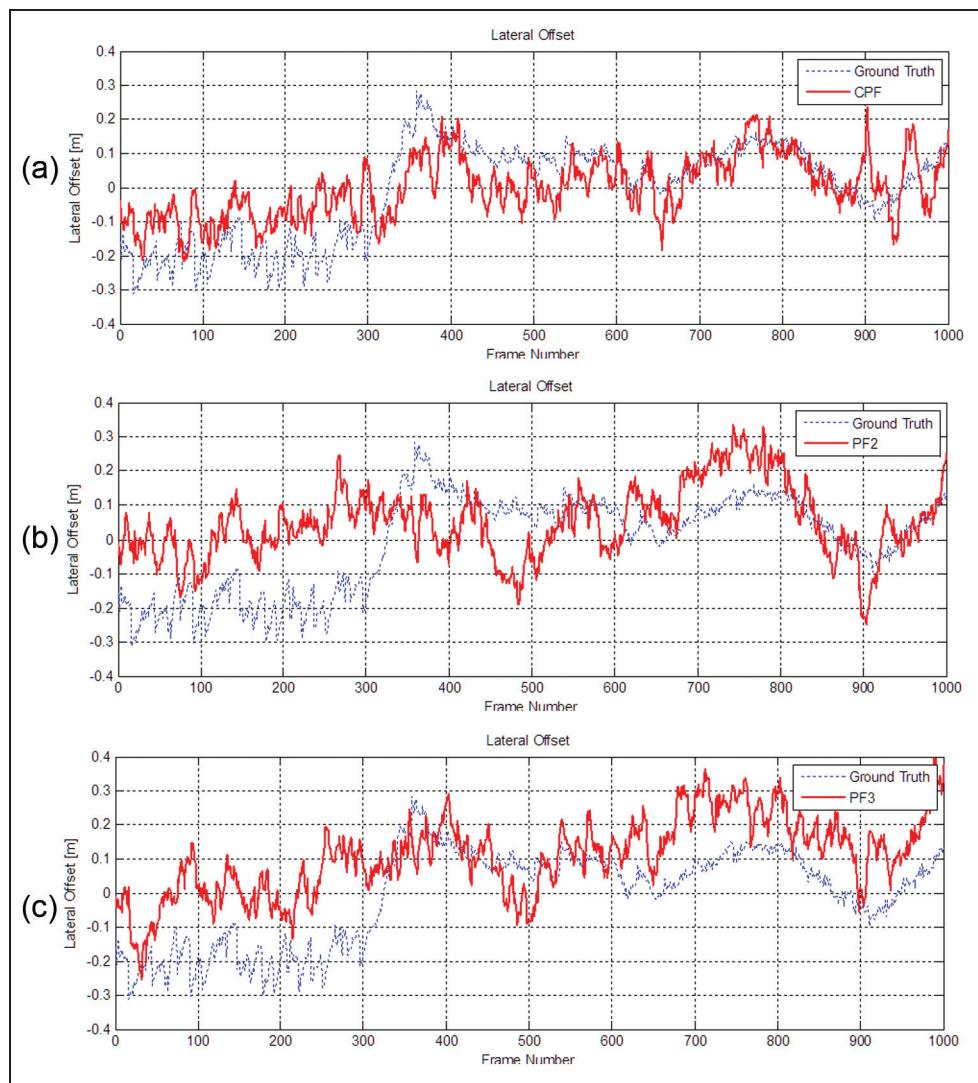


Figure 14. Comparisons between the lateral offsets for the CPF and the conventional particle filter.
CPF: cascade particle filter; PF2: second-order particle filter; PF3: third-order particle filter.

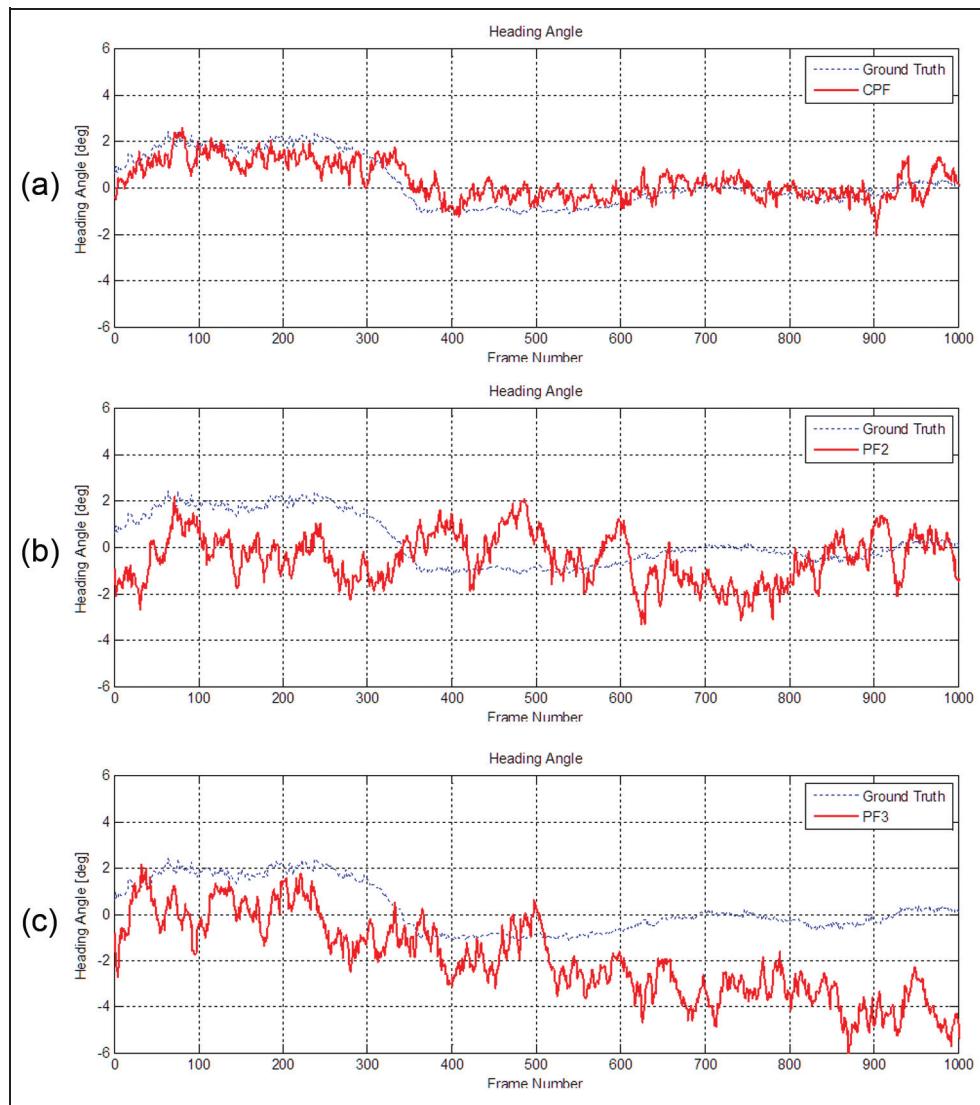


Figure 15. Comparisons between the heading angles for the CPF and the conventional particle filter.

CPF: cascade particle filter; PF2: second-order particle filter; PF3: third-order particle filter.

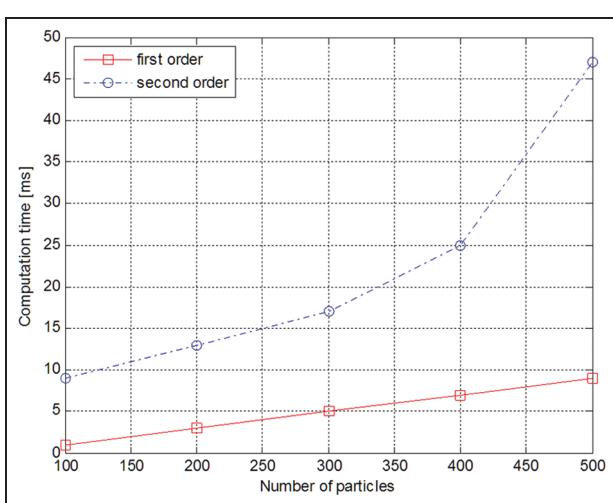


Figure 16. Comparison between the computation times for the first-order model and the second-order model.

$O(m^2) + O((n - m)^2)$ by decomposing the states into two-stage models. As a result, the computation time of the particle filter is reduced. Moreover, the reduced computation time can be used to increase the number of particles to enhance the robustness of the lane-tracking system. Figure 16 shows the comparison between the computation times for the first-order model and the second-order model. Because of the number of states of the second-order model, the computation time is increased. As the number of particles increases in the second-order model, the computation time increases exponentially. On the other hand, as shown in Figure 17, the computation time for the CPF shows a greater reduction than does the second-order model, because of the efficient structure of the CPF algorithm. In this study, the overall computation time of the lane detection system is less than 40 ms, and the deviation of the computation time is less than 5 ms.

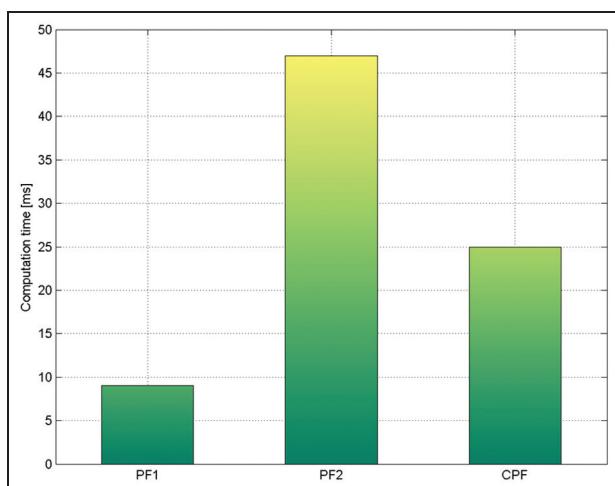


Figure 17. Comparison between the computation times of the particle filters.

PF1: first-order particle filter; PF2: second-order particle filter; CPF: cascade particle filter.

Conclusion and future work

In this study, a lane detection algorithm and a CPF are proposed to improve the robustness and the stability of a lane-tracking system. The proposed CPF improved the structure of the conventional particle filter by cascading multiple particle filters and decomposing system models. The model decomposition scheme reduces the complexity and the computation time of the particle filters. In particular, because of the limitation of the conventional particle filter, stable state estimation cannot be easily obtained while preserving a wide range of tracking coverage. On the other hand, the CPF and model decomposition can increase the stability and the coverage of lane tracking by separating the lane models by considering the significance of the state variables and the systems architecture.

In order to evaluate the performance of the proposed lane detection algorithm and the CPF, we conducted extensive experiments in a variety of environmental conditions. Even though the theoretical analysis of the cascade structure is difficult, experimental results proved that, in many cases, the decomposed models and the CPF show superior performances by tracking and estimating the lane state more accurately than the conventional particle filter does. Consequently, the experimental results show that the proposed lane detection system is suitable for autonomous vehicles as well as for intelligent vehicles to improve the road and vehicle safety.

In future work, sensor data fusion will be performed with various onboard automotive sensors and a priori knowledge such as wheel speed sensors, a steering-angle sensor, a GPS and a high-precision map. In order to integrating this lane detection system into a localization system, sensor data fusion is necessary to supplement the absence of lane markings. The proposed lane detection system is suitable for combining multiple sensor

data owing to the flexible structure and scalability of the CPF.

Declaration of conflict of interest

The authors declare that there is no conflict of interest.

Funding

This work was supported by the National Research Foundation of Korea grant funded by the Korea government (grant number 2011-0017495), the Industrial Strategy Technology Development Program (grant numbers 10039673 and 10042633), the Energy Resource R&D program (grant number 2006ETR11P091C) of the Ministry of Knowledge Economy and the BK21 plus program (grant number 22A20130000045) under the Ministry of Education, Republic of Korea.

References

1. NHTSA. National motor vehicle crash causation survey. Report DOT HS 811 059, US Department of Transportation, National Highway Traffic Safety Administration, Washington, DC, USA, 2008.
2. Jo K, Chu K and Sunwoo M. Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning. *IEEE Trans Intell Transpn Systems* 2012; 13: 329–343.
3. Wang R, Xu Y, Libin and Zhao Y. A vision-based road edge detection algorithm. In: *2002 IEEE intelligent vehicles symposium*, Versailles, France, 17–21 June 2002, Vol 1, pp. 141–147. New York: IEEE.
4. Lopez A, Serrat J, Canero C et al. Robust lane markings detection and road geometry computation. *Int J Automot Technol* 2010; 11: 395–407.
5. Cafiso S, Di Graziano A, Di Silvestro G et al. Development of comprehensive accident models for two-lane rural highways using exposure, geometry, consistency and context variables. *Accid Analysis Prevention* 2010; 42: 1072–1079.
6. Bertozzi M, Bombini L, Broggi A et al. VIAC: an out of ordinary experiment. In: *2011 IEEE intelligent vehicles symposium*, Baden-Baden, Germany, 5–9 June 2011, pp. 175–180. New York: IEEE.
7. Pohl J, Birk W and Westervall L. A driver-distraction-based lane-keeping assistance system. *Proc IMechE Part I: J Systems and Control Engineering* 2007; 221: 541–552.
8. Shin D-H and Joo B-Y. Design of a vision-based autonomous path-tracking control system and experimental validation. *Proc IMechE Part D: J Automobile Engineering* 2010; 224(7): 849–864.
9. Shim T, Adireddy G and Yuan H. Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control. *Proc IMechE Part D: J Automobile Engineering* 2012; 226(6): 767–778.
10. Bertozzi M and Broggi A. GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans Image Processing* 1998; 7: 62–81.
11. Bertozzi M, Broggi A and Fascioli A. VisLab and the evolution of vision-based UGVs. *Computer* 2006; 39: 31–38.

12. Broggi A, Cappalunga A, Caraffi C et al. TerraMax vision at the urban challenge 2007. *IEEE Trans Intell Transpn Systems* 2010; 11: 194–205.
13. Cerri P, Soprani G, Zani P et al. Computer vision at the Hyundai Autonomous Challenge. In: *2011 14th international IEEE conference on intelligent transportation systems*, Washington, DC, USA, 5–7 October 2011, pp. 777–783. New York: IEEE.
14. Sehestedt S, Kodagoda S, Alempijevic A and Dissnayake G. Efficient lane detection and tracking in urban environments. In: *3rd European conference on mobile robots*, Freiburg, Germany, 19–21 September 2007, pp. 126–131. Freiburg: University of Freiburg.
15. Jiang R, Reinhard K, Tobi V and Wang S. Lane detection and tracking using a new lane model and distance transform. *Mach Vision Applic* 2011; 22: 721–737.
16. Danescu R, Nedevschi S, Meinecke MM and To TB. Lane geometry estimation in urban environments using a stereovision system. In: *2007 IEEE Intelligent transportation systems conference*, Seattle, Washington, USA, 30 September–3 October 2007, pp. 271–276. New York: IEEE.
17. Tian M, Liu F and Hu Z. Single camera 3D lane detection and tracking based on EKF for urban intelligent vehicle. In: *2006 IEEE International conference on vehicular electronics and safety*, Beijing, People's Republic of China, 30 September–3 October 2006, pp. 413–418. New York: IEEE.
18. Borkar A, Hayes M and Smith MT. Robust lane detection and tracking with RANSAC and Kalman filter. In: *2009 16th IEEE international conference on image processing*, Cairo, Egypt, 7–10 November 2009, pp. 3261–3264. New York: IEEE.
19. Vacek S, Schimmel C and Dillmann R. Road-marking analysis for autonomous vehicle guidance. In: *3rd European conference on mobile robots*, Freiburg, Germany, 19–21 September 2007, pp. 1–6. Freiburg: University of Freiburg.
20. Danescu R and Nedevschi S. Probabilistic lane tracking in difficult road scenarios using stereovision. *IEEE Trans Intell Transpn Systems* 2009; 10: 272–282.
21. Dickmanns ED and Mysliwetz BD. Recursive 3-D road and relative ego-state recognition. *IEEE Trans Pattern Analysis Mach Intell* 1992; 14: 199–213.
22. Ristic B, Arulampalam S and Gordon N. *Beyond the Kalman filter – particle filters for tracking applications*. Norwood, Massachusetts: Artech House, 2004.
23. Thrun S, Burgard W and Fox D. *Probabilistic robotics*. Cambridge, Massachusetts: The MIT Press, 2005.
24. Han J, Kim D, Lee M and Sunwoo M. Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor. *IEEE Trans Veh Technol* 2012; 61: 971–985.
25. Park I and Sunwoo M. FlexRay network parameter optimization method for automotive applications. *IEEE Trans Ind Electron* 2011; 58: 1449–1459.
26. Lee M. Probabilistic lane detection and tracking for autonomous vehicles using a cascade particle filter, <https://www.youtube.com/watch?v=azkoJMGIZEI&feature=youtu.be> (2012).