Journal of Korea Multimedia Society Vol. 17, No. 6, June 2014(pp. 655-670)
http://dx.doi.org/10.9717/kmms.2014.17.6.655

Model-Based Robust Lane Detection for Driver Assistance

Tan-Hung Duong[†], Sun-Tae Chung[††], Seongwon Cho[†††]

ABSTRACT

In this paper, we propose an efficient and robust lane detection method for detecting immediate left and right lane boundaries of the lane in the roads. The proposed method are based on hyperbolic lane model and the reliable line segment clustering. The reliable line segment cluster is determined from the most probable cluster obtained from clustering line segments extracted by the efficient LSD algorithm. Experiments show that the proposed method works robustly against lanes with difficult environments such as ones with occlusions or with cast shadows in addition to ones with dashed lane marks, and that the proposed method performs better compared with other lane detection methods on an CMU/VASC lane dataset.

Key words: Lane detection, Line segment extraction, Line segment clustering, Lane model fitting

1. INTRODUCTION

Lane detection is a fundamental ingredient for Driver Assistance systems which provide aid or assistance to the driver while driving such as Lane Departure Warning, Driver-Attention Monitoring, Automated Vehicle-Control Systems, and etc. [1]. Lane detection systems are based on various sensors including camera sensors, internal vehicle-state sensors, line sensors, LASER radio detection and ranging (RADAR) sensors, global positioning system (GPS) sensors, and so on. Since the camera sensor-based one can operate under various environments whereas the other lane detection systems work well under some special environments, research interests about the lane detection have been concentrated more on vision(camera sensor)-based systems during the past two decades [1-19].

The task of vision-based lane detection is relatively easy when the texture of the road is uniform and when the lanes present very clear markings. However, since road lanes usually have environmental variations in addition to shape variations (straight or curved), the task becomes non trivial.

Lanes can be marked by segmented lines, circular reflectors, physical barriers, or even nothing at all as well as well-defined solid lines. The road surface can be comprised of light pavement, dark pavement, or even combinations of different pavements. The roads can be under various weather conditions (cloudy, foggy, rainy, snowy, etc.) and illumination conditions (sunny, cloudy, dark). In addition to these, neighborhood vehicles, trees or buildings can cast shadows over lanes or the other vehicles or obstacles can occlude the lanes and distract the tracking system. Most of vision-based lane detection works in the past two decades [1-19] have focused on dealing with such difficulties of the road environments.

The lane detection in this paper is defined as a

※ Corresponding Author : Sun-Tae Chung, Address: (156-743) School of Electronic Eng, Soongsil Univ., 369, Sangdo-Ro, Dongjak-Gu, Seoul, Korea, TEL : +82-2-820-0638, FAX : +82-2-821-7653, E-mail : cst@ssu.ac.kr

Receipt date : Feb. 4, 2014, Revision date : Mar. 10, 2014
Approval date : May. 9, 2014

[†] School of Electronic Engineering, Soongsil University (tanhung.duong@gmail.com)

[††] School of Electronic Engineering, Soongsil University

[†††] Dept. of Electrical Engineering, Hongik University (swcho@hongik.ac.kr)

※ This work was supported by the Soongsil University Research Fund and Basic Science Research Program through the National Research Foundation of Korea (2012-006883).

problem of estimating the left and right boundaries of a lane that are designated by the solid or the dashed markers, and finds only the immediate left and right lane boundaries of the lane, which may be only interests for the driver.

In this paper, we propose an efficient and robust lane detection method based on hyperbolic lane model and line segment clusters. Hyperbolic lane model is known to represent well a parametric model of the projective projection of the natural lane borders for most of lane applications [12]. The proposed method firstly localizes a horizon. The localized horizon is used in two ways; first for restricting ROI and second for hyperbolic lane model. One does not have to search line segments above the horizon since the line segment coming from the lanes cannot be above the horizon. Next, the proposed method applies LSD(Line Segment Detector) [20] for extracting line segments, which is more robust to noises and faster than the conventional line extraction technique of Canny edge detector [21] followed by Hough transform[22]. LSD also provides endpoints of each extracted line segment so that one can calculate the slope of the line segment. Since line segments coming from lane marks are only concerned, the proposed method eliminates the infeasible line segments which have length less than a minimum threshold or almost horizontal angle. Next, the proposed method clusters the remaining feasible line segments based on slope and distance, and then determine the most probable left and right lane cluster candidate by calculating the weights of line segments belonging to each cluster. Since line segments coming from lanes head toward vanishing points, the weight for each line segment is defined in relation with vanishing points. The vanishing point in this paper is only necessary for the calculation of weights of line segments so that the performance of the proposed lane detection method is less dependent on the estimation of the vanishing points than the other lane detection methods where more precise vanishing point estimation is desirable [13,14,16,18,19]. Finally, the proposed lane detection method fits the left and right hyperbolic lane model using the most probable left and right lane cluster, respectively. Since our proposed method fits the lane model by utilizing the most probable line segment cluster, it is more robust compared with other model-based methods based on control points (including vanishing points) [14,18,19] or on any pairs of edge points with a distance of the proper value [13], or based on lane candidate pixels[15]. The precise estimation of control points, lane candidate pixels, and edges are more erroneous than that of line segments under difficult environments such as cast shadows, occlusions, bad weathers, and so on.

Experimental results on a publically available CMU/VASC lane dataset[27] shows that the proposed lane detection method works robustly against some worse land environments such as occlusions and cast shadows and performs better compared with other methods tested on the same dataset. The proposed method shows a processing speed of average 146ms per frame on the CMU/VASC lane dataset. Since the experimental computing environments (CPU speed, DDR memory size, and etc.) used in the literature are not publicized or not the same as ours so that comparison of the processing speed of ours with those in the other works is hardly figured out. For example, [18] reports the processing speed of lane detection as below 4 sec per frame on the similar dataset under Pentium 3 system with 128 MB RAM.

The rest of the paper is organized as follows. Section 2 describes related works, and Section 3 explains the proposed lane detection method in detail. Experimental results are explained in Section 4, and finally conclusion is given in Section 5.

2. RELATED WORKS

The goal of vision-based lane detection is to locate the lane boundary from image sequences.

During the past two decades, various vision-based lane detection methods have been proposed in the literature. These methods can be broadly grouped in two approaches, namely the feature-based approach [3-7] and the model-based approach [8-19].

The Feature-based methods extract and analyze local lane features in order to separate the lane from the background pixel by pixel. One of the most popular lane features is the edges since the lane boundaries are normally assumed to be clearly marked with white or yellow paint in high contrast. However, lane edges may not be clearly detected due to weak or worn-out paints, or cast shadows by neighborhood trees, buildings and vehicles or occlusion by other vehicles or obstacles, or bad weather conditions. Also, if the edge threshold is either too low or too high, a large number of irrelevant features will be included or correct lane features can be missed. As a result, if the driving environments are noisy or varying, the edges can mislead the detection algorithms.

On the other hand, lane modeling can greatly increase lane detection performance since through model fitting, one can eliminate false positives via outlier removal or recover the lane parts of weak appearance (due to cast shadows, worn-out paints, or bad weathers) or of hidden appearance (due to occlusion). Moreover, the model-based approaches just use a few parameters to represent the lanes. Assuming the shapes of lane can be presented by either straight line [8,9] piecewise linear lines [10], parabolic curves [11], hyperbolic curves[12-16], clothoids, splines [17,19], and snakes [18]. The processing of lane detection is approached as the processing of fitting those model parameters. In this way, the model-based techniques are much more robust against noise and missing data, compared with the feature-based techniques. To estimate the parameters of lane model, the likelihood function, Hough transform, the chi-square fitting, and mean square error fitting have been applied.

The classical technology used a simple model - straight linear lane model [8,9] or extended piecewise linear model [10] by considering the lane marks are straight. This technique is simple but it works not well to locate the curved lanes. In an application such as lane-departure warning system, it is required to calculate the trajectory of the vehicle a few seconds ahead. At freeway speeds, this can require accurate road modeling for 30-40 m or more ahead of the vehicle to catch a TLC (Time to Line Crossing) of around 1s . Thus, such an application may require lane detection in the far-field. In this situation, simple lane models like piecewise linear model cannot represent the lane shapes accurately, especially in the far-field so that more accurate curvature lane models such as parabolic, hyperbolic or spline-based model would be better for vehicle trajectory forecasting.

With Inverse Perspective Mapping (IPM), perspective effect on the image is removed and road lanes are modeled as parallel lines with fixed width in [17]. IPM technique is based on taking a top view of the image to remove the perspective effect with the assumption that the lanes are parallel. The IPM image is filtered to detect the vertical straight parallel lines using some optimized filters for enhancing the vertical edges. [17] generates a top view of the road, filtering using selective oriented Gaussian filters, using RANSAC line fitting to give initial guesses to a new and fast RANSAC algorithm for fitting Bezier Splines, which is then followed by a post-processing step. The big disadvantage of this approach is that it requires the camera is initially calibrated-intrinsic (focal length and optical center) and extrinsic (pitch angle, yaw angle) information not to mention the high computation time. The calibration parameters, especially extrinsic parameters of the cameras mounted on the moving vehicles can easily be changing due to the vibration of the moving vehicles.

In order to support more flexibility in modeling the arbitrary shape of road, [18] and [19] introduced the B-snake and B-spline model, respectively. The B-snake and B-spline model needs a precise calculation of control points which requires precise estimation of vanishing points. To estimate vanishing points, [18] and [19] apply the CHEVP algorithm which utilizes Canny edge detector and Hough transform. The Canny edge detector is sensitive to noises and Hough transform may be affected by shadow or weather change.

For more reliable line segments, [9] proposed a clustering technique of line segments. The proposed method in this paper also adopts the similar line segment clustering technique, but does differently from [9] in that it determines the most probable left and right line segment clusters, and it fits the different lane model of hyperbola as opposed to the linear regression lane model of [9].

As adopted in the proposed method, many of the previous research works adopting hyperbolic lane model have been reported [13-15]. [13] first detects edges by applying canny edge detector, which is sensitive to noises so that it can detect false lane edges. How to classify the edge points as lane points or not and how to choose a pair of left lane points and right lane points are not clearly stated in [13] and The lane detection method in [13] may be erroneous due to wrong edges detected by Canny detector. Also, [13] uses linear least square error method, which is less robust and more slowly convergent compared with Levenberg-Marguardt algorithm adopted in this paper. In [14], the image is divided into horizontal strips. Then, [14] clusters the road boundary line segments into left and right groups using both geometrical and statistical reasoning. Pairs of road boundary line segments—one from each group—are then used to detect VP (Vanishing Point) for each image strip. And [14] fits hyperbolic lane model by solving a weighted linear matrix equation using multiple VPs obtained for each image strip. The Hyperbolic lane model fitting of [14] highly depends on precise estimation of vanishing points. In [15], each point on each extracted line under the horizon is compared to the intensity average of left and right neighbors within a window defined by maximum lane width. The pixel having intensity higher than left and right neighbor pixels with given threshold is selected to be a lane candidate pixel. Lane candidate pixels are used to fit hyperbolic lane model using RANSAC (RANdom Sample Consensus). [16] fits the hyperbolic model using the Sobel gradient map and the steerable filtered feature map based on the Maximum A Posterior (MAP) estimation, which is more complicate and more computationally time consuming than the case of using line segments based on least square error estimation used in the proposed method.

Most of line segment extraction in the literature utilizes Hough transformation after they detect edges by applying edge detectors such as Canny edge detector. Canny edge detector and Hough transform together needs some sizable computational time and Hough transformation has some drawbacks in detecting line segments which will be explained later. Instead we adopt LSD(Line Segment Detector) [20] to extract line segments from images, which is faster and more robust than Canny edge detector/Hough transform.

3. THE PROPOSED LANE DETECTION

The Model-based lane detection method proposed in this paper consists of 4 steps; i) horizon localization, ii) line Segment detection, iii) line segment clustering, and iv) lane model fitting.

In the below, each step is explained in detail.

3.1 Horizon Localization

Localization of the horizon line in the road scenes is important at least for two reasons in this paper. First, the road cannot exit above the horizon so that searching for lane feature can be confined below the horizon so that one can save the lane detection processing time. More importantly, position (y - coordinate) of horizon line (V_H in (*)) is necessary for hyperbolic lane model adopted in this paper.

[13] and [14] obtain the horizon line based on the intersection of the lane segments, whose performance depends on reliable estimation of the lane segments. Even though [14] obtains y-coordinate of the horizon more reliably by fitting a line to several vanishing points calculated by intersecting between a left line segment and a right line segment, it is noticed that localization of many vanishing points and fitting VPs to a line is computationally heavy. As opposed to [13] and [14], [15] includes the y - coordinate of horizon line as one of fitting parameters of the hyperbolic lane model, and [16] treats it as known since the y - coordinate of horizon line can be calculated from the fixed camera parameters based on the assumption of a flat road.

In this paper, we develop a simple but efficient horizon localization based on the notion of the horizon line proposed in [4]. The horizon line divides the scene into the sky region and the road region. The sky region pixels usually show higher intensity than road pixels, and it might have a big difference of intensity as one approaches from sky to the ground. Nevertheless, the horizon line does not often happen at the global minimum or the first local minimum from the above in the curve. Hence, a regional minimum search had better be utilized to ensure the correct localization for the horizon line dividing the sky and road region accurately.

As suggested as in [4], to enhance the effect of lower intensity around horizon line in the scene image, a minimum value filter with a 3×3 mask is first applied to the scene image so that the value of a pixel in the image becomes the value of the darkest pixel in the neighbor of 3×3 mask. In the example of Fig. 1 below, the center value 66 in the window is changed into 0, which is the darkest in the window, if one applies the minimum filter of 3×3 mask around the center pixel in the window in Fig. 1.

18	25	48
44	66	66
15	0	210

Fig. 1. An image of 3×3 window.

Next, a vertical mean distribution is obtained by computing the average of gray values of each row in the filtered image, as shown in Fig. 2(b). Instead of searching the first minimum value along the vertical mean distribution curve, the curve is divided into n segments to obtain regional minimum (in this paper, n is chosen to be 10). A regional minimum S is represented as (m, p), where m is the average of gray values of the row where the regional minimum occurs, and p is the number of the row.

$$S = \{(m_i, p_i) \mid m_i \in [0{,}255], p_i \in [1, h]\};\ 1 \le i \le n. \quad (1)$$

where h is the image height, and i increases from the top of the image. Naturally, the sky region (the region above the horizon) is always located on the top of the road image. Therefore, the first region's regional minimum m_1 is taken as reference minimum and the global mean value m_g of the entire image is calculated to determine the overall change in intensity. Now, the y-coordinate of the horizon line, V_H is determined as p_i where i is the first integer number such that m_i satisfying the following relationship;.

$$[(m_i \le m_{i-1}) \cap (m_i \le m_{i+1})] \cap (m_i \le m_1) \cap (m_i \le m_g) \quad (2)$$

If there is no such V_H satisfying the relation (2), then we take p_1 as the y-coordinate of the horizon.

3.2 Line Segment Detection

In this paper, we extract line segments as lane features, which are later utilized to fit the hyperbolic lane model.

Line segment detection is an old and recurrent problem in computer vision. Standard methods first apply Canny edge detector [21] followed by a Hough transform [22] extracting all lines that con-

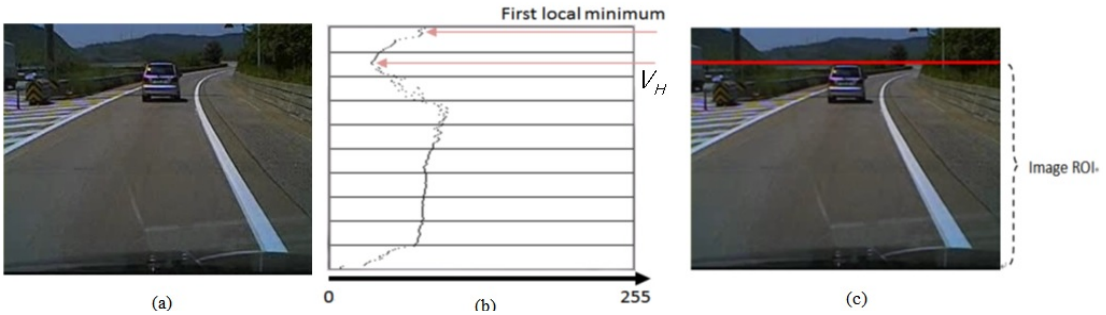


Fig. 2. Horizon Localization developed in this paper.

tain a number of edge points exceeding a threshold. These lines are thereafter cut into line segments by using gap and length thresholds. The Canny edge detector is sensitive to the low and high hysteresis thresholds so that inappropriate choice of fixed thresholds can lead to a significant number of false positives or false negatives. Moreover the Canny edge detector is well know to be computationally heavy [23], and the Hough transform method is also well known to have serious drawbacks. Textured regions that have a high edge density can cause many false detections [19]. There are many variants of the Hough transform, each trying to solve different shortcomings of the standard Hough transform. However, all Hough transform based techniques require binary edge map as input, and they usually generate infinitely long lines - rather than line segments - in (angle, radius) representation, which must then be broken down to line segments. A clean input edge map is very critical for these techniques, but the parameters to be used for edge map generation (for example, Canny edge detector's thresholds) are not automatic and have to be determined by the user. This approach is not suitable for such real time systems. An alternative approach is to use LSD [20] which shows a clear improvement of HT method in term of computation cost and false detection control. LSD does not only run in high speed, but also is capable of detecting the lines in segment form without the need for any further processing. Fig. 3 shows the advantage of using LSD for line segment detection.

3.3 Line Segment Clustering

3.3.1 Line segment elimination

The line segments extracted from LSD may contain the infeasible line segments which do not come from lane markings as shown in Fig. 3(b). For efficient clustering, the proposed method eliminates some of infeasible line segments based the following criteria:

Criteria for selection of line segments which can be candidates for lane marks

- The line segments must have a minimal length at least (the minimal length is 5 pixels in our implementation).
- The orientation of the line segments must not be horizontal or approximately horizontal (the slope of the line segment is not less than 5 degrees in our implementation).
- The end points of the line segments must lie under the horizontal line.

Fig. 4 shows a line segment map obtained after the above criteria is applied to Fig. 3(b).

3.3.2 Vanishing point estimation:

In the previous process, the line segments that have a very small probability of belonging to lane marks are eliminated. By checking the slope angle and length value of the extracted line segments, many infeasible line segments may be eliminated further, but this filtering process is not enough

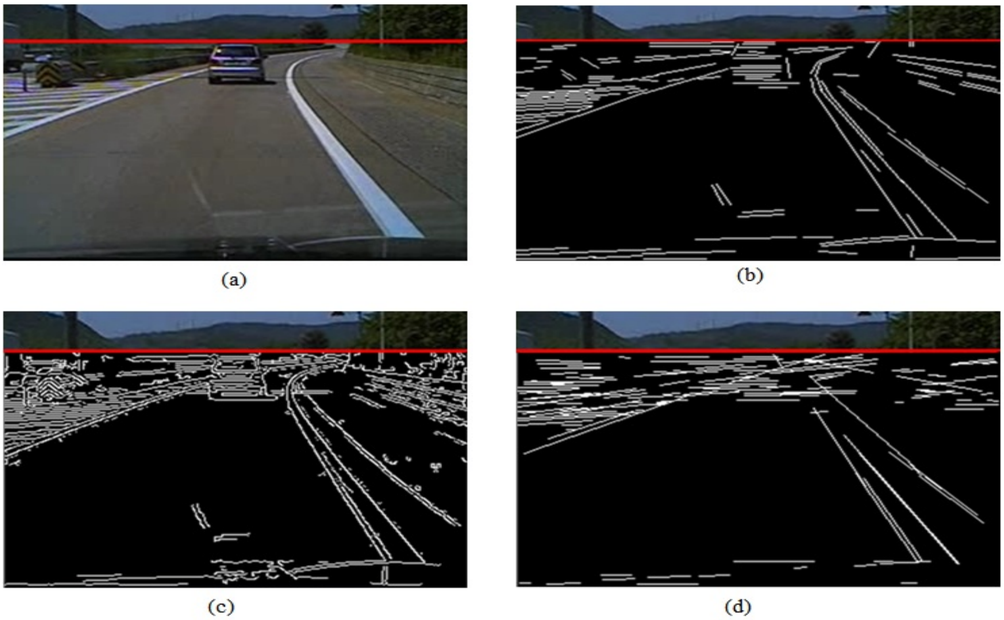


Fig. 3. (a) original image, (b) line segment image by LSD, (c) Canny edge map (low threshold = 50, high threshold = 100, window size = 3), (d) line segment image obtained by applying the probabilistic HT with threshold = 50, min line length = 5, max line gap = 5 on Canny edge map (c).

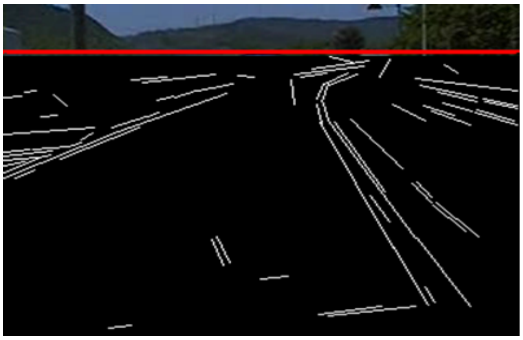


Fig. 4. A line segment map after the horizon-like line segements are removed.

since there can still be many line segments far from parts of lane marks since any fixed threshold for slope and length of line segments cannot be a panacea working for variant road scenes. In this paper, we develop a secure criterion on the selection of the most probable line segment cluster which can be appropriately used for the lane model fitting process later. The most probable line segment cluster is selected as the one which has the maximum weight. The weight of a line segment cluster is defined as the sum of the weight of each line segment in the cluster and the weight of a line segment is defined to be based on the vanishing point.

In general, the vanishing point is defined as the intersection point of the perspective projection of the set of lines that are parallel in the world, but not parallel in the image plane. The vanishing point in the image plane gives important information about the vehicle's orientation to the ground (pitch angle) and to the lane borders (yaw angle), the distance of the objects in the scene, and so on[23]. Therefore, vanishing point estimation plays an important role in driver assistance application like lane detection [13,14,16,18,19,24].

Although there can be several vanishing points determined by different sets of parallels in the scene (the vertical and horizontal directions of the panels, buildings, trees, and etc.), in this paper we focus on estimation of a dominant vanishing point since we want to utilize vanishing point information for a guideline for selecting the most probable line segment cluster and thus a less accurate estimation of vanishing point in this paper does not deteriorate the system performance as much as those in [13,14,16,18,19] where more precise vanishing points are better for more correct lane model fitting.

Estimation of VPs generally utilizes the intersection of a pair of line segments, but this approach is sensitive to outliers (wrong line segments) so that some method ([24]) assigns a weighting to each line segment to alleviate the sensitivity. In order to strengthen the reliability of estimating the dominant vanishing point, we also assign a weighting for each intersection and choose the intersection point of the maximum weighting as the dominant vanishing point.

Estimation of the dominant vanishing point

```
for i-th pair of line segments ( i =0, ···, n ) //n= the number of a pair of line segments,
{ //beginning of "for"
  compute the intersection point $p_i = (x_i, y_i)$ and the intersection angle $\theta_i$ of the pair of line segments ;
  if the intersection angle $\theta_i$ >= threshold
  { //beginning of "if"
    compute the weight $\omega(x_i, y_i)$ of this intersection point $p_i = (x_i, y_i)$;
    if there exists a registered intersection point group represented by $(x, y)$ where $p_i$ is located within the neighbor of 5 pixel Euclidean distance around $(x, y)$,
      { join the intersection point $(x_i, y_i)$ into the intersection point group $(x, y)$;
        update the weight of the intersection point group (x, y) such that $\omega(x, y) = \omega(x_i , y_i) + \omega(x, y)$ ; }
    else
      { register a new intersection point group represented by $(x_i, y_i)$;
        join the intersection point $(x_i, y_i)$ into the intersection point group $(x_i, y_i)$;
        assign the weight of the new intersection point group $(x_i, y_i)$ as $\omega(x_i, y_i)$; }
  } //end of "if"
} // end of "for"
Find the intersection point group $(\acute{x}, \acute{y})$ which has the maximum weight.
Finally, determine the dominant vanishing point as the average position of all intersection points belonging to the intersection point group $(\acute{x}, \acute{y})$.
```

In the above estimation algorithm, the weight $\omega(x, y)$ is defined as $\omega(x, y) \equiv \theta \frac{l_1}{l_2}$ where θ is the intersection angle in radians between two line segments , l_1 and l_2 are the length of two line segments with condition $l_1 < l_2$. If the slopes of two lines are m_1, m_2 then the intersection angle θ is calculated from the following formula:

$$\tan(\theta) = \left| \frac{m_1 - m_2}{1 + m_1 m_2} \right|.$$

Two straight lines with slopes $m_1 \neq m_2$ are perpendicular to each other if and only if $m_1 \cdot m_2 = -1$. Otherwise, their angle of intersection always satisfied the following inequalities:

$$0 < \theta < \frac{\pi}{2}.$$

Fig. 5 shows an example of estimating the dominant vanishing point for Fig. 4.

Fig. 5 shows also the estimated dominant vanishing point does not match the localized horizon

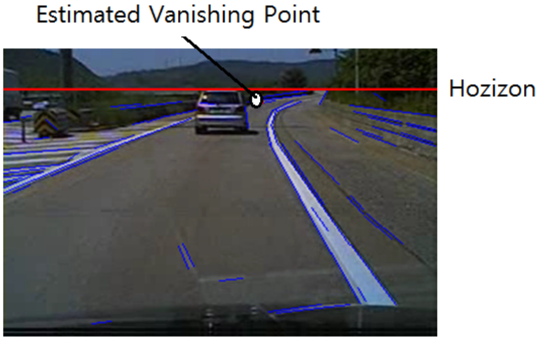


Fig. 5. Dominant Vanishing point estimated from Fig. 4.

and is not precise but does not usually cause any big problem for the proposed lane detection method since the estimated vanishing point is utilized for calculation of the weight of line segments. Fig. 8 shows an example of correct lane detection for Fig. 4 irrespective of the not precise estimation of vanishing point for Fig. 5.

3.3.3 Weighting of a Line Segment

We need to assign a weight for a line segment for choosing the most probable line segment cluster in the later process. The weighting factors for line segments are decided from the following observations:

- The longer a line segment, the higher is the probability that this line segment corresponds to a lane mark. .
- The greater the distance between the dominant vanishing point and the more distant endpoint of a line segment l_i, the higher is the probability that this line segment corresponds to a lane mark.

Based on this observation, we define the weight of a line segment l_i as follows:

$$w_i \equiv \frac{\mathrm{In}_i}{\sum \mathrm{In}_i} \cdot \frac{d_i}{\sum d_i} \tag{3}$$

where In_i is the length of line segment l_i, d_i is the distance of the line segment l_i to the dominant vanishing point, and Σ (summation) is calculated for all line segments.

3.3.4 Clustering of Line Segments

In this paper, we cluster the line segments by their slope and relative locations in two consecutive processes:

1) Clustering with respect to the slopes of line segments

2) Clustering again with respect to the relative location between line segments

The process of clustering starts with many clusters where each line segment represents one cluster, and with different levels of clustering they get merged according to their closeness with other clusters. The clustering with respect to the slope helps to find the set of line segments with similar slope angles and the clustering with respect to the relative location merges the closely placed clusters. The closeness (similarity) measure between any two clusters plays important role. A brief description about the clustering algorithm utilized in this paper in the similar way as in [8] is given as below:

Clustering of line segments

```
Input: a set of line segments L = {$l_1$, $l_2$, $l_3$, ..., $l_n$} , T-Threshold to stop merging of
       sub-clusters
Output: clusters with maximum inter class distance greater than or equal to T
1. Start with n disjoint clusters {$C_1$,$C_2$, $C_3$, ..., $C_n$}
   , each one containing one line segment, that is.
   $C_i$ ={$l_i$ } for each i=1, ⋯,n.
2. Compute the distance between each pair of clusters $C_i$ and $C_j$ ; d($C_i$, $C_j$)
3. Find the most close pair of clusters, $C_i$ and $C_j$ and merge into one cluster if their distance
   is less than T
4. Decrease the number of cluster by 1
5. Repeat step 2, 3 and 4 until either there is no two clusters with the inner class distance less
   than T or reached to single cluster.
```

The distance d(C_i, C_j) between two clusters C_i and C_j in the above clustering algorithm is defined to be the average distance from all the member of one cluster to all member of other cluster as given below:

$$d(C_i, C_j) \equiv mean_{(C_i x C_j)} d(l_m, l_n) \text{ where } l_m \in C_i, l_n \in C_j \tag{4}$$

And, for clustering with respect slopes, $d(l_m, l_n)$ is defined to be the absolute difference between the slopes of two segments, l_m and l_n. For clustering with respect to the relative location, d(l_m, l_n) is defined to be the minimum one among Euclidean distances between one of two endpoints of one line segment l_m and that of the other line segment l_n. Different threshold (T) values (for slope and the relative location) are used to stop the process of merging clusters.

Fig. 6(a) shows the outcome of clustering on Figure3 with respect to slope and Fig. 6(b) shows the outcome of further clustering on Fig. 6(a) with respect to the relative location where resulting clusters are painted in different colors.

3.4 Lane Model Fitting

3.4.1 Determination of the left and right lane candidate cluster

After clustering the line segments according to the slope and the relative location, only a few line segment clusters remain. Thus, the final task is to determine the most probable line segment cluster candidate respectively for each left and right lane. To achieve this task, firstly, we divide the set of clusters into two groups of left line segment clusters and right line segment clusters based on the slope of the line segment with the lowest location in each cluster. Due to the perspective effect, left lane marks appear to oriented toward left and right lane marks appear oriented toward right in the image planes. Next, we determine the cluster with the highest weight from the left cluster group as the most probable left cluster. Also, we determine the most probable right cluster in the same way. In this paper, the weight of the cluster is defined as the total weight of all line segments belonging to the cluster and the weight of the line segment is defined as (3).

Fig. 7 shows the resulting final most probable

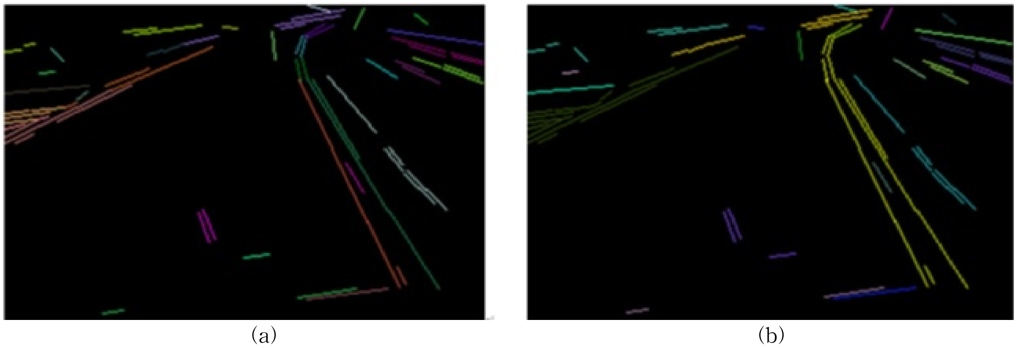


Fig. 6. (a) clustering on Fig. 3. with respect to slopes, (b) further clustering on Fig. 6(a) with respect to the relative location.

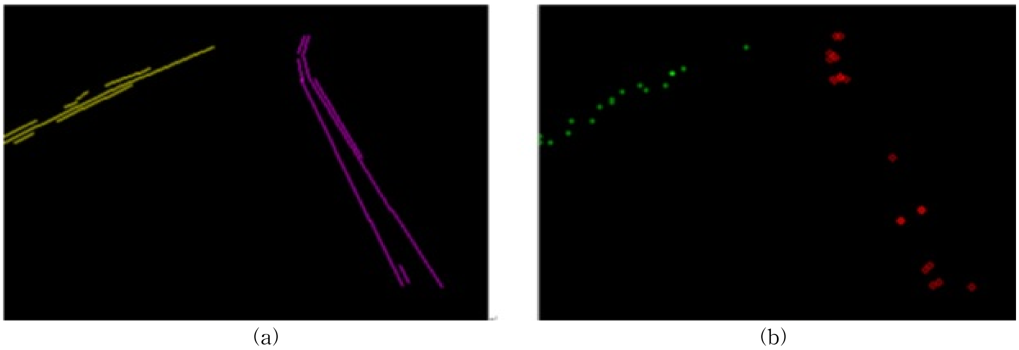


Fig. 7. (a) the most probable left and right line segment cluster from Fig. 6(b) the end points of the line segments of (a).

left and right cluster.

3.4.2 Hyperbolic lane model

The hyperbolic lane model adopted in this paper is expressed by the following formula:

$$u = \frac{k}{v - v_H} + b(v - v_H) + u_H \qquad (5)$$

where (u, v) is the lane boundary point in the image plane, v_H is the y-coordinate of the horizon; k,b and u_H are the parameters of the curve, which can be calculated from the shape of the lane on the ground. Geometrically, (u_H, v_H) is a vanishing point, b is a slope of the asymptote of the hyperbolic curve, and k controls the curvature of the curve.

The two lines formed by lane markings will have the same parameters of k and u_H if the left and right lane is parallel in the real world. Assume that the two lane marking curves can be represented by

$$\begin{aligned} u_l &= \frac{k}{v_l - v_H} + b_l(v_l - v_H) + u_H \\ u_r &= \frac{k}{v_r - v_H} + b_r(v_r - v_H) + u_H \end{aligned} \qquad (6)$$

where (u_l, v_l) is a (u, v) coordinate of the left lane and (u_r, v_r) is a (u, v) coordinate of the right lane. The hyperbolic lane model (5) is characterized by only three parameters (k,b,u_H). The parameter v_H is determined by the y-coordinate of the horizon localized in Section 3.1. To determine these parameters, we use a least squares curve fitting method, fitting the proposed model to the images acquired by the camera. This procedure is applied independently for each lane boundary, and is described in next section.

3.4.3 Estimating the hyperbolic lane model parameters

We estimate the hyperbolic lane model parameters $\beta = (k,b,u_H)$ by formulating the estimation problem as a nonlinear least square curve fitting problem as follows.

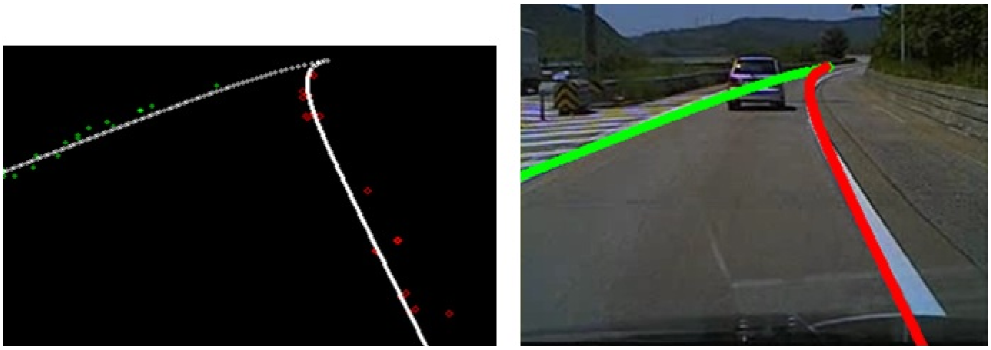
Hyperbolic lane model fitting problem

Given a set of m empirical datum pairs of independent and dependent variables, (u_i, v_i), optimize the parameters β of the model curve $f(v_i, \beta) \equiv \frac{k}{v_i - v_H} + b(v_i - v_H) + u_H$ so that the sum of the squares of the deviations $S(\beta) \equiv \sum_{i=1}^{m}[u_i - f(v_i, \beta)]^2$ become minimal.

In our case, m in the above is the total number of the end points of the line segments in the most probable line segment cluster.

Levenberg - Marquardt algorithm is a well-known robust method to solve the nonlinear least square error optimization problem [25]. In this paper, we fit the hyperbolic lane model of (6) by using the ' lmfit - a C library for Levenberg-Marquardt least-squares minimization library' [26].

Fig. 8 shows an example of hyperbolic lane



(a) (b)

Fig. 8. (a) the estimated hyperbolic lane model using LM algorithm, (b) the final result.

model fitting by LM algorithm. In Fig. 8(b), left lane is painted in green color and right lane is pained in red color both with some thickness.

4. EXPERIMENTAL ANALYSIS

4.1 Experimental Environments

For experiments in this paper, we evaluate the CMU/VASC lane dataset in [27] in order to compare the proposed method with other methods in the literature, which has been tested in [5], [6] and [19] on the same dataset. [18] tested their algorithm on many datasets including some of CMU/VASC datasets. [3,4,8-17] tested their algorithm on their own home-made datasets, and did not publish the test results clearly. The CMU/VASC dataset consists of 160 images of a sunny 1-lane road, containing curved lane images and cast shadowed lane images, and all images have the resolution of 256×240. But since this dataset does not contain lane images under variant road environments such as foggy lanes, we test our proposed method for some lane images collected from Internet. The testing PC in the our experiments has Intel® Core (TM) 2 Duo E6750 CPU running at 2.66GHz with 2 GB DDR2-800 RAM.

Any works of [1-19] do not provide the criterion to decide whether the lane is detected or not. In this paper, we determine that the lane is detected when the detected lane turns out to follow the real lane reasonably well by human eyes.

4.2 Experimental Results

Table 1 of the comparison between the experimental results on lane detection rate of previous lane detection methods [5,6,19] and that of our proposed method shows that our proposed method performs better than those methods of [5,6,19] with respect to lane detection rate.

From experiments on CMD/VASC [27], it turns out that the proposed method can process about 7 frames/sec in average (146ms per frame in average). A few of works in the literature report processing speed of their methods. Furthermore, the experimental computing environments (CPU speed, DDR memory size, and etc.) used in the experiments are not the same as ours so that we cannot directly compare the processing speed of ours with those of the other works. For example, [18] reports the processing speed of lane detection as below 4 sec per frame on the similar dataset under Pentium 3 system with 128 MB RAM.

Table 1. Comparisons of Lane Detection rate

Methods	Detection rate	The number of test images
[5]	94.26%	157
[6]	95.54%	157
[19]	82%	567(*)
Proposed method	96.18%	157

(*) [19] tested more lane images from CMU/VASC[28] in addition to [27], but it does not clearly state which lane images are used for its experiments from CMU/VASC[28] .

Fig. 9 and Fig. 10 show some of success cases and some of failure cases by our proposed method on CMU/VASC lane image dataset under variant road environments, respectively.

Our proposed method fails when lane curves are steep but lane marks of left or right lane are not clear as in Figure10 (a) and (b) or when both lane marks are unclear as in Figure10 (c) (see the area of ellipse). In the case of steep lane curves, the lane marks from the same lane are differently clustered as shown in rectangles of Figure10 (d) and (e) . Also lane marks in the other lane are not clearly detected as shown in the area of ellipse in Fig. 10(d). In Fig. 11 below , left and right lane marks are clearly detected so that fitting the hyperbolic left lane model helps correct fitting of the hyperbolic right lane model even though the lane marks from the same lane are differently clustered. In case of the unclear lane marks of both left and right lane, the unclear lane marks are not extracted as line segments as shown in the areas of ellipses

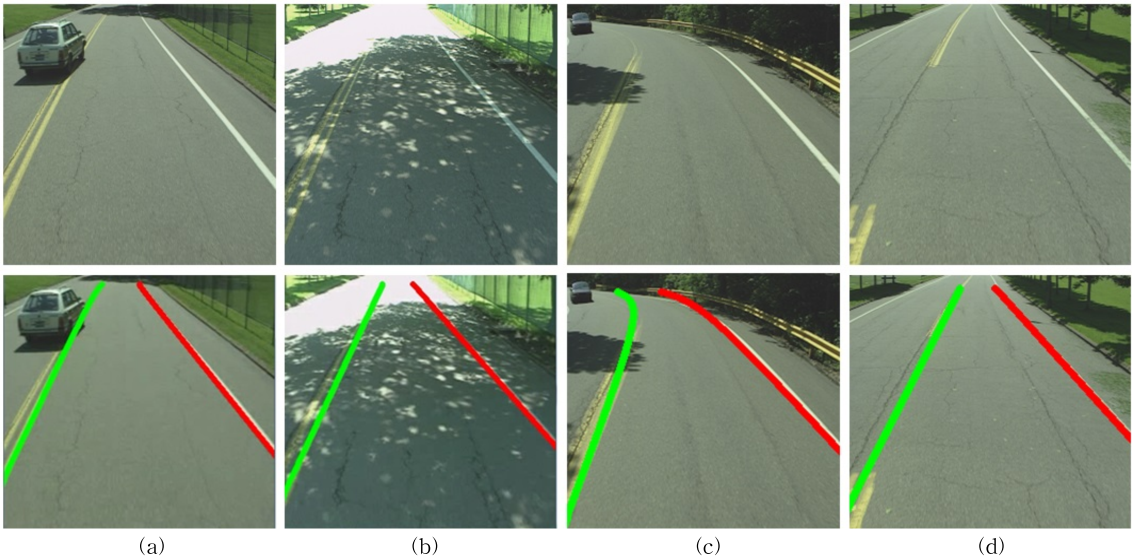


Fig. 9. Some of success cases of the proposed method on CMU/VASC dataset under various road environments. (a) a lane by other vehicle, (b) a lane with cast shadow, (c) a curved lane, (d) a lane with dashed lane marks.

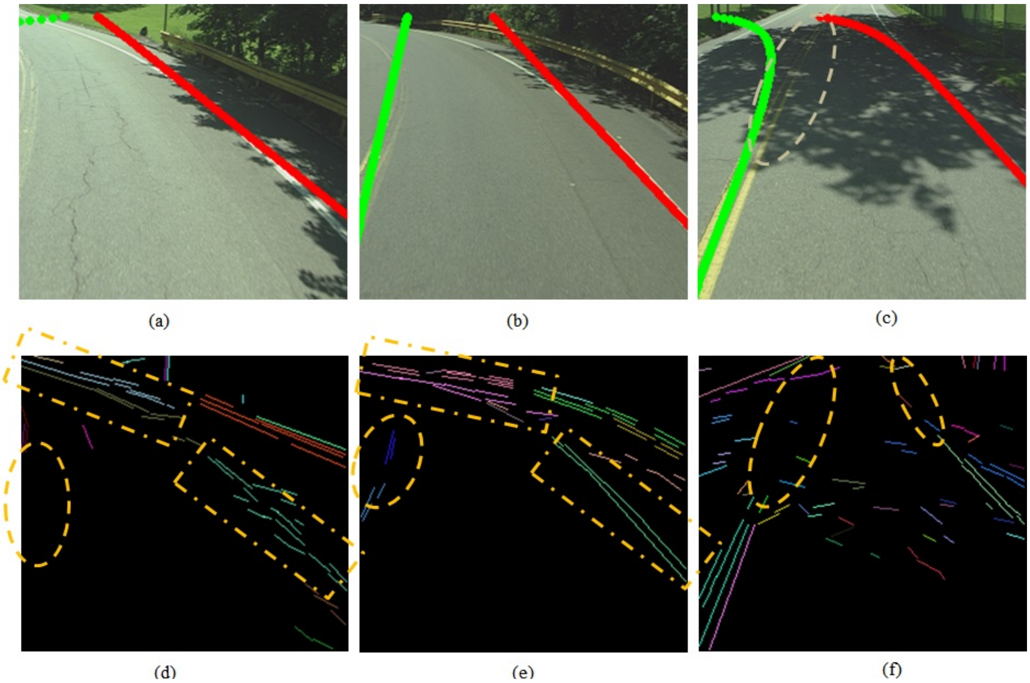


Fig. 10. Some of failure cases of the proposed method on CMU/VASC dataset.

of Fig. 10(f). Even if we adjust the slope difference threshold to handle the situations like Fig. 10(a) and (b), the different threshold may cause some other spurious (infeasible) line segments included in the most probable cluster. In the same way, adjusting the threshold of LSD algorithm may cause other spurious line segments detected .

Fig. 12 shows an example of application of our

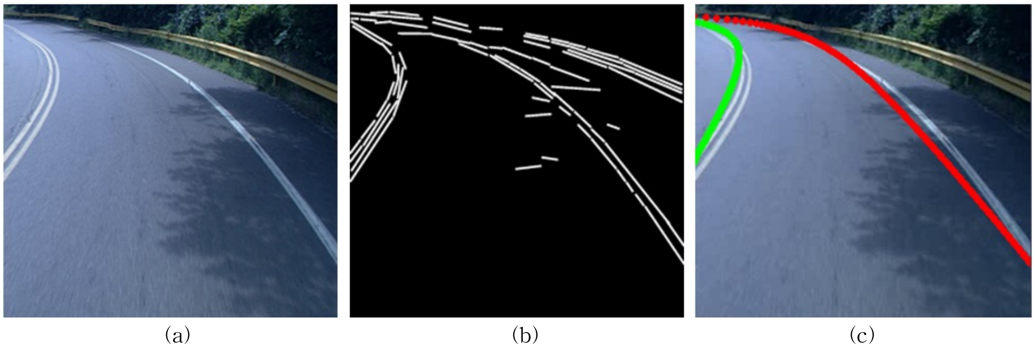


Fig. 11. An example of success case of the proposed method for a steep lane.

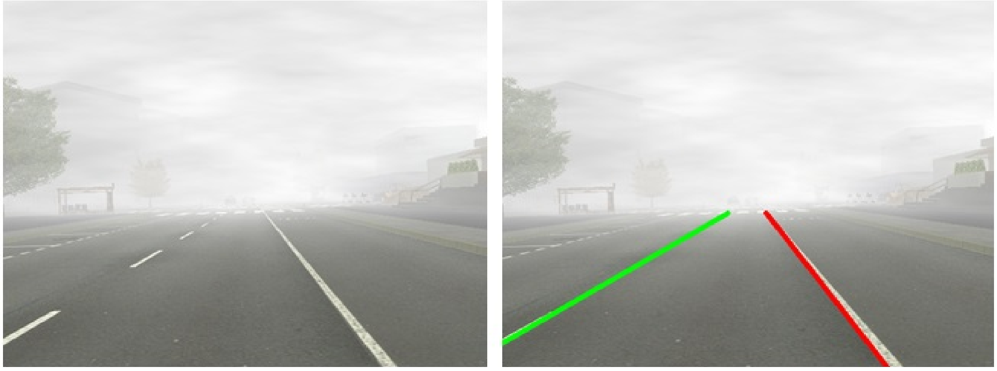


Fig. 12. An example of application of the proposed method to a foggy lane.

proposed method to a foggy lane.

4. CONCLUSION

In this paper, we proposed an efficient and robust method for detecting immediate left and right lane boundaries of the lane in the roads. The proposed method are based on hyperbolic lane model and line segment clustering. The more robust performance is achieved mainly by fitting the hyperbolic lane model with the most probable line segment cluster. The most probable line segment cluster is obtained after clustering line segments extracted by the efficient LSD. The experimental results show that the proposed method performs better compared with other lane detection methods on an CMU/VASC lane dataset. The processing speed of the proposed method on the same dataset was about 7 frames/sec in average on the same dataset.

Since the proposed method depends on horizon localization, line segment extraction, and line segment clustering, currently we are working for improvement of those steps (horizon localization, line segment extraction, and line segment clustering) in order to achieve further performance gaining with respect to the lane detection rate as well as the processing speed.

REFERENCE

[1] J.C. McCall and M.M. Trivedi, "Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation," *IEEE Transactions on Intelligent Transportation*, Vol. 7, No. 1, pp. 20-37, 2006.

[2] A. B. Hille, R. Lerner, D. Levi, and G. Raz, "Recent Progress in Road and Lane Detection:

a Survey," *Machine Vision Applications*, Vol. 25, pp.727-745, 2014.

[3] H. Andrew, S. Lai, H. Nelson, and C. Yung, "Lane Detection by Orientation and Length Discrimination," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol. 30, No. 4, pp. 539-548, 2000.

[4] K.H. Lim, K.P. Seng, and L.-M. Ang, "Improvement of Lane Marks Extraction Technique Under Different Road Conditions," IEEE International Conference on *Computer Science and Information Technology*, Vol. 9, pp. 80-84, 2010.

[5] Y. Fan, W. Zhang, X. Li, L. Zhang, and Z. Cheng, "A Robust Lane Boundaries Detection Algorithm based on Gradient Distribution Features," *Proceedings of the 8th International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 1714-1718, 2011.

[6] A. Parajuli, M. Celenk, and H.B. Riley, "Robust Lane Detection in Shadows and Low Illumination Conditions using Local Gradient Features," *Open Journal of Applied Sciences*, Vol. 3, pp. 68-74, 2013.

[7] E.-J. Lee, "Lane Extaction using Grouped Block Snake Algorithm," *Journal of Korea Multimedia Society*, Vol. 3, No. 5, pp. 445-453, 2000.

[8] X. Youchun, W. Rongben, and J. Shouwen, "A Vision Navigation Algorithm based on Linear Lane Model," *Proceeding of the IEEE Intelligent Vehicles Symposium*, pp. 240-245, 2000.

[9] R.N. Hota, S. Syed, S. Bandyopadhyay, and P. Radhakrishna, "A Simple and Efficient Lane Detection using Clustering and Weighted Regression," *COMAD, Computer Society of India*, 2009.

[10] C. Lipski, B. Scholz, B. Berger, C. Linz, and T. Stich, "A Fast and Robust Approach to Lane Marking Detection and Lane Tracking," *Proceeding of the IEEE Southwest Symposium Image Analysis Interpretation*, pp. 57-60, 2008.

[11] M. Meuter, S. Muller-Schneiders, A. Mika, S. Hold, C. Num, and A. Kummert, "A Novel Approach to Lane Detection and Tracking," *Proceeding of the IEEE International Conference Intelligent Transportation Systems*, pp. 1-6, 2009.

[12] A. Guiducci, "Parametric Model of the Perspective Projection of a Road with Applications to Lane Keeping and 3D Road Reconstruction," *Computer Vision and Image Understanding*, Vol. 73, No. 3, pp. 414-427, 1999.

[13] Z. Wennan, Q. Chen, and W. Hong, "Lane Detection in Some Complex Conditions," IEEE/RSJ International Conference on *Intelligent Robots and Systems*, pp. 117-122, 2006.

[14] Y. Wang, L. Bai, and F. Michael, "Robust Road Modelling and Tracking using Condensation," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 4, pp. 570-579, 2008.

[15] Y. Timar and F. Alagoz, "Lane Detection for Intelligent Vehicles in Challenging Scenarios," *2nd International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 37-43, 2010.

[16] Y. Wang, N. Dahnoun, and A. Achim, "A Novel System for Robust Lane Detection and Tracking," *Signal Processing*, Vol. 92, No. 2, pp. 319-334, 2012.

[17] M. Ally, "Real-Time Detetion of Lane Markers in Urban Streets," *IEEE Intelligent Vehicles Symposium*, pp. 7-12, 2008.

[18] Y. Wang, E.K. Teoh, and D. Shen, "Lane Detection and Tracking using B-Snake," *Image and Vision Computing*, Vol. 22, No. 4, pp. 269-280, 2004.

[19] H. Xu, X. Wang, H. Huang, K. Wu, and Q. Fang, "A Fast and Stable Lane Detection

Method based on B-spline Curve," *IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*, Vol. 1-3, pp. 1036-1040, 2009.

[20] R.G. Gioi, J. Jakubowicz, J.M. Morel, and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 4, pp. 722-732, 2010.

[21] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679-698, 1986.

[22] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, No. 2, pp. 111-122, 1981.

[23] G.T. Shrivakshan and C. Chandrasekar, "A Comparison of Various Edge Detection Techniques used in Image Processing," *International Journal of Computer Science Issues*, Vol. 9, Issue 5, pp.269-276, 2012.

[24] T. Suttorp and T. Bucher, "Robust Vanishing Point Estimation for Driver Assistance," *IEEE Conference on Intelligent Transportation Systems*, pp. 1550-1556, 2006.

[25] K. Madsen, H.B. Nielsen, and O. Tingleff, *Methods for non-linear least squares problems*, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2004.

[26] lmfit - a C library for Levenberg-Marquardt least-squares minimization and curve fitting, http://apps.jcns.fz-juelich.de/doku/sc/lmfit , 2014.

[27] CMU/VASC Lane Image Database, http://vasc.ri.cmu.edu/idb/html/road/may30_90, 2014.

[28] CMU/VASC Lane Image Database, http://vasc.ri.cmu.edu/idb/html/road, 2014.

Tan-Hung Duong

He received the B.S. degree in Computer Science from HoChi Minh City University of Science, Vietnam, in 2009. Currently, he is a master student at School of Electronic Engineering, Soongsil University, Korea. His research interests include image processing, visual traffic motoring, and face recognition.

Sun-Tae Chung

He received B.E. degree from Seoul National University, and M.S. degree and Ph.D. degree in Electrical Eng. and Computer Science from the University of Michigan, Ann Arbor, USA, in 1986 and 1990, respectively. Since 1991, he has been with the School of Electronic Eng. at the Soongsil university, Seoul, Korea where he is now a full professor. His research interests include: computer vision, visual surveillance, biometrics, and embedded systems.

Seongwon Cho

He received B.E. degree from Seoul National University, and MS. degree and Ph.D. degree in Electrical and Computer Engineering from the Purdue University, Indiana, USA in 1987 and 1992, respectively. He is a professor of the School of Electronic and Electrical Engineering at the Hongik University, Seoul, Korea. His research interests include image processing, recognition, and intelligent systems.