Robust Lane Detection and Tracking for Real-Time Applications

Chanho Lee, *Senior Member, IEEE*, and Ji-Hyun Moon

***Abstract*—An effective lane-detection algorithm is a fundamental component of an advanced driver assistant system, as it provides important information that supports driving safety. The challenges faced by the lane detection and tracking algorithm include the lack of clarity of lane markings, poor visibility due to bad weather, illumination and light reflection, shadows, and dense road-based instructions. In this paper, a robust and real-time vision-based lane detection algorithm with an efficient region of interest is proposed to reduce the high noise level and the calculation time. The proposed algorithm also processes a gradient cue and a color cue together and a line clustering with scan-line tests to verify the characteristics of the lane markings. It removes any false lane markings and tracks the real lane markings using the accumulated statistical data. The experiment results show that the proposed algorithm gives accurate results and fulfills the real-time operation requirement on embedded systems with low computing power.**

***Index Terms*—Lane detection, lane tracking, line clustering, real-time, region of interest, Kalman filter.**

I. Introduction

RECENTLY, vigorous research on the technologies for intelligent vehicles is increasing. Especially, technologies for the advanced driver assistant system (ADAS) are being applied to commercial vehicles in the interest of safe autonomous driving. The lane-detection functionality plays an important role in the ADAS by providing basic information for drivers such as the lane structure and the positions of other vehicles in the lanes. The detection results should be accurate and promptly provided.

The visual data from a camera is becoming the most common modality due to its high information content, lower cost and operating power, and the ability to use ultra-sonic sensors or radar as auxiliary sensors if necessary [1]. In the vision-based approach for lane detection, the 2-D image captured from a camera (usually mounted behind the front windshield to retrieve lane information) is analyzed by image processing. Vision-based methods can be categorized into two classes: feature-based and model-based.

The model-based methods commonly use mathematical models with parameters to describe the lane structure [2]–[4]. While it has been demonstrated that the model-based methods are more robust against noise than the feature-based methods, they are difficult to implement since they require some prior-known geometric parameters and heavy computation.

The feature-based methods analyze images and detect the gradients of pixel information or the color of patterns to recognize the lane markings [5]–[9]. Otsuka [5] utilized edge images only, and they removed the noises by analyzing histograms and examined the directions of the edge points to find the lane boundaries. Daigavane [6] developed an ant colony optimization (ACO) to link disjointed edges that should be joined; they used the Hough transform to extract the lane markings. Another approach of the feature-based methods utilizes the color information in the images [7]–[9]. After applying color segmentation in the color models such as HSV or YUV, instead of the RGB model, the pixels are classified as either the lane marking class or the road-background class. The methods using gradient information require clarity of lane markings that are easily influenced by shadows and occlusion, while the methods using color information are less affected. However, color consistency depends on the illumination, light reflection, and sensor type.

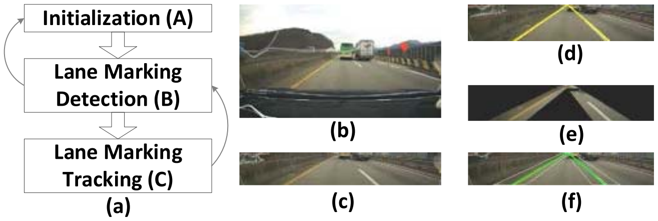

Fig. 1. Procedure of the proposed algorithm: (a) flowchart, (b) resized input image, (c) rectangular ROI, (d) lane markings are detected in the rectangular ROI, (e) Λ-ROI, and (f) lane markings are tracked in the Λ-ROI.

The low complexity of the computation is also required for the algorithm to be implemented for real-time operation on embedded systems of vehicles. Most of the previous works do not meet the requirement and cannot apply for the real world applications unless a vehicle carries a PC.

In this paper, a robust and fast lane detection and tracking algorithm is proposed that consists of three steps: initialization, lane detection, and lane tracking. The lane detection and lane tracking schemes are sophisticated and include many new features such as a colored lane detection, line clustering, a scan-line test, and statistical information tests compared with the authors' previous work. The proposed algorithm verifies the lane markings using the scan-line method that determines the validity of the detected lane markings. It removes the false lane markings and tracks the real lane markings using accumulated statistical data. The improved lane tracking methods in Λ-ROI reduces noisy line segments and increases the detection rate. The experiment results show that the proposed algorithm is very robust in a variety of environments with different types of lane markings, and under problematic weather conditions such as rainy and snowy environments. An implementation result shows that the proposed algorithm works in real-time on embedded systems with low computing power.

II. Proposed Algorithm

The proposed algorithm consists of initialization, lane marking detection, and lane marking tracking, as shown in Fig. 1(a). The lane marking tracking is the main operation process, while the initialization and the lane marking detection set up the rectangular ROI and the Λ-ROI.

Manuscript received October 18, 2016; revised June 28, 2017, December 5, 2017, and January 2, 2018; accepted January 4, 2018. This work was supported by the National Research Foundation of Korea by the Korean Government under Grant (MOE, NRF-2016R1D1A1B01008846) and the EDA tools were supported by IDEC. The Associate Editor for this paper was L. M. Bergasa. *(Corresponding author: Chanho Lee.)*

The authors are with the Electronic Engineering Department, Soongsil University, Seoul 06978, South Korea (e-mail: chlee@ssu.ac.kr; moondi@ssu.ac.kr).

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

Digital Object Identifier 10.1109/TITS.2018.2791572

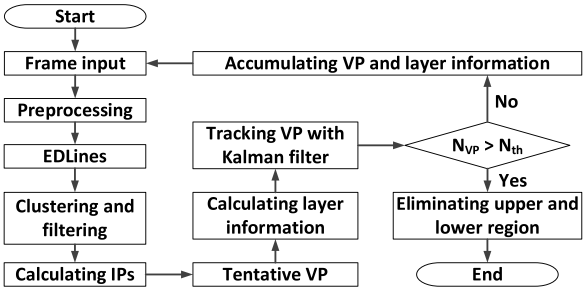
Start
Frame input
Preprocessing
EDLines
Clustering and filtering
Calculating IPs
Tentative VP
Calculating layer information
Tracking VP with Kalman filter
Accumulating VP and layer information
$N_{VP} > N_{th}$
No
Yes
Eliminating upper and lower region
End

Fig. 2. Procedures of initialization using improved adaptive road ROI algorithm.

A. Initialization

The rectangular ROI is configured in the initialization stage, and includes the road region only when removing the region above the vanishing point (VP) and the lower region corresponding to the vehicle in which the camera is attached. The rectangular ROI is determined by an improved algorithm based on the adaptive road ROI determination algorithm [10]. The input image is preprocessed first, as shown in Fig. 2. It is scaled down to a low resolution since the lane detection results are almost the same if the resolution of an image is larger than a certain value. The colored image is transformed to grayscale during the scaling-down process.

The edges are extracted from the initial grayscale images using EDLines method [11]. The edges shorter than a threshold value are removed and the surviving edges are clustered to form line segments. A slope filter is applied in order to only retain the line segments that are parallel to the lane markings. The image is divided into the right half region (RHR) and the left half region (LHR) using a vertical line. A line segment in the RHR and a line segment in the LHR are selected to obtain an intersection point (IP) by extending the line segments. Intersection points are calculated from all of the pairs of line segments, and most of them are found near the actual VP since most of the surviving line segments are almost parallel. The tentative VPs and layer information are accumulated and are tracked using a Kalman filter for a certain number of frames until the number of valid VPs (N_{VP}) becomes larger than a threshold number (N_{th}).

The improved adaptive road ROI algorithm determines the rectangular ROI better than the previous algorithm [10]. The previous algorithm occasionally finds a false VP when the number of line segments from the noises is comparable to that of the real lane markings. In addition, it does not remove the hood of a vehicle when line segments on the hood are parallel to the lane markings due to the illumination. The improved adaptive road ROI algorithm detects the rectangular ROI correctly for all the test videos, while the average detection ratio of the previous algorithm is 94%.

B. Lane Marking Detection

The flowchart of the detected lane markings in the rectangular ROI is shown in Fig. 3. After the rectangular ROI has been determined, the lane markings are searched in this region. The color image of the rectangular ROI is transformed into grayscale. one that consists of the pixels with the value (V) component in the HSV color model.

The EDLines scheme extracts line segments from the grayscale image [11]. The EDLines outperforms the Canny edge detection and the PPHT that were used in our previous algorithm [12] in terms of both accuracy and performance; furthermore, it increases the detection rate and reduces the computation time. The line segments can be filtered with a slope filter since the line segments from the lane markings point to the VP and have limited slope values. The line segments that pass the slope filter are parallel to the lane markings. The rectangular ROI is divided into the LHR and the RHR by the vertical line going through the VP of the image, and the filtering slopes are different in the LHR and RHR. The slope filters remove many noisy line segments efficiently [12].

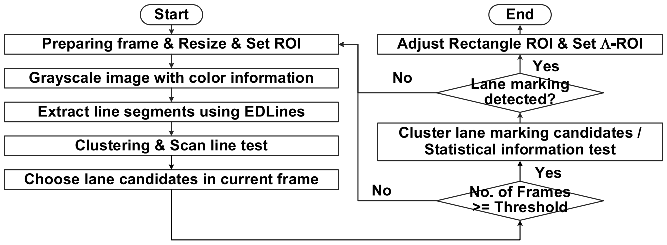
Start
Preparing frame & Resize & Set ROI
Grayscale image with color information
Extract line segments using EDLines
Clustering & Scan line test
Choose lane candidates in current frame
No. of Frames >= Threshold
No
Yes
Cluster lane marking candidates / Statistical information test
Lane marking detected?
No
Yes
Adjust Rectangle ROI & Set Λ-ROI
End

Fig. 3. Algorithm for the detecting of lane markings in the rectangular ROI.

The grayscale information works well for the lane marking detection for which the white lane markings on the road background are used; however, it is difficult to detect colored lane markings (e.g., yellow or blue colors) using grayscale information due to the low contrast, especially when the colored lane markings are not clear. The proposed algorithm includes an adaptive method for detecting colored lane markings using the YUV color model, as colored lane markings are much easier to detect in the YUV color space [13]. The yellow and blue lane markings have higher contrasts in the color difference images, and the color difference information is added to the grayscale image so that the lane marking patterns are amplified; this increases the detection rate of the lane markings. However, the colored lane marking patterns are not clear in the color difference image if the background color appears yellow due to a street light at night or in a tunnel with yellow lights. In this case, the color information is not added to the grayscale image since it increases the noise components in the grayscale image.

A lane marking is a painted bright and thick line, so that it produces more than two edges according to the thickness of the lane markings, curved lane markings, and unclear or distorted images. Obviously, line segments that are extracted from the boundaries of a lane marking will have similarities regarding the slope and the position, which are obtained using the endpoint positions. This feature can be used to distinguish the line segments that originates from lane markings and those that originates from noises. The hierarchical agglomerative clustering (HAC) method is applied here to cluster the line segments so that all the line segments from a single lane marking are merged into a single line segment. The merging process consists of a vertical clustering and a horizontal clustering. The vertical clustering merges the line segments extracted from the same side of a lane marking and the horizontal clustering merges the line segments from the opposite sides of a lane marking. A merged line segment has a weight factor of larger than 1 depending on the number of the horizontal merging.

Several line segments often remain after the clustering due to the line segments caused by noises such as cracks, new pavement, shadows, and other vehicles. The unique characteristics of the lane markings are their greater brightness than the background (or road) and their appearance at similar locations continuously. If horizontal lines are drawn through a merged cluster and the intensities are scanned along the lines, the low-high-low intensity patterns can be obtained, as shown in Fig. 4. The derivatives of the intensities show both a positive peak and a negative peak at the edge points of the lane markings. The intensity patterns and their derivatives can be exploited to verify whether or not the line segments are extracted from the lane markings by comparing the average intensity of the pixels inside the two peaks with that of the pixels outside the two

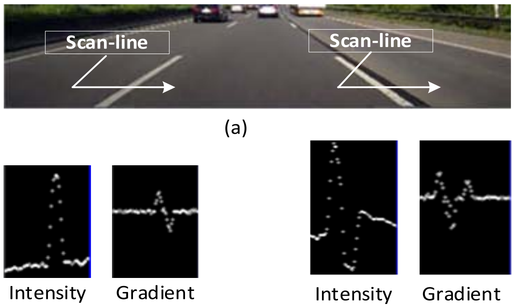

Fig. 4. Illustration of the finding of a pair of edge points: (a) directions of scan lines, (b) intensity and gradient profiles of the left scan line, and (c) intensity and gradient profiles of the right scan line.

peaks. If the difference between the two intensities is higher than a particular threshold, the line segment satisfies the intensity condition. The verification process described above for the lane markings, called a "scan-line test" is useful for the detection of lane markings. The test scans virtual horizontal lines that cross a line segment to find the pairs of edge points on a real lane marking, as shown in Fig. 4(a). The scan-line test starts from the virtual horizontal line that crosses the low endpoint, and it is repeated along the equidistant horizontal lines toward the upper endpoint. If a surviving line segment passes a specified number of the scan-line test, it is considered as a real lane marking candidate in the current frame. The line segment is collected for a further processing step. It is possible that none of the line segments satisfies the scan-line test when the road condition is very poor; for example, the lane markings are eroded, they are influenced by the light reflections on the road during rainy nights., or they are not painted on the road boundary.

The line segments from other vehicles, shadows, or buildings may survive the slope filtering and the clustering if they are parallel to the lane markings. The positions of the lane markings in consecutive frames change only slightly under normal driving conditions. The line segments from real lane markings will appear continuously within a limited range, while those from noises will appear in several contiguous frames only, or the positions vary. The proposed algorithm accumulates the lane marking candidates for a certain number of frames (e.g., 30), and it clusters the candidates in similar positions, which is called "inter-frame clustering." For the inter-frame clustering, the algorithm accumulates statistical data of the slopes, the positions of the lower IPs and VPs, and lane widths. The algorithm checks the lane width (symmetry condition) and the positions of the lower IPs (similarity condition) loosely for the pairs of lane marking candidates that are selected from the LHR and RHR of the rectangular ROI, respectively Furthermore, the upper IP of the pair of lane marking candidates should be in a position that is similar to that of the VP (VP condition). The test is repeated for all pairs.

The Λ-ROI is set up when each line segment from the lane markings in the RHR and LHR is respectively determined. The line segment is extended to the upper and lower boundaries of the rectangular ROI. Eight vertices are determined for the Λ-ROI, so that two distorted trapezoids are formed and the extended line segments are located in the centers; consequently, the detection rate is increased when compared with that from the triangular shape of the previous algorithm [10].

C. Lane Marking Tracking

After the Λ-ROI has been set up, the lane markings can be detected and tracked efficiently in the subsequent frames, since the lane markings in the next frame will be found in the Λ-ROI under normal driving conditions. Most of the noises are excluded since the Λ-ROI includes only the road region and the area is small, and the calculation amount is markedly reduced as a result. While the Λ-ROI has a triangular shape with fixed boundaries in the original scheme, the shape has been changed to a distorted trapezoid and the boundaries are variable in the modified Λ-ROI scheme. The short lane markings in the upper area are more easily found in the modified Λ-ROI, while they are often missed in the old scheme because the Λ-ROI is set up based on the lane markings in the previous frame and the extended upper ends of the lane markings in the current frame are not likely to coincide with the vertices of the triangle. In addition, the modified Λ-ROI can become wider if a lane marking in either side is not detected in the previous frame. This enables the lane marking to be detected again when the lane width is changed or the vehicle shifts horizontally beyond the allowed range.

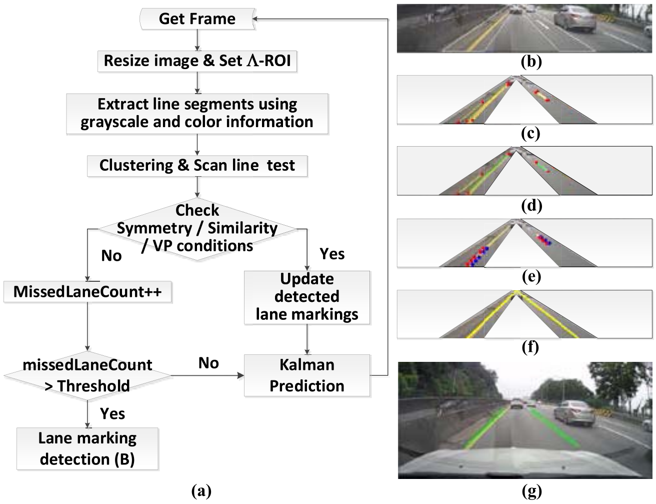
Get Frame
Resize image & Set Λ-ROI
Extract line segments using grayscale and color information
Clustering & Scan line test
Check Symmetry / Similarity / VP conditions
No
Yes
MissedLaneCount++
Update detected lane markings
missedLaneCount > Threshold
No
Kalman Prediction
Yes
Lane marking detection (B)
(a)
(b)
(c)
(d)
(e)
(f)
(g)

Fig. 5. Algorithm for lane-marking tracking in the Λ-ROI. (a) Flowchart, (b) Λ-ROI setup, (c) Line segments extraction in the Λ-ROI, (d) Line segments clustering, (e) Scan-line test, (f) Detected lane markings, and (g) Final result combined with Kalman filter output.

Fig. 5 shows the algorithm for the tracking of the lane markings in the Λ-ROI, which is similar to the lane marking detection stage. First, the image frame is obtained and resized. The Λ-ROI is then configured and set as an active region. The line segments are extracted in the Λ-ROI using the grayscale image with the color information. After the clustering and the scan-line tests, the similarity, symmetry and VP conditions are checked and the weight factors of line segments are adjusted for a robust validation of the lane marking detection. If more than one pair of lane marking candidates with the same total weight factor satisfies the conditions, the pair with the smaller distance, called the "inner pair," is chosen.

If a lane marking is not detected in the Λ-ROI, a Kalman filter is employed for the lane marking. The slopes and the lower IPs of the detected lane markings are given to the Kalman filter to track the lane markings in the next frame. The predicted line from the Kalman filter is used when a line segment that originates from the lane marking is not detected.

D. Comparison of the Proposed Algorithm With Others

The proposed algorithm is compared with other algorithms that have shown relatively good detection results and have been reported recently. Son [14] proposed an illumination invariant lane detection algorithm that configures ROI by removing the upper region above the VP obtained using Canny edge detector and Hough transform in a single frame. It occasionally results in an incorrect VP when the ROI includes surviving noisy line components, although it works in most cases, and it does not remove the hood and the interior of the vehicle that appear in the lower part of the image. The proposed adaptive ROI algorithm accumulates and tracks the VPs over a certain number of frames so that noisy components are eliminated, and removes the unnecessary part of the vehicle in the lower regions of the images. Son utilized the Cb component of YUV color space to detect the yellow lane markings. The binary value of a pixel from the intensity and the Cb value are combined using OR operation. Both the intensity and the yellow value are below the threshold when the yellow lane markings have eroded or are covered with heavy dusts in our experiments. The proposed algorithm calculates the grayscale intensity and the |U-V| values in YUV color space and adds them to boost the values for the yellow markings; also, the yellow lane markings are detected even if they are eroded or covered with heavy dusts. Son utilized the connected component clustering method [15] and least square line fitting algorithm. The initial data of the clustering in Son's algorithm are binary pixels while those of the proposed algorithm are line segments from edge detection. Son's algorithm cannot avoid wrong detection due to noisy line segments, although it is robust against the illumination effects. The proposed algorithm is robust against most of the noise line segments due to the strong lane marking verification procedure and can recover the missing lane markings due to an improved lane marking tracking scheme using the Kalman filter.

Jung [16] proposed a lane detection algorithm based on spatiotemporal images. While Jung's algorithm can suppress noisy lane points near the lane markings, it has several potential problems. First, the detection is likely to fail when the number of noisy lane points that exceeds a certain threshold. The noisy lane points can be obtained on the road far from the lane markings due to markings on the road, other vehicles, illumination, weather conditions, and so on. Secondly, Jung's lane detection algorithm will results in errors when the width of the lane is increasing or decreasing because it attempts to find vertical lines and depends on the road width. In addition, the spatiotemporal image depends on the speed of the vehicle, which will affect the detection results.

Yoo [17] proposed a lane detection algorithm using gradient-enhanced grayscale images. While it is an efficient algorithm to remove the illumination effects, it also enhances edges of noisy patterns on the road or other vehicles, and is not effective in that cases.

The algorithms mentioned above do not have lane verification procedures and lane tracking schemes which increase the detection rate when the road and weather conditions render it difficult to detect the lanes. The Λ-ROI of the proposed algorithm removes most of the noisy patterns of the roads and the lane verification procedure excludes lane-like cracks or boundary of new pavement near the lane markings. It also reduces the amount of computation. The lane tracking scheme enables to detect lanes even if the lane makings are completely eroded or not painted for a certain number of frames. Table I summarizes the comparison results of the proposed algorithm with Son's and our previous algorithm.

III. Experimental Results

The proposed lane detection algorithm is implemented on a prototyping embedded system using C++ and OpenCV library with Ubuntu 14. The intention is to operate the proposed algorithm in vehicles in real-time. The target embedded system has dual ARM Cortex-A9 processors running at 800 MHz. To evaluate the performance of the implemented software based on the proposed algorithm, 48 video clips (approximately 92,000 frames) captured in USA and Korea using dashboard cameras were collected; they include various weather conditions, illuminations, and lane markings [18].

TABLE I

Comparison of the Proposed Algorithm With Others

Algorithms	Son [14]	Ding [12]	Proposed
ROI upper boundary	VP with HT	Fixed position	VP with Clustering and tracking scheme
ROI lower boundary	N/A	Fixed position	Layer information
Color lane detection	(Binary Intensity) OR (Binary Cb)	N/A	Intensity(V) + Cb+Cr → Binarization
Line segment detection	Pixel-based. Connected component clustering and fitting	Edge-based. Canny+HT, inner line selection	Edge-based. EDLines, Line clustering
Lane verification	Slope and Intercept	Slope	Scanline test (req.), Slope/ Lane width/ IP
Lane tracking	N/A	Kalman filter in Λ-ROI (triangular and fixed boundary)	Kalman filter in Λ-ROI (distorted trapezoidal and adaptive boundary)

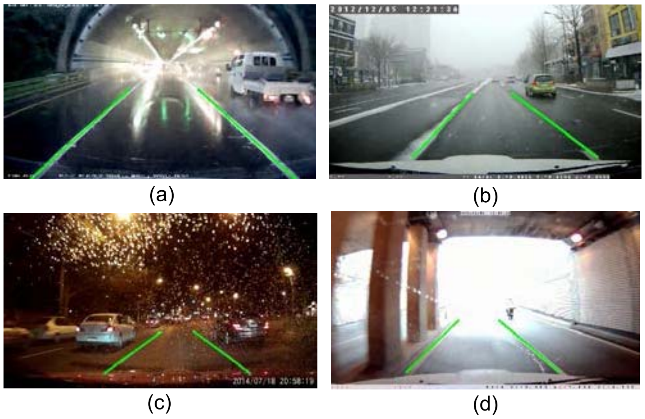
(a) (b) (c) (d)

Fig. 6. Lane marking detection results under various driving conditions: (a) rainy day, (b) snowy day, (c) rainy night, (d) exiting tunnel.

The ground truth is obtained manually by drawing two boundary lines along the long edges of lane markings for each frame and extending them to the upper and lower boundaries of the road ROI if lane markings exist. The boundary lines are drawn with the tolerance of 1 pixel considering the quantization errors. Virtual boundary lines are generated based on the boundary lines of the previous frame, and the distance between them is increased by 50% considering the horizontal shift of the vehicle if the lane markings do not exist, such as when the lane markings are not seen on the road by any reasons or the road is under construction. The road ROI is divided into 3 sections in the vertical direction and the lane detection is determined to be correct when the detected lane markings are located between the boundary lines in the middle and lower sections. The detection rate is the number of correct frames divided by the total number of frames under evaluation.

Fig. 6 shows the successful lane marking detection results for which the implemented software and the video clips of various weather and road conditions are used. Both the successful lane marking detection with the rain drops on the windshield during a rainy night that is shown in Fig. 6(c) and that with the white-out on the tunnel exit shown in Fig. 6(d) verify the robustness of the proposed algorithm. The proposed algorithm cannot, however, detect lane markings when they are fully covered with snow and when they

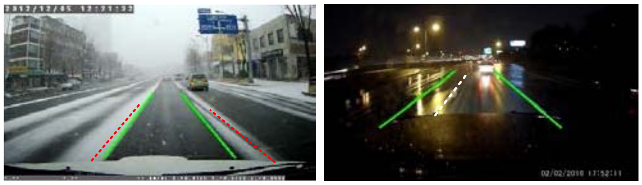

Fig. 7. Examples of false detection results (dotted lines are real lane markings).

TABLE II

DETECTION RATES IN RECTANGULAR ROI AND IN THE Λ-ROI WITHOUT AND WITH TRACKING FUNCTION USING KALMAN FILTER

Detection Rate		Rectangular ROI [%]	Λ-ROI without tracking [%]	Λ-ROI with tracking [%]
Day	Clear (5)	80.5	88.8	99.3
	Rainy (5)	73.1	80.6	96.9
	Snowy (5)	71.3	85.9	93.0
Night	Clear (5)	73.4	89.7	97.7
	Rainy (5)	43.0	66.3	94.4
	Snowy (2)	54.4	78.4	92.2

TABLE III

RESULTS OF THE PROPOSED ALGORITHM COMPARED WITH OTHER WORKS

	Detection rate [%]					Performance	
	Day			Night		Processing time / frame [ms]	Platform, CPU, Image resolution
	Clear	Rainy	Snowy	Clear	Rainy		
Aly [2]	96.3[7]	N/A				20	PC, Core 2 @ 2.4 GHz, 640 x 480
	72.94 [16]						
Son[14]	93.1 (86)*	89	-	94.3	93 (80)*	33	PC(unknown) 1280x800
Jung[16]	88~98	-	-	87.7	-	**3.9**	PC, 2.8GHz, 4GB 640 x 480
Yoo [17]	96.36	96.19	-	96.5	-	50	PC (unknown) 1280 x 800
Liu [19]	-	**98.67**	-	94.06	-	200	PC, 2.3 GHz, 640 x 480 / 256 x 240
Proposed	**99.0**	96.9	**93.0**	**98.0**	**93.6**	35.4** (7.4)	ARM A9 @800 MHz, 1920x1080 (PC, 3.4GHz)

* Detection rates according to our metric

** The time for decoding video data is not included

are not seen due to reflected street lights in rainy nights, as shown in Fig. 7.

Table II shows the outstanding effects of the Λ-ROI and the Kalman filter in the lane detection applications. The rows represent detection rates in clear, rainy, and snowy weather conditions at day and night, respectively. The numbers in parentheses beside the weather condition labels represent the number of videos used for the experiments. The first column represents the detection rates in the rectangular ROI, and the second column represents those in the Λ-ROI without the tracking function using the Kalman filter. Table II shows that the Λ-ROI is very effective for reducing the noise components and increasing the detection rate; the Λ-ROI increases the detection rate by 8.3% ∼ 24% (14.7% in average). The third column represents the detection rate with the Λ-ROI and the tracking function using the Kalman filter. The tracking function increases the detection rate by 7.1% ∼ 28.1% (14.2% in average). Snow often covers the lane markings even after the road has been cleaned, and the white color of the snow generates strong noise components. The reflection of the street lights on the road during rainy nights renders the lane marking detection difficult since the lane markings are not even visible to the human eyes. Table II shows that the Λ-ROI and the tracking function are responsible for major contributions regarding the improvement of the lane detection rate.

The performance of the proposed algorithm is compared with that of the previous works [2], [14], [16], [17], [19]. Table III shows the comparison results of the detection rates and the calculation time per frame. The values in Table III are copied or estimated using the data of the reference papers mentioned above, although the video datasets are different. Aly reported the detection rate of 96.3 % with his own database on a clear day [2]; however, the detection rate of Aly's algorithm is 72.94 % according to Yoo whose database includes video clips during the night and the rainy day [17]. In addition, the different metric of calculating detection rate is another issue. The detection rates of Son's and Yoo's results are comparable to the results of the proposed algorithm for the conditions of clear and rainy days and nights. However, the measured detection rates of Son's work according to our metric are 86% and 80% in clear day and rainy night, respectively, compared with those of 93.1% and 93% according to Son's metric. It shows that our metric is stricter than Son's. Table III shows that the detection rates of the proposed algorithm are superior to those of the other works, and more road conditions are considered in this work than in the other works. Especially, video clips of roads with snow are not used in other works. The computation time is even superior to that of the other algorithms. With the embedded processor, the results showed that the proposed algorithm provides much faster processing speed than the other algorithms with powerful PC environments. Practically, the proposed algorithm can be employed in real world applications such as ADAS on embedded systems.

IV. CONCLUSION

An efficient and robust lane detection and tracking algorithm is proposed in this paper. The proposed algorithm consists of initialization, lane detection, and lane tracking. The initialization finds a rectangular ROI using the VP, and the lane detection finds the Λ-ROI, which is a distorted trapezoidal region surrounding the lane markings. The lane-tracking detects and tracks the lane markings in the Λ-ROI. Many sophisticated methods such as the introduction of color cues, line clustering, scan-line test, lane verification, and variable Λ-ROI are proposed in this study, and the proposed algorithm shows better performance for road conditions with noisy components than others. The proposed algorithm is verified using 48 video clips, which represent 12 road and weather conditions. The computation time satisfies the real-time operation with embedded processors running at 800 MHz. The detection rate and the computation time for the proposed algorithm are compared with those of other works, and it is demonstrated that the proposed algorithm is superior to them.

REFERENCES

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

[2] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Vehicles Symp.*, Eindhoven, The Netherlands, Jun. 2008, pp. 7–12.

[3] J. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[4] S. Zhou, Y. Jiang, J. Xi, J. Gong, G. Xiong, and H. Chen, "A novel lane detection based on geometrical model and Gabor filter," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, San Diego, CA, USA, Jun. 2010, pp. 59–64.

[5] Y. Otsuka, S. Muramatsu, H. Takenaga, Y. Kobayashi, and T. Monj, "Multitype lane markers recognition using local edge direction," in *Proc. IEEE Intell. Vehicles Symp.*, vol. 2. Versailles, France, Jun. 2002, pp. 604–609.

[6] P. M. Daigavane and P. R. Bajaj, "Road lane detection with improved canny edges using ant colony optimization," in *Proc. ICETET*, Nov. 2010, pp. 76–80.

[7] T.-Y. Sun, S.-J. Tsai, and V. Chan, "HSI color model based lane-marking detection," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Toronto, ON, Canada, Sep. 2006, pp. 1168–1172.

[8] Z. Teng, J.-H. Kim, and D.-J. Kang, "Real-time lane detection by using multiple cues," in *Proc. Int. Conf. Control Autom. Syst.*, Gyeonggi-do, South Korea, Oct. 2010, pp. 2334–2337.

[9] L. N. P. Boggavarapu, R. S. Vaddi, H. D. Vankayalapati, and J. K. Munagala, "A robust multi color lane marking detection approach for Indian scenario," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 5, pp. 71–75, May 2011.

[10] D. Ding, C. Lee, and K.-Y. Lee, "An adaptive road ROI determination algorithm for lane detection," in *Proc. TENCON*, Xi'an, China, Oct. 2013, pp. 1–4.

[11] C. Akinlar and C. Topal, "Edlines: Real-time line segment detection by Edge Drawing (ed)," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 2837–2840.

[12] D. Ding and C. Lee, "Efficient real-time lane detection algorithm using V-ROI," *J. IKEEE*, vol. 16, no. 4, pp. 349–355, Dec. 2012.

[13] Q. Lin, Y. Han, and H. Hahn, "Real-time lane departure detection based on extended edge-linking algorithm," in *Proc. 2nd Int. Conf. Comput. Res. Develop.*, Kuala Lumpur, Malaysia, May 2010, pp. 725–730.

[14] J. Son, H. Yoo, S. Kim, and K. Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Syst. Appl.*, vol. 42, pp. 1816–1824, Oct. 2014.

[15] R. N. Hota, S. Syed, S. Bandyopadhyay, and P. Radhakrishna, "A simple and efficient lane detection using clustering and weighted regression," presented at the COMAD, Mysore, India, Jun./Jul. 2009.

[16] S. Jung, J. Youn, and S. Sull, "Efficient lane detection based on spatiotemporal images," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 289–295, Jan. 2016.

[17] H. Yoo, U. Yang, and K. Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1083–1094, Sep. 2013.

[18] *DSDLDE v.0.9: Video Clips for Lane Marking Detection*. [Online]. Available: https://drive.google.com/file/d/1315Ry7isciL-3nRvU5SCXM_-4meR2MyI/view?usp=sharing

[19] G. Liu, S. Li, and W. Liu, "Lane detection algorithm based on local feature extraction," in *Proc. Chin. Autom. Congr.*, Changsha, China, Nov. 2013, pp. 59–64.