공학박사학위논문

Monocular Camera-based Real-time Image Recognition for Autonomous Vehicle

무인 자율주행 차량을 위한 단안 카메라 기반 실시간 주행 환경 인식 기법에 관한 연구

2014년 2월

서울대학교 대학원

전기·정보공학부

강 승 남

Monocular Camera-based Real-time Image Recognition for Autonomous Vehicle

무인 자율주행 차량을 위한 단안 카메라 기반 실시간 주행 환경 인식 기법에 관한 연구

지도교수 서승우

이 논문을 공학박사 학위논문으로 제출함

2014년 2월

서울대학교 대학원

전기·정보공학부

강승남

강승남의 공학박사 학위논문을 인준함

2014년 2월

위원장 최진영 (인)

부위원장 서승우 (인)

위원 조남익 (인)

위원 김현진 (인)

위원 김창수 (인)

Abstract

"Homo Faber" refers to humans as controlling the environments through tools. From the beginning of the world, humans have created tools for chasing the convenient life. The desire for the rapid movement let the human ride on horseback, made the wagon and finally made the vehicle. The vehicles allowed humans to travel the long distance very quickly as well as conveniently. However, since human being itself is imperfect, plenty of people have died due to the car accident, and people are dying at this moment. The research for autonomous vehicle has been conducted to satisfy the human's desire of the safety as the best alternative. And, the dream of autonomous vehicle will come true in the near future.

For the implementation of autonomous vehicle, many kinds of techniques are required, among which, the recognition of the environment around the vehicle is one of the most fundamental and important problems. For the recognition of surrounding objects many kinds of sensors can be utilized, however, the monocular camera can collect the largest information among sensors as well as can be utilized for the variety of purposes, and can be adopted for the various vehicle types due to the good price competitiveness. I expect that the research using the monocular camera for autonomous vehicle is very practical and useful.

In this dissertation, I cover four important recognition problems for autonomous driving by using monocular camera in vehicular environment. Firstly, to drive autonomously the vehicle has to recognize lanes and keep its lane. However, the detection of lane markings under the various illuminant variation is very difficult. Nevertheless, it must be solved for the autonomous driving. The first research topic is the robust lane marking extraction under the illumination variations for multilane detection. I proposed the new lane marking extraction filter that can detect the imperfect lane markings as well as the new false positive cancelling algorithm that can eliminate noise markings. This approach can extract lane markings successfully even under the bad illumination conditions. Secondly, the problem to tackle is, if there is no lane marking on the road, then how the autonomous vehicle can recognize the road to run? In addition, what is the current lane position of the road? The latter is the important question since we can make a decision for lane change or keeping depending on the current position of lane. The second research is for handling those two problems, and I proposed the approach for the fusing the road detection and the lane position estimation. Finally, to drive more safely, keeping the safety distance is very important. Measuring accurate inter-vehicle distance by using monocular camera and line laser is the third research topic. To measure the inter-vehicle distance, I illuminate the line laser on the front side of vehicle, and measure the length of the laser line and lane width in the image. Based on the imaging geometry, the distance calculation problem can be solved with accuracy.

There are still many of important problems remaining to be solved, and I proposed some approaches by using the monocular camera to handle the important problems. I expect very active researches will be continuously conducted and, based on the researches, the era of autonomous vehicle will come in the near future.

Keywords: lane marking extraction, road detection, lane position estimation, inter-vehicle distance measurement.

Student number: 2010-30798

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivations

The research for autonomous vehicle has been conducted to satisfy the human’s desire of the safety as the best alternative. And, the dream of autonomous vehicle will come true in the near future. For the implementation of autonomous vehicle, many kinds of techniques are required, and among them, the recognition of the environment around the vehicle is one of the most fundamental and important problems. For the recognition of surrounding objects many kind of sensors can be utilized, however, the monocular camera can collect the largest information among sensors as well as can be utilized for the variety of purposes, and can be adopted for the various vehicle types due to the good price competitiveness. The research by using the monocular camera for autonomous vehicle is very practical and useful. In this dissertation, three important recognition problems for autonomous driving are proposed by using monocular camera in vehicular environment.

Lane marker extraction is the fundamental process that provides candidates of lane markers for the lane fitting process in a lane detection system. Most of the existing approaches extract lane markers by using either particular filters or heuristic algorithms. In both of these, the threshold values play a critical role in detectability. The challenges to conventional approaches are the illumination variance, such as cluttered shadow and sunset, which may degrade detectability. The main reason of the detectability degradation is the inaccurate determination of threshold values under illumination variations.

One of the most difficult parts for road detection based on monocular images is the road boundary determination under the illuminant-variant environments since the color of road and the background outside the road under the shadow is often very similar thus hard to be distinguished.

Inter-vehicle distance measuring technique is widely used for Driving Assistance System (DAS) such as Adaptive Cruise Control (ACC) and Forward Collision Warning (FCW) which are useful for safety enhancement as well as driving convenience. However, Measuring distance based on monocular camera is known as a difficult problem.

1.2 Contributions and Outline of the Dissertation

1.2.1 Illumination-Tolerant Lane Marking Extraction for Multilane Detection

In this paper, a new filter as well as an extraction algorithm that can determine the threshold values efficiently was proposed, while tolerating the illumination variance. As a new filter, named the *cone hat filter*, which is a modified top hat filter that provides higher extraction performance, as compared to the top hat filter in our experiments. As a new extraction algorithm, the *reiterated extraction* is proposed, which is a set of sub algorithms that determine the threshold adaptively, and filter out false positives accurately. As a result, the proposed algorithm outperforms other algorithms in more illuminated scenarios. For overall system performance evaluation, extensive experiments are conducted by comparing our approach with five conventional approaches in various scenarios, including not shaded, shaded, sunset, night-time, curve, up-downhill, urban road and tunnel passing.

1.2.2 Fusing Road Detection and Lane Position Estimation for the Robust Road Boundary Estimation

In this paper, the algorithm for robust road detection based on the information fusion between the road detection and the lane position estimation is proposed. For the robust road boundary determination, the initial road detection is conducted by using chromaticity-based flood-fill method, then the road boundary is estimated by utilizing the results of lane position estimation based on the cross-ratio. The proposed algorithm shows more robust detection result as compared to the original road detection result. In addition, the proposed algorithm can provide very useful information about the current lane position and the total number of lanes on the road.

1.2.3 Accurate Inter-Vehicle Distance Measurement based on monocular camera and line laser

In this paper, a new method providing accurate longitudinal inter-vehicle distance measurement by using a monocular camera and a line laser is proposed. By utilizing the proposed imaging geometry based on the pin-hole model, accurate inter-vehicle distance can be measured. Experimental results show that the proposed method outperforms the disparity-based method and the error in distance measurement is low enough even in nighttime as much as daytime.

Chapter 2

Illumination-Tolerant Lane Marking Extraction for Multilane Detection

2.1 Introduction

Lane detection systems use visual information to autonomously detect single or multiple lanes on roads. This method is currently applied to Autonomous Driving Assistance Systems (ADAS), such as the Lane Departure Warning (or Prevention) System (LDWS/LDPS), Overtaking Assistance System (OAS), and Parking Assistance System (PAS). In addition, in the near future, it will be an important part of the unmanned autonomous driving system. Lane detection systems are in general comprised of three main components: the lane marker extraction function collects lane marker information from input images, the lane fitting function finds lanes with provided lane marker candidates based on the lane model, and finally, the lane tracking function is utilized for robust and stable lane detection. Among these, lane marker extraction is the most fundamental and important, since the final lane detection result can vary, depending on how accurate lane markers are extracted, and to what extent false positives are excluded.

Previous research on lane detection systems deals with the problem of how to extract lane marker candidates [1-15]. Due to its simplicity, most research has adopted edge detection, in conjunction with thresholding. Many well-known algorithms have been applied, from Sobel/Canny/Prewitt/Roberts' algorithm[1], to a morphology-based algorithm [2]. To increase the contrast between lane marker and road intensity in gray scale images, an RGB combination ratio for converting images from color to gray scale was proposed by Chang *et al.*[3]. Wu *et al.* used YIQ space instead of RGB space [4]. Liu *et al.* utilized Otsu's binarization algorithm for the histogram-equalized image in HSV space [5]. A gradient magnitude and orientation based algorithm was proposed by [6-8]. Since a lane marker basically has a uniform gradient magnitude and orientation, the gradient-based detection algorithm can distinguish lane markers from backgrounds fairly well. Satzoda *et al.* proposed an edge map generated by using a gradient angle histogram, and the lane marker is identified based on a Hough line transform [6]. Chen *et al.* also utilized the gradient orientation, named gradient direction (GD)) feature and lane markers are detected by conducting MAP estimation with GD Gaussian distribution [7]. Deusch *et al.* adopted a gradient of ridge features (low-high-low intensity patterns), called double gradients, tensor orientation, and coherence which are used for thresholding [8]. A steerable filter was used by Shang *et al.* [11]. The filter has predefined orientations (e.g. 0°, 30°, 60°, 90°, and 120°), and each filter only detects lane markers that have the matched orientation. Though the filter shows a robust and satisfactory performance, the orientation of lane marking is hard to determine. Therefore, the authors estimated the orientation of lane markers by using the vanishing point. Template matching in Inverse Perspective Mapping (IPM) images was studied by Linarth *et al.* [14]. They generated several synthetic lane marking template images, before comparing the distance of the Histogram of Oriented Gradient's (HoG) features between the template and the captured image, by moving the window. For lane marker identification of the captured image, the PDF was calculated by using the likelihood function. Borkar *et al.* proposed the ground truth image generation by using a *time slicing* process [9].

The *time slicing* process collects the lane detection results at pre-defined positions of input images during given time periods, and interpolation among the detected points was conducted with IPM coordinates. This is then regarded as the ground truth image. The ground truth image is compared with the other lane detection results, generated by the RANSAC algorithm in the IPM image. A visual attention-based lane detection algorithm [10], as well as double threshold (low and high)-based [12], stereo vision-based [13], and multiple constraints-based algorithms [15], were proposed.

However, while there has not been an abundance of research conducted to show the performance of the lane marker extraction function, some research has focused on improving the performance of a typical lane marker extraction. Pollard *et al.* proposed a method for long distance lane markers by using a two (low-high) thresholds combination [12], Popescu *et al.* proposed a method for double line detection, by counting transitions between marker and non-marker pixel sequences [16], and Sun *et al.* proposed a method for color lane marker detection based on the HSI color model [17]. We specially conduct experiments under illumination variations, to measure the illumination-tolerant characteristic with five conventional lane marking extraction algorithms. The contribution of this paper can be summarized as follows:

- *Proposing an illumination tolerant lane marker extraction*: The proposed approach is not much degraded under illumination variations, as compared to the other algorithms.

- *Proposing a new filter*: The proposed *cone hat* filter, modified from the top hat filter [21], provides better performance in our experiments.

- *Proposing a new algorithm for the enhancement of filter detectability*: The proposed *reiterated extraction* algorithm determines the threshold adaptively, and conducts false positive cancelling.

- *Extensive experiments in various scenarios*: The performance of the algorithms are tested in various illuminations (not shaded, shaded, sunset, night-time, tunnel passing, and rainy day), and geometrical changes (curve, and up-downhill).

The remainder of the paper is structured as follows. Section II describes the proposed lane marker extraction filter. Section III outlines an overview of the proposed lane marker extraction algorithm and explains the sub-algorithms in detail. Section IV evaluates the experimental results, and Section V concludes our work.

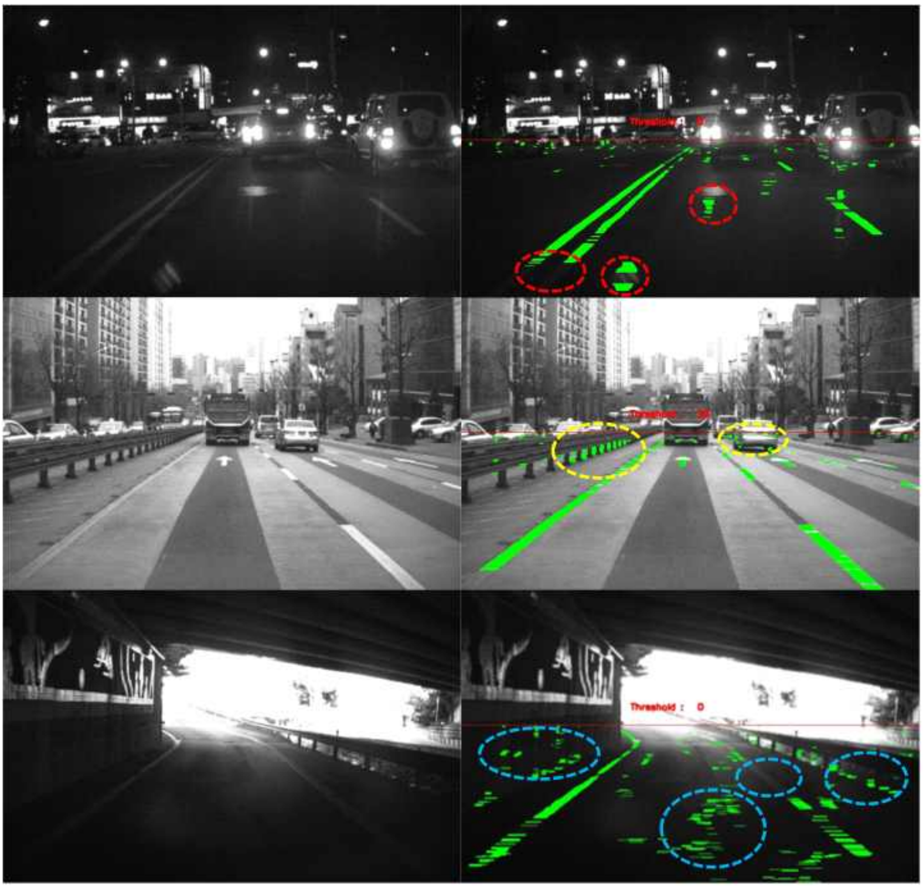


Figure 2.1. Some results of the conventional lane marker extraction approaches in challenging illumination scenarios. Dotted circles indicate extraction errors because of either the illuminative effect, or false lane marking patterns. The figures at right are the extraction result of the local threshold, positive negative gradient, and symmetry local threshold algorithm, respectively.

2.2 Lane Marking Candidate Extraction Filter

In this section, both the reason why the top hat filter is chosen, and the proposed cone hat filter is useful are explained. To do that, we first define the specification of the filter, and then show, based on the specification, a comparison of filter characteristics between the proposed and existing filters. After that, we study the top hat and cone hat filter, and compare them to each other. We also investigate the optimal parameters, such as filter width and height.

2.2.1 Requirements of the Filter

Basically, filters must satisfy the following requirements: (1) The filter has to provide either left/right points, a centre point, or all of them. (2) The lane marker extraction filter has to meet the hard real time requirement, since the lane detection system requires processing of 15-30 fps, in general. (3) Extractability decision parameters such as threshold, filter width and orientation, should be easy to calculate. In addition, the following conditions are optional. (4) For illumination tolerant lane marking extraction, the filter is required to extract lane markings under the conditions of shading, low light sunset, and road reflections on a rainy day. (5) To improve the robustness of lane marking extraction against false positives coming from small cracks, tar patches, and other deformities on the road, the filter is recommended to have a false positive cancelling function.

2.2.2 A Comparison of Filter Characteristics

We selected the most frequently used algorithms for comparison. The important characteristics of lane marker extraction filter are considered in Table 2.1. We summarize a comparison of the above mentioned filter characteristics. To satisfy the hard real time requirement, 1D filter can help to reduce the execution time. Since 1D filter has an independent relationship with the next line, the lane marking extraction algorithm may skip the extraction of next line, and jump to the line of next interval. However, we have to consider that this trick also decreases the total number of true positives. In general, 2D filter shows more accurate and robust extraction performance, but computation time increases, too. Another important thing to consider is the extractability decision parameter. Every filter depends on a threshold. Additionally, filters, except the positive negative gradient, require other parameters, such as kernel size, orientation and averaging distance. The more accurate estimation of a parameter generates better extraction performance. Fortunately, the steerable filter provides a method for optimal orientation calculation. However, the accurate estimation of kernel size, filter width, and averaging distance over various lane marking shape variations is still the open problem. Among all filters, we chose the top hat (cone hat) filter, since we found these two filters have some special strong points for extracting lane markings.

	Proposed filter	Edge detection	Pos. neg. gradient	Steerable filter	Local threshold
Left/right point	Provided	Possible	Provided	Provided	Provided
Embedded smoothing	Provided	Not provided	Not provided	Provided	Provided
Dimension	1D (Horizontal)	2D (1D possible)	1D (Horizontal)	2D Only	1D (Horizontal)
Extraction interval controllability	Possible	Possible (ifonly1D)	Possible	Limited	Possible
Computation complexity	Medium	Low	Low	High	Medium
Extractability decision parameter	Filter width	Kernel size, Threshold	Lane marking width, Threshold	Kernel size, Orientation, Threshold	Averaging distance, Threshold

Table 2.1. The characteristics comparison between the proposed filter and existing filters.

2.2.3 Cone Hat Filter

Figure 2.2 illustrates the shape of the top hat and proposed cone hat filter. The cone hat filter is also symmetrical, and two times wider than the lane marking width, to extract as well as the top hat filter. One big difference between them is just the weight; the top hat is more normalized than the cone hat i.e. the cone hat gives more weight to the centre point, than the side points. Both top hat and cone hat filter can detect a ridge feature, which has a low-high-low intensity pattern very nicely, such as the lane markings. And they extract the exact point of the left and right, if the filter width is the same as the lane marking. The important thing here is that if the filter width is slightly narrower than the lane marking, they still provide the exact point of left and right, until the filter width is greater than half of the lane marking. This characteristic greatly improves lane marking extraction capability, as well as decreasing the dependency on accurate lane marking width estimation.

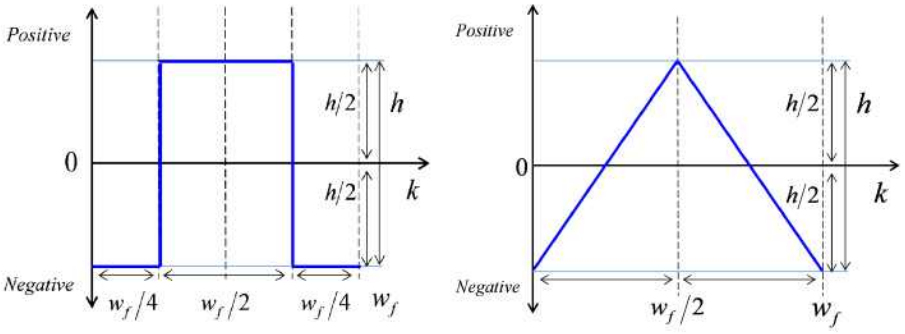
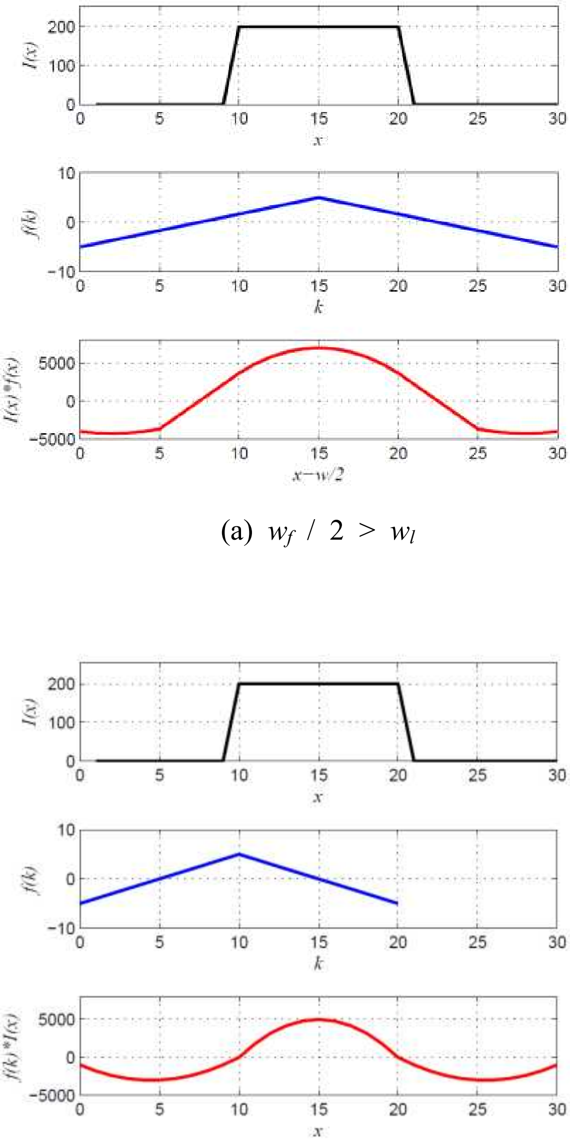


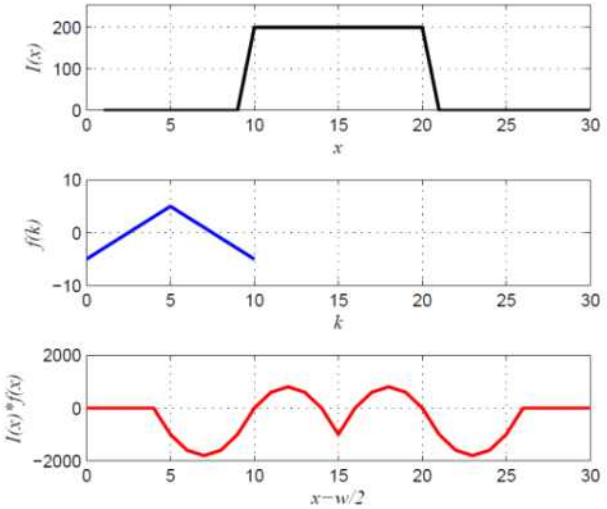
Figure 2.2. The filter is defined by the parameters of width (w_f) and height (h). The top hat filter is more normalized than the cone hat filter, while cone hat filter gives more weight to the centre point, than the side points.

Figure 2.3 illustrates filter responses according to the variation of the filter width. In terms of the optimal filter width and height, two times the lane marking width is the optimal width, but the height only affects the magnitude of the filter response, therefore we simply set it to 2.



(a) $w_f \ / \ 2 > w_l$

(b) $w_f \ / \ 2 = w_l$



(c) $w_f\ /\ 2\ <\ w_l\ <\ w_f\ /\ 2$

Figure 2.3. Filter responses according to the filter width variation. The positive area is regarded as lane marking candidates. Note that all of them successfully extract lane markings and provide a correct centre point, and two ((b) and (c)) also give correct left and right points. This characteristic makes the filter easier for the dynamic lane marking width estimation problem.

The top hat and cone hat filter are defined as follows:

$$f(k) = \begin{Bmatrix} -h, & \text{if } k < w_f / 4, k > 3w_f / 4 \\ h, & \text{otherwise} \end{Bmatrix}$$

$$\text{where} \qquad\qquad (1)$$

$$0 \le k < w_f,$$

$$4 \le w_f, \qquad (w_f \text{ is an even number})$$

$$f(k) = \begin{cases} -h/2 + m(k)k, & \text{if } k < w_f/2 \\ h/2 + (k - (w_f - 1)/2)m(k), & \text{otherwise} \end{cases}$$

where

$$0 \le k < w_f, \tag{2}$$

$$3 \le w_f, \quad (w_f \text{ is an odd number})$$

$$m(k) = \begin{cases} 2h/(w_f - 1), & \text{if } 0 \le k \le w_f/2 \\ -2h/(w_f - 1), & \text{otherwise} \end{cases}$$

.

2.3 Overview of the Proposed Algorithm

In this section, we outline the top hat (cone hat) filtering procedures, and the proposed *reiterated extraction* algorithm. *Reiterated extraction* uses two different top hat (cone hat) filters, having different filter widths. The algorithm is composed of lane marking extraction and false positive cancelling process.

2.3.1 Filter Width Estimation

For the expectation of better extraction performance, a dynamic estimation of filter width is desirable. However, we adopted a naive linear filter width estimation method, since the top hat (cone hat) can still extract lane marking candidates very well, although lane marking edges are slightly inaccurate. Experimental results for the curve scenario support the reliability of the naive estimation method. For the estimation, we assumed that both the maximum and minimum lane marking widths are static and already measured empirically. The maximum/minimum filter widths are exactly two times those of the lane marking widths. In addition, we set the minimum lane marking width to two times greater than the actually measured marking width, since that provides more robust extraction results. Filter width estimation is defined as follows:

$$w_f = \left\lfloor w_f^{\max} - (w_f^{\max} - w_f^{\min})(\frac{y^{\max} - y}{y^{\max} \,.\, y^{\min}}) \right\rfloor$$
$$w_f = \begin{cases} w_f - 1, & w_f \bmod 2 = 1 \\ w_f, & \text{otherwise} \end{cases} . \tag{3}$$

where $w_f^{\max}$, $w_f^{\min}$, $y^{\max}$ and $y^{\min}$ denote the given max./min. filter widths, and lower and upper limit of ROI, respectively. The filter width should be an odd number for the symmetricity of the filter.

2.3.2 Top Hat (Cone Hat) Filtering

The filtering process is the combination of a filter generation, a convolution of the filter with a series of intensities, and a shifting for an alignment. The filter is generated according to the vertical coordinate of each line input, by using either (1) or (2), and (3). Convolution and alignment are defined by:

$$r(x - w_f / 2) = f(k) * I(x), \;\; 0 \le k \le w_f \tag{4}$$

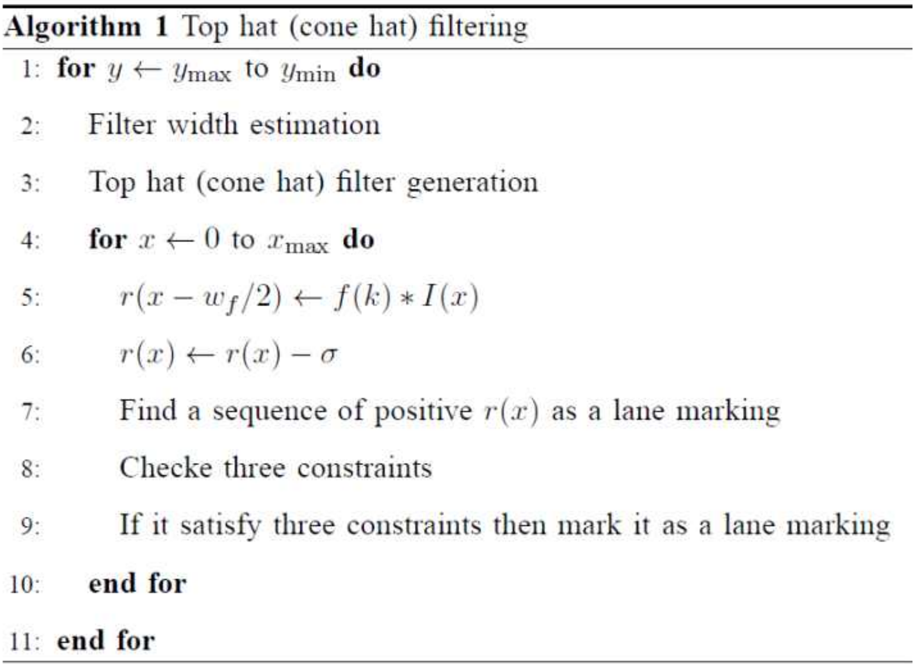
where, w_f /2, $f(k)$, $I(x)$ and $r(x)$ denote the magnitude of shifting, the filter, intensities, and the filter responses, respectively. To recover the empty right-end values because of shifting, we extended the size of the line input, and reused the right-end pixel intensity. To extract lane marking candidates, we regard positive filter responses are lane markings. Here, we used the threshold (σ) as below:

$$M(x) = \begin{cases} 1, & \text{if } r(x) - \sigma > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where, 1 denotes lane markings, while 0 means the background. Usually, applying the threshold σ is very useful to eliminate false positives (FPs), since some FPs having very weak but positive filter responses appear. However, FPs are not clearly erased by using the threshold alone. Therefore, we adopted three constraints for FP cancelling as follows:

$$\begin{aligned}&0.5w_l < l < 1.2w_l,\ l = |p_R - p_L| \\ &0.2 < r_L^{\min} / r_R^{\min} < 5 \\ &\overline{I}_L + |\overline{I} - \overline{I}_C| / 2 < \overline{I}_C < \overline{I}_R + |\overline{I} - \overline{I}_C| / 2\end{aligned} \tag{6}$$

where l, $p_{R(L)}$ denote the length of lane marking, and right and left-end point of lane marking, respectively. $r_L{}^{\min}$ and $r_R{}^{\min}$ mean the minimum filter response of the left and right side of lane marking. And finally, $\overline{I}_L, \overline{I}_R, \overline{I}_C$ and $\overline{I}$ denote the average intensity of left, right and centre of lane marking, and the average of average intensities, respectively. The first constraint prevents the extraction of lane markings having either too wide or too narrow a length as compared with the estimated lane marking width. The second and third constrain filter FPs generated from an abnormal intensity pattern, such as either low-high-high or high-low-low. Since these kinds of patterns also make positive responses, these two constraints check the symmetricity of the input pattern in the filter response and the intensity level, respectively. The algorithm of the top hat (cone hat) filter is shown as follows:



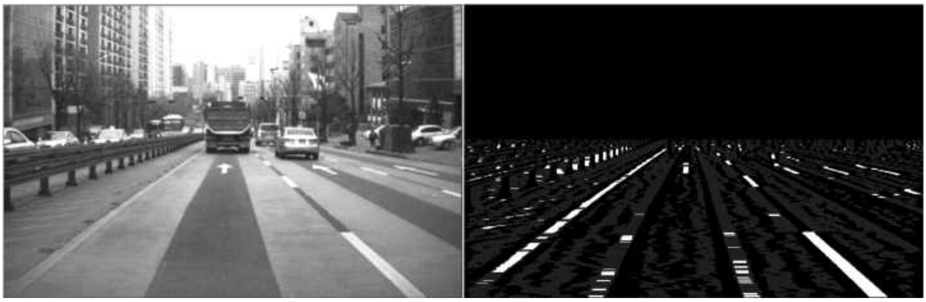
Algorithm 1 Top hat (cone hat) filtering

```
for $y \leftarrow y_{\max}$ to $y_{\min}$ do
    Filter width estimation
    Top hat (cone hat) filter generation
    for $x \leftarrow 0$ to $x_{\max}$ do
        $r(x - w_f/2) \leftarrow f(k) * I(x)$
        $r(x) \leftarrow r(x) - \sigma$
        Find a sequence of positive $r(x)$ as a lane marking
        Checke three constraints
        If it satisfy three constraints then mark it as a lane marking
    end for
end for
```

Figure 2.4. Algorithm of the Top hat (Cone hat) filtering.

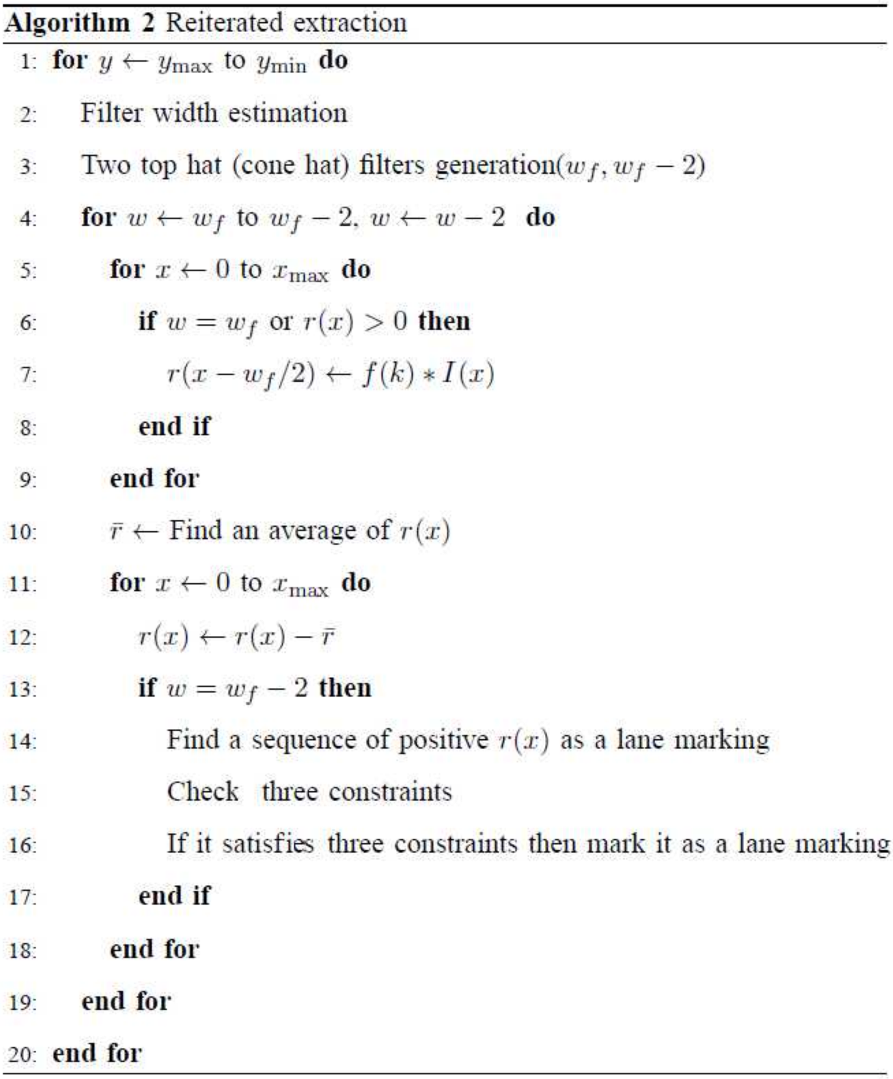
2.3.3 Reiterated Extraction

Reiterated extraction is the extended version of top hat (cone hat) filtering. Reiterated extraction uses two of the same top hat (cone hat) filters, but having different filter widths. The broader filter firstly searches lane marking candidates without any constraints, even though they contain FPs, and then the narrower filter searches lane marking candidates once again with three constraints, only within the positive range of filtering result of the broad filter. Recall that the narrower filter always gives either equal or narrower lane markings than the broader filter. To reduce FPs, the broad filter narrows the search ranges, and makes it easier to determine the threshold adaptively. Here, instead of adopting the user defined threshold to filter out FPs, we utilized the average of the filter responses of the narrow filter as an adaptive threshold. Empirically, we found that adopting the average as a threshold is useful, since this does not erase weak lane markings, though some FPs still remain. The example result and the algorithm of reiterated extraction are shown in Figure 2.5 and Figure 2.6, respectively.



(a) Original image (b) Reiterated extraction result

Figure 2.5. Gray and white pixels are the result of broader and narrower filter, respectively. The narrower filter extracts only within the broader filter's extraction results. Double extraction adopted the average of the narrow filter's results in each line as an adaptive threshold.



```
Algorithm 2 Reiterated extraction
for $y \leftarrow y_{\max}$ to $y_{\min}$ do
    Filter width estimation
    Two top hat (cone hat) filters generation($w_f, w_f - 2$)
    for $w \leftarrow w_f$ to $w_f - 2$, $w \leftarrow w - 2$ do
        for $x \leftarrow 0$ to $x_{\max}$ do
            if $w = w_f$ or $r(x) > 0$ then
                $r(x - w_f/2) \leftarrow f(k) * I(x)$
            end if
        end for
        $\bar{r} \leftarrow$ Find an average of $r(x)$
        for $x \leftarrow 0$ to $x_{\max}$ do
            $r(x) \leftarrow r(x) - \bar{r}$
            if $w = w_f - 2$ then
                Find a sequence of positive $r(x)$ as a lane marking
                Check three constraints
                If it satisfies three constraints then mark it as a lane marking
            end if
        end for
    end for
end for
```

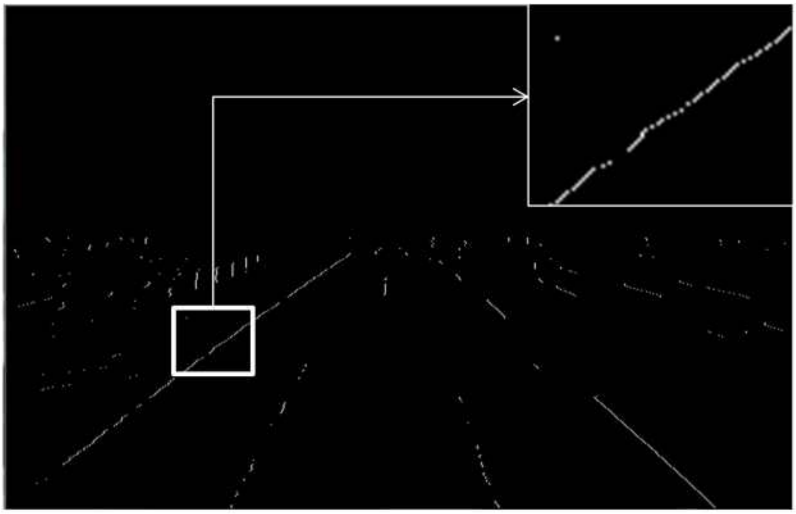
Figure 2.6. Algorithm of the Reiterated extraction.

2.3.4 False Positive Cancelling

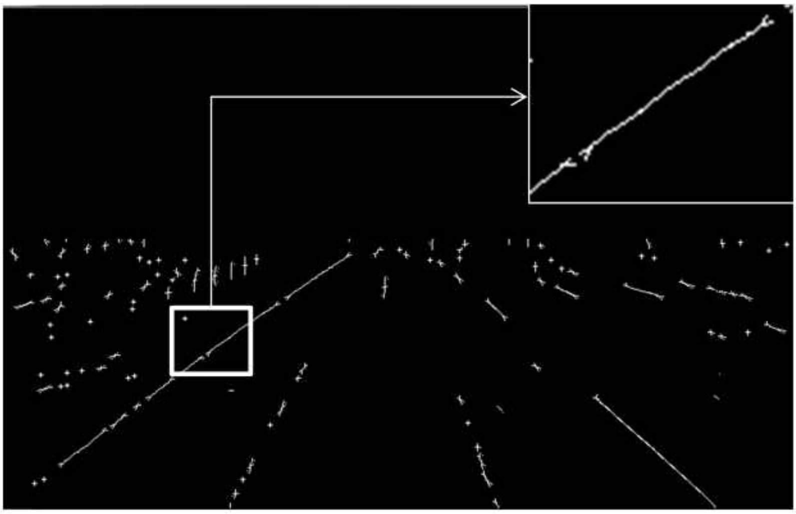
False positive cancelling is the post-processing of double extraction. The proposed false positive cancelling method is very effective, and it impressively increases the scores in both the receiver operating characteristic (ROC), and dice similarity coefficient (DSC) measures. Since the proposed double extraction algorithm is designed for illumination tolerant lane marking extraction, the strategy for the extraction was set up as follows: Lane marking candidates are extracted as much as possible during the filtering process, and then false positives are eliminated, by using illumination invariant constraints. The key idea is checking the direction of each segment, and comparing it with the direction to the vanishing point, since most of the lane markings face the vanishing point. For doing that, we divide the false positive cancelling process into five sub-processes: (1) lane marking centre point extraction, (2) fast centre point clustering, (3) vanishing point detection, (4) segment extraction, and (5) false positive filtering.

2.3.4.1 Lane Marking Center Point Extraction

Lane marking centre points are extracted with the result of reiterated extraction (Figure 2.5 (b)). Centre points are used for finding the vanishing point, and false positive cancelling. The result is shown in Figure 2.7 (a).



(a) The set of lane marker centre points

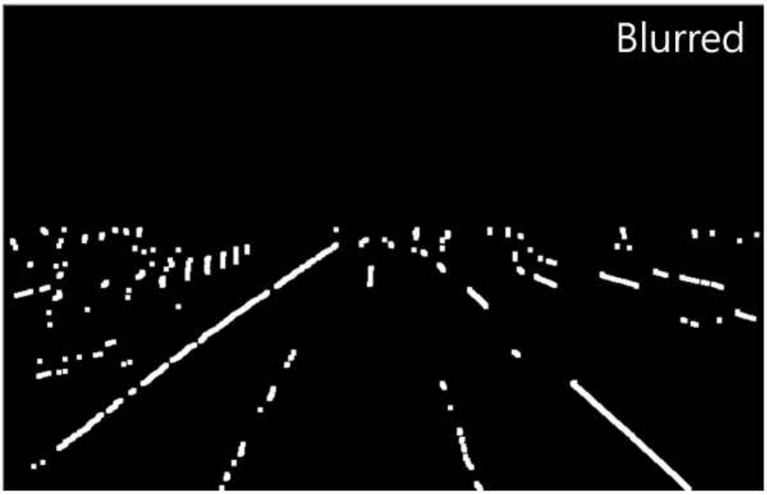


(b) Final centre point clustering result

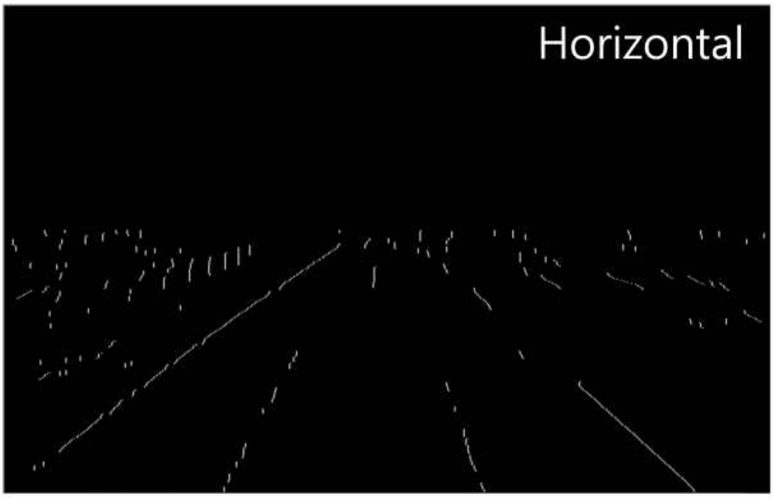
Figure 2.7. Lane marking centre points extraction and clustering result. Clustering process allows to extract the centre points segments easier in the segment extraction step.

2.3.4.2 Fast Center Point Segmentation

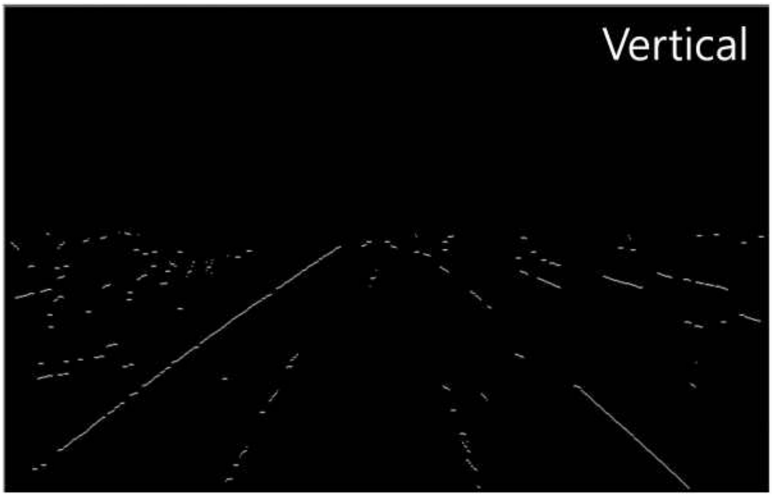
Clustering means here, connecting nearby discrete centre points of a lane marking to be a line segment. It is an essential process, since the segment having a small number of points can be deleted during the FP filtering process, even though the segment is a true positive. In particular, the lane marking centre points on the side of an image are not actually well connected. To make the process faster, we devised a simple but effective method. Firstly, we blur the image of the lane marking centre points, and then extract the centre points again from the blurred image in both the horizontal and vertical direction, and finally, combine them together. The result of centre point clustering is shown in Figure 2.7 (b), and each step is shown in Figure 2.8.



(a) Blurred image



(b) Horizontal extraction result

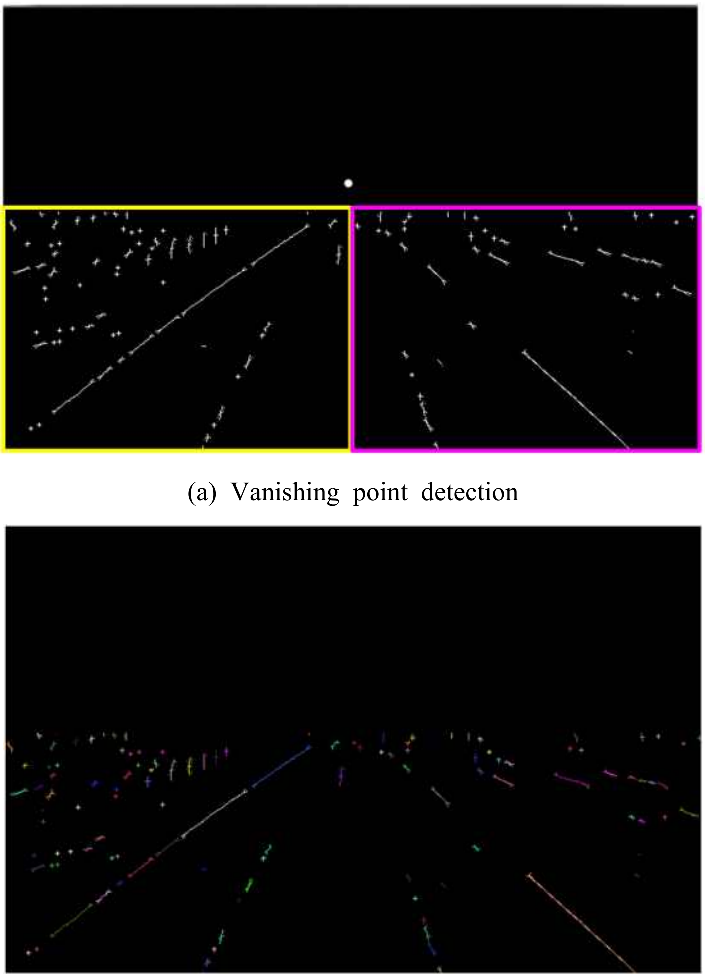


(c) Vertical extraction result

Figure 2.8. Fast centre points clustering sub-process: Horizontal (b) and vertical (c) extraction results are generated from the centre point detection result of the blurred image (a) in the horizontal and vertical direction, respectively. Final centre point clustering result is the addition of two images.

2.3.4.3 Vanishing Point Detection

For calculation of the vanishing point, we separate the ROI to two sub sections, as shown in Figure 2.9 (a). In each sub-section, we conduct a Hough line transform, and find the dominant line.

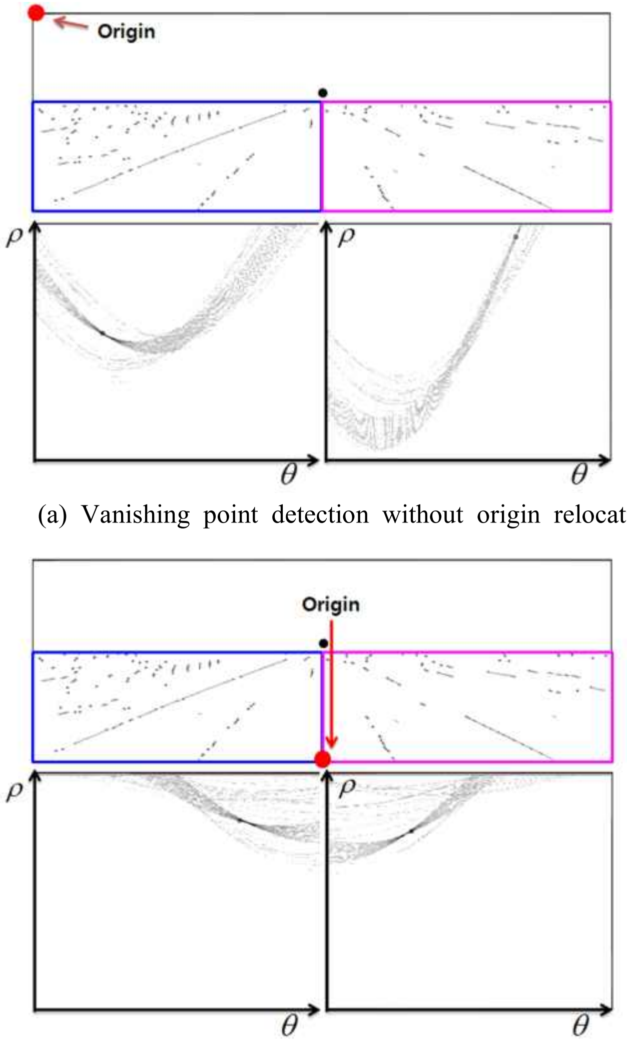


(a) Vanishing point detection

(b) Segment extraction result

Figure 2.9. Vanishing point detection and segment extraction result. Yellow and magenta rectangular box in figure 7a indicate two regions for extracting one or two dominant lines, and the crossing point of them is the vanishing point. The segments of centre points shown in figure 7b will be checked the direction of the line segment and filter out if it does not face the vanishing point.

Additionally, we relocated the origin point from (0,0) to ($x^{\max}/2$, $y^{\max}$), to find the more accurate vanishing point without the scale adjustment of angle θ. This is illustrated in Figure 2.10.



(a) Vanishing point detection without origin relocation

(b) Vanishing point detection with the origin relocation

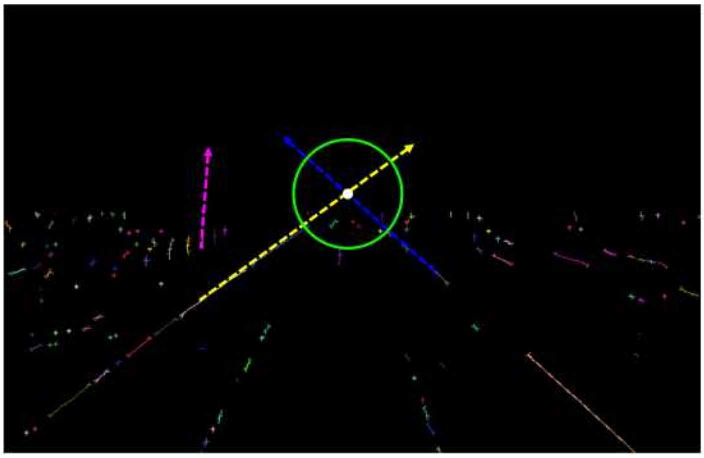
Figure 2.10. Dominant line searching in two given search region by using the Hough transform is shown. Left (a) and right (b) show the voting results without and with the origin relocation, respectively. The origin relocation gives more accurate voting result without scale adjustment of angle θ.

2.3.4.4 Segment Extraction

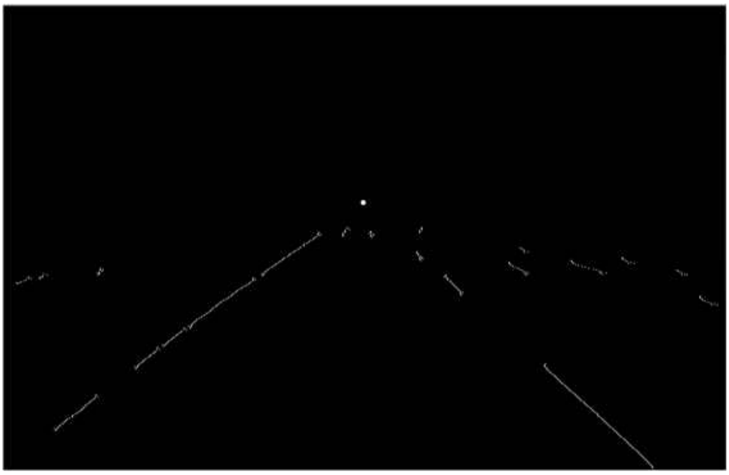
To extract line segments from the clustered image, we adopted the contour extraction algorithm based on [19-20]. Thanks to the contour extraction algorithm, each line segment can be retrieved one by one, and depending on the following false positive filtering result the false positive segment will be eliminated. The segment extraction result is shown in Figure 2.9 (b). Different color indicates different segment.

2.3.4.5 False Positive Filtering

The dominant angle of each segment is calculated based on Hough voting, and then the decision of false positive is made by checking the shortest distance between the line of a segment, and the vanishing point. We also eliminate the relatively too small segment, as compared to the estimated filter width. The false positive filtering algorithm is shown in Figure 2.13. The function *MAX*(*a, b*) means finding the bigger value between *a* and *b*, and we set the threshold ε_1, ε_2 to 80, 50, respectively. The conceptual illustration of false positive filtering is shown in Figure 2.11.



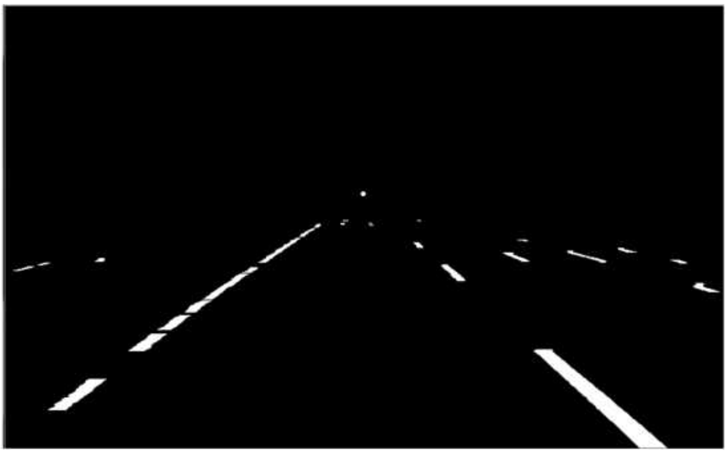
(a) Illustration of the false positive filtering



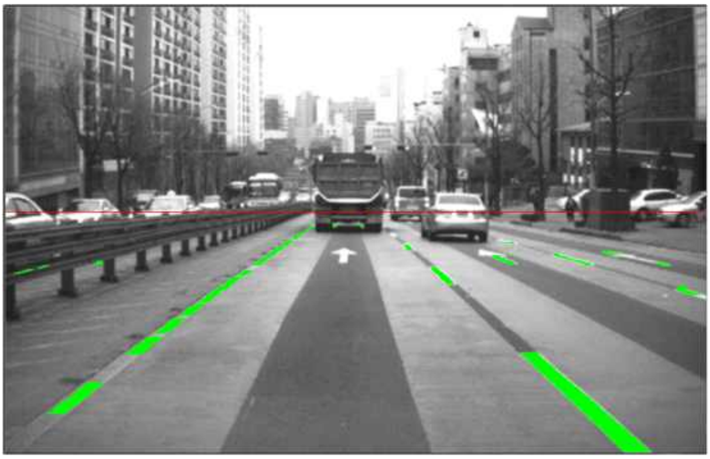
(b) False positive filtering result

Figure 2.11. The illustration of false positive filtering process (a) and the false positive filtering result (b).

After the false positive filtering process, we can recover lane markings, by using the set of centre points and the result of reiterated extraction. The final result of the reiterated extraction and FP cancelling is shown in Figure 2.12.



(a) Lane marking recovering using centre points



(b) Final result of the false positive cancelling process

Figure 2.12. Final result of the lane marking extraction is shown in (a) and (b).

Algorithm 3 FP filtering

```
$s^{\max}$ ← total number of segments
for $i \leftarrow 0$ to $s^{\max}$ do
    Read a segment
    $c_i$ ← total number of points in the segment
    $w_f$ ← estimated filter width
    if $c_i \leq MAX(w_f, 3)$ then
        Delete the segment
    else
        $a_i$ ← dominant angle of the segment
        $d_i$ ← shortest distance between the line of the segment and the vanishing point
        if $d_i > \epsilon_1$ then
            Delete the segment
        else
            if $-20 \leq a_i \leq 20$ then
                if $d_i > \epsilon_2$ then
                    Delete the segment
                end if
            end if
        end if
    end if
end for
```

Figure 2.13. False positive filtering algorithm.

2.4 Experiments and Evaluation

In this section, we present our experimental results in comparison with other conventional algorithms in ten scenarios.

2.4.1 Experimental Set-up

To investigate the performance over the illumination changes, we experimented with eight scenarios, whose lightning conditions are very different to each other. Additionally, we also tested curve and up-downhill scenarios, for testing geometrical changes. The images contain not only road and lane markings, but also cars and barriers. These objects on the road are the main reason for false positive generation. The list of experimental scenarios is shown in Table II. Each scenario has a hundred gray images, and there are *rgb* and ground truth images in the database as much as gray images. To make ground truth images, we used a graphics software to label the lane markings manually. The specification of the input image is shown in Table 2.2.

Perspective	Scenario	Location	Quantity
Illumination changes	Not shaded road	Highway, Local	100
	Shaded road		100
	Sunset	Highway	100
	Night time	Urban	100
	Urban road (daytime)		100
	Urban road (rainy day)		100
	Tunnel (daytime)	Local	100
	Tunnel (rainy day)		100
Geometrical changes	Curve	Urban, Local	100
	Up-downhill		100
Total			1,000

Table 2.2. Experimental scenarios

List	Condition
Resolution	752(H) x 480(V)
ROI	752(H) x 260(V)
Color space	Gray(1CH)
Normalization	$I(u,v) = I(u,v) / I^{\max} \times 255$

Table 2.3. Experimental conditions

To reduce experimental bias, images are normalized by the maximum intensity of the image. For evaluation, we adopted two classic measures [12]: the Receiver Operating Characteristic (ROC), and Dice Similarity Coefficient (DSC). These are defined as follows:

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \tag{7}$$

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{8}$$

ROC measures the ratio between the true positive rate (TPR) and false positive rate (FPR). A higher TPR with lower FPR is better. The quality of the DSC curve is measured by the maximum value and width of the peak; the higher and larger, respectively, have better performance. The threshold at the peak point is optimal.

2.4.2 Conventional Algorithm for Evaluation

For evaluation, we implemented an additional seven algorithms as shown in Table 2.4. For all algorithms except the global threshold, we apply the same lane marking width estimation, equation (3), and lane marking width constraint as follows:

$$0.5 \times w_f / 2 \le w_l \le 1.2 \times w_f / 2 \tag{9}$$

where, w_l and w_f denote the lane marking width and filter width, respectively.

Algorithm	**Abbreviation**
Global threshold	GT
Positive negative gradient	PNG
Local threshold	LT
Symmetry local threshold	SLT
Symmetry local threshold + Double extraction	MLT-SLT
Top hat	TH
Cone hat	CH
Gaussian	GS
Top hat + Reiterated extraction + FP cancelling	RITR-TH
Cone hat + Reiterated extraction + FP cancelling	RITR-CH
Gaussian + Reiterated extraction + FP cancelling	RITR-GS

Table 2.4. Implemented algorithms for performance evaluation

2.4.2.1 Global threshold

Global threshold simply finds pixels having a higher intensity than the threshold in a gray image. It does not find the start and end point of lane markings. As you expect, it is used for a reference, to quantify how much better the other algorithms work.

2.4.2.2 Positive Negative Gradient

Positive negative gradient finds lane markings having a low-high-low intensity pattern in the horizontal image line, by using the consecutive positive and negative gradient; the former is greater than the positive threshold, and the latter is lower than the negative threshold.

Additionally, the length between the two should satisfy the expected lane marking width constraint, as shown in (9). Smoothing is conducted before extracting lane markings, since a noise pixel having high intensity can be detected as a lane marking start point, since it satisfies the positive gradient condition. To achieve better performance, we applied a five pixel distance gradient, which calculates the gradient of not the next pixel but the five pixel distanced pixel. It is tricky, but the strong point is that it is very efficient for lane markings having very low intensities, such as one in shadows, since the gradient values become greater than the original gradient values.

2.4.2.3 Local Threshold

Local threshold [18] uses the average image intensity of local range, which varies, based on the estimated lane marking width. Local threshold is determined by the sum of the horizontal local average intensity and the threshold. By using the local threshold, the acceptance test is defined as below:

$$I(u,v) > \overline{I}(u,v) + \sigma \tag{10}$$

where, $I(u,v), \overline{I}(u,v)$ and σ denote the intensity of the current pixel position, the local average intensity and the threshold, respectively. All extracted pixels are saved in the extraction map and the horizontally connected features satisfying the above mentioned lane marking width constraint are selected as the lane markings. We set the local range to $2w_f$.

2.4.2.4 Symmetry Local Threshold

Symmetry local threshold [18] adopted the average of left and right local ranges. For the acceptance test, lane marking pixels should satisfy two conditions, as below:

$$I(u,v) > \overline{I}_L(u,v) + \sigma, \;\; I(u,v) > \overline{I}_R(u,v) + \sigma \tag{11}$$

where, $\overline{I}_L(u,v)$ and $\overline{I}_R(u,v)$ denote the left and right local threshold, respectively. The lane marking extraction rule is the same as the local threshold. We set the left/right local range to $2wf$.

2.4.2.5 Double Extraction using Symmetry Local Threshold

Double extraction [12] combines the result of two extractions of the same algorithm (here, we used the symmetry local threshold), but those thresholds are different: one is higher, the other is lower. The result from the higher threshold is transformed by the morphological dilate operation, which expands the extraction regions. And finally, the pixels satisfying two conditions at the same time are selected as lane markings. The conditions are defined as follows:

$$I_H(u,v) > \sigma,\ I_L(u,v) > \sigma \qquad (12)$$

where, $I_H(u,v)$ and $I_L(u,v)$ denote the result of the symmetry local threshold, using the high threshold and low threshold, respectively. We set the local range and threshold difference to $2w_f$ and 5, respectively.

2.4.2.6 Gaussian filter

Gaussian filter is proposed for the evaluation of the detectability as compared to the proposed cone hat and top hat filter. The shape of the gaussian filter is illustrated as shown in Figure 2.14. Likewise the cone hat filter, it is also symmetrical, and two times wider than the lane marking width. Main difference from cone hat filter is the weight. Gaussian filter is more normalized than the cone hat filter but less normalized than top hat filter.

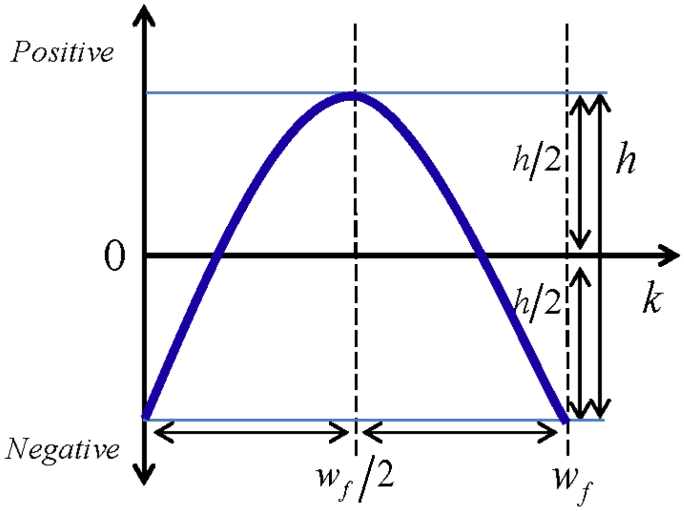


Figure 2.14. Illustration of the Gaussian filter. Gaussian filter is more normalized then the cone hat filter but less normalized than top hat filter.

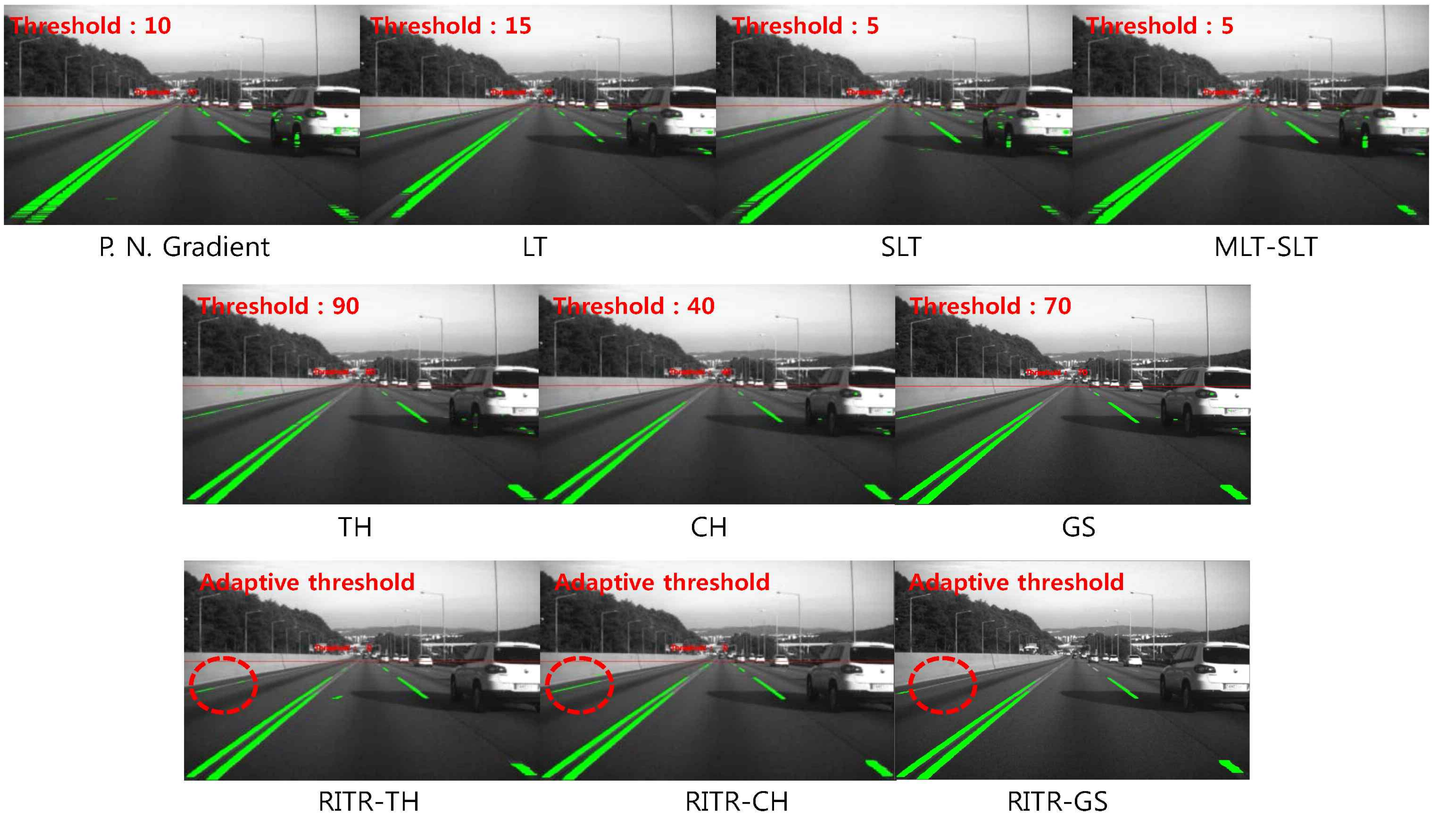
Gaussian filter is defined as follows:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} + d \tag{13}$$

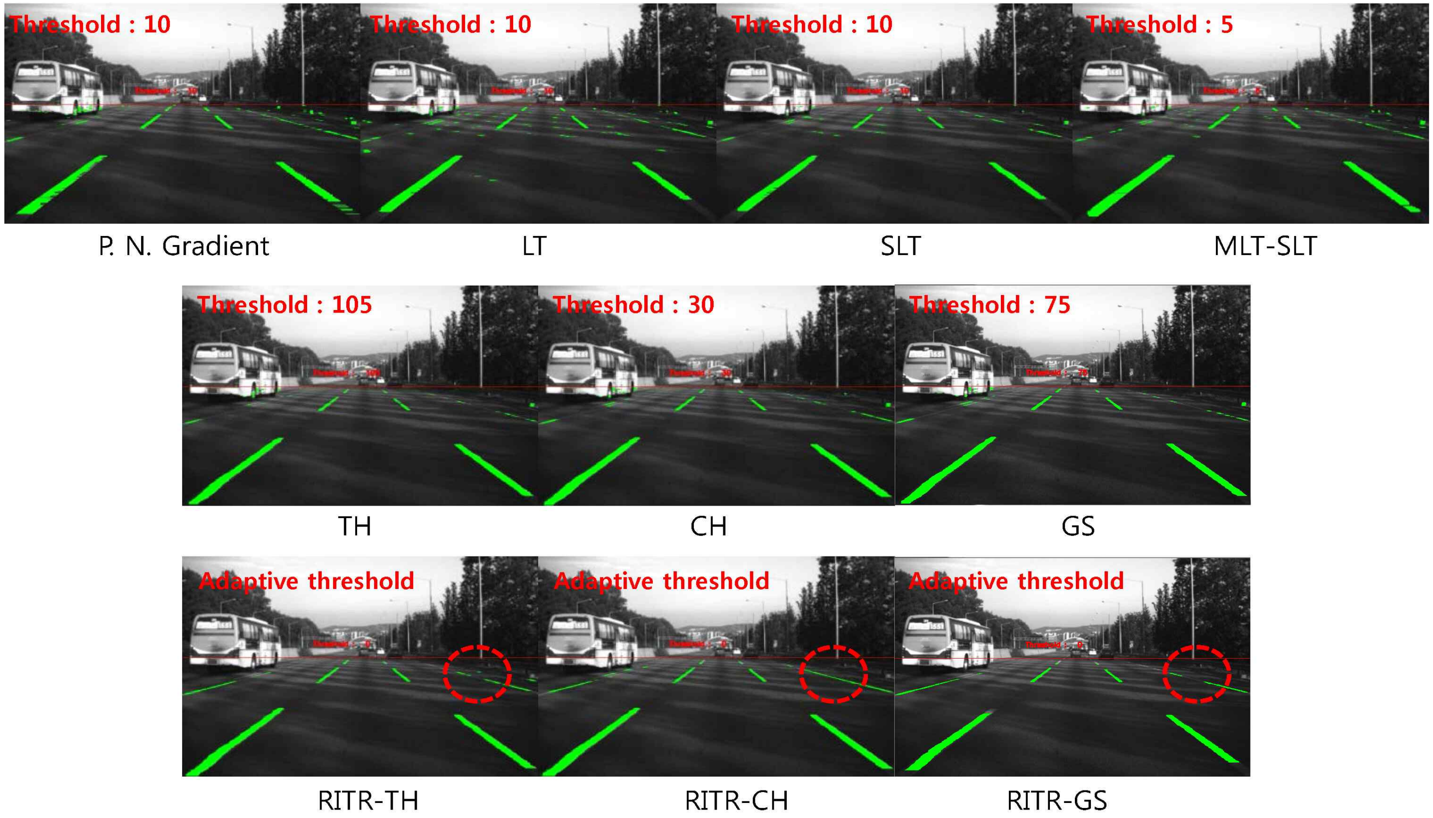
where, a is the height of the curve's peak, b is the position of the center of the peak, and c controls the width of the filter. We set 2, $w_f/2$, $w_f/5$, and -1 for a, b, c and d, respectively.

2.4.3 Experimental Results

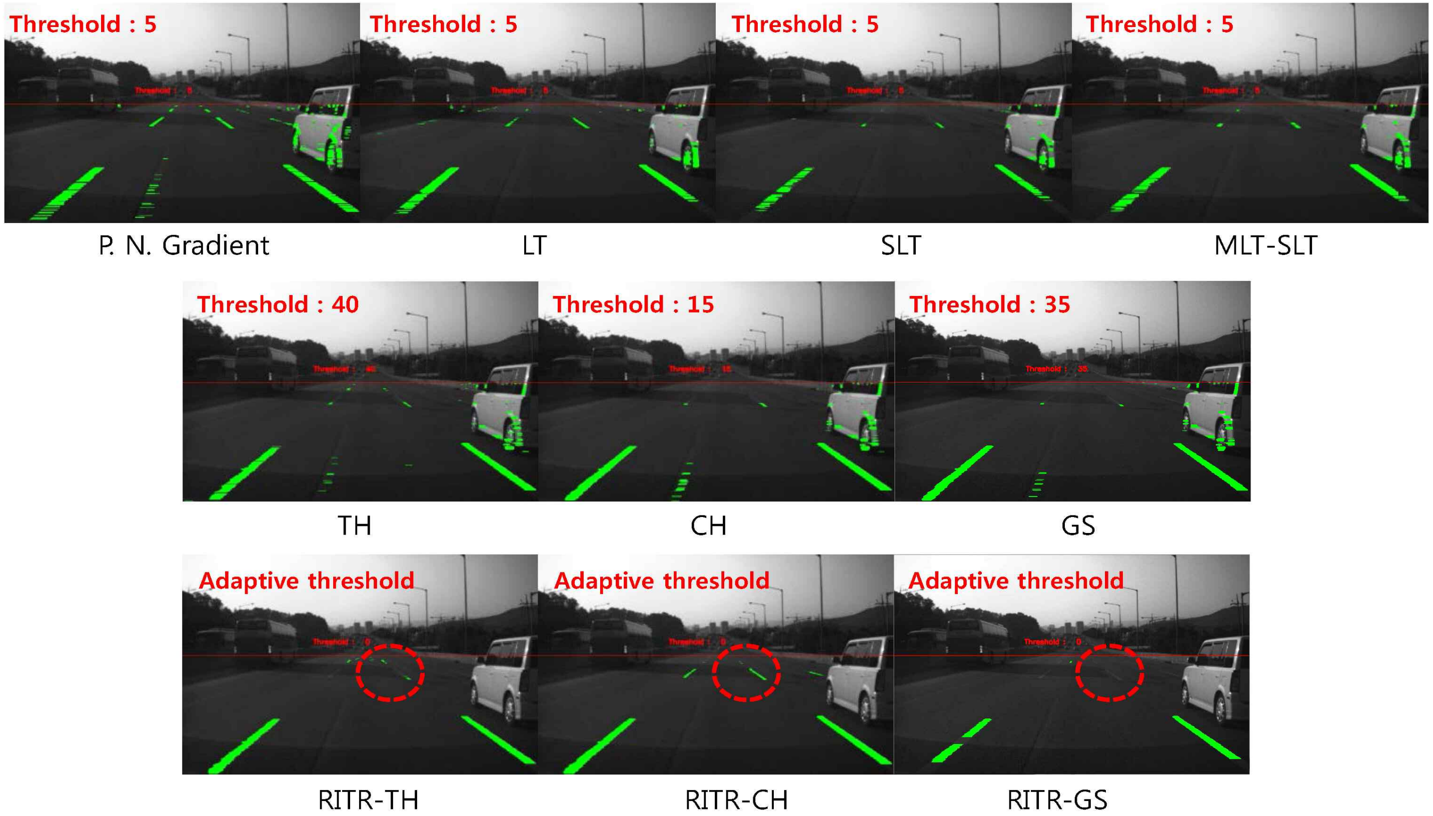
To produce a richer and more comprehensive understanding of the evaluation of experiments, we present both qualitative (Figure 2.15) and quantitative (Figure 2.17-19) data. Figure 2.15 represents the lane marking extraction results for each algorithm, over varying illumination environments.



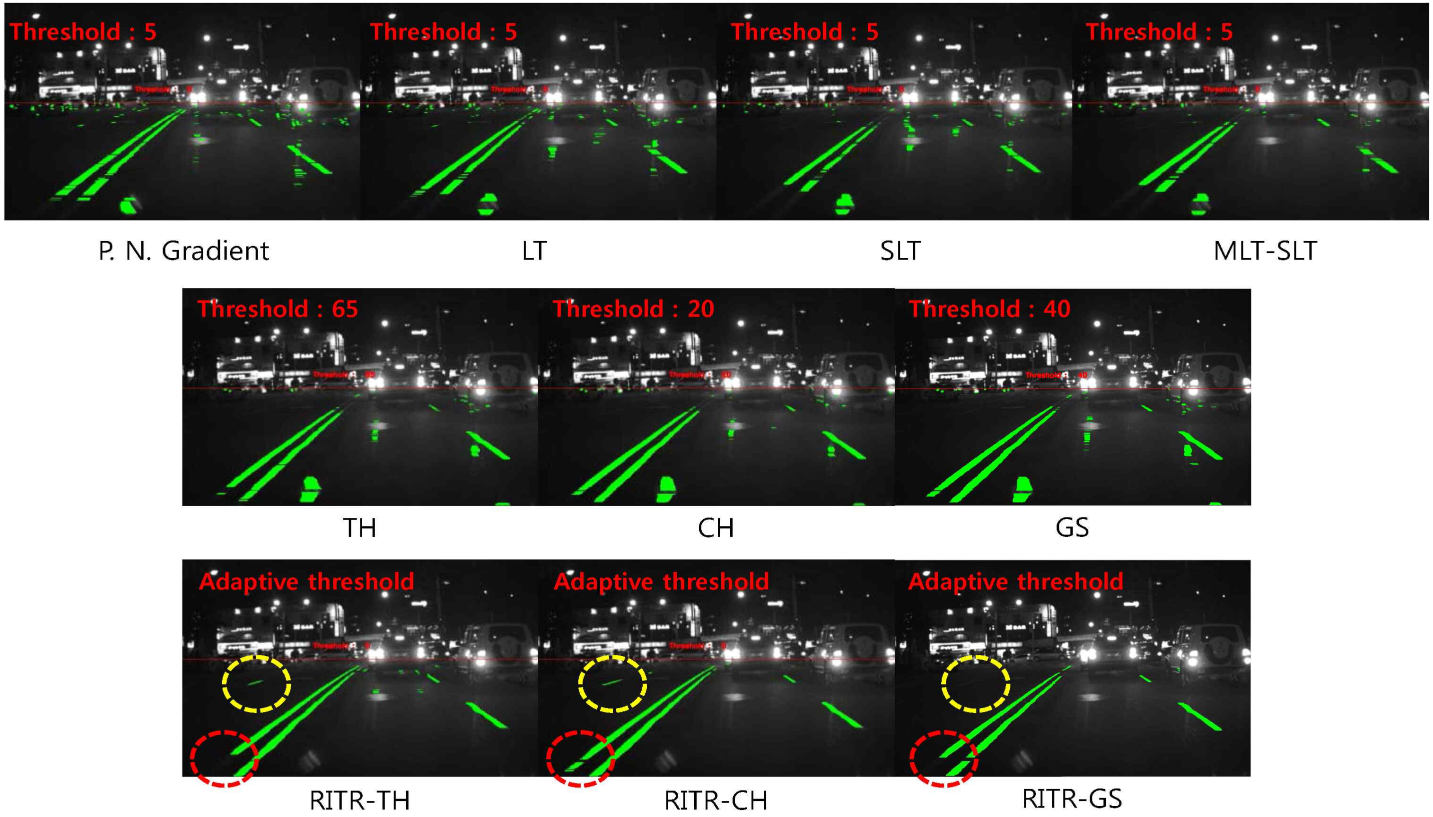
(a) An example of lane marking extraction results in the “Not shaded road” scenario.



(b) An example of lane marking extraction results in the "Shaded road" scenario.



(c) An example of lane marking extraction results in the “Sunset” scenario.

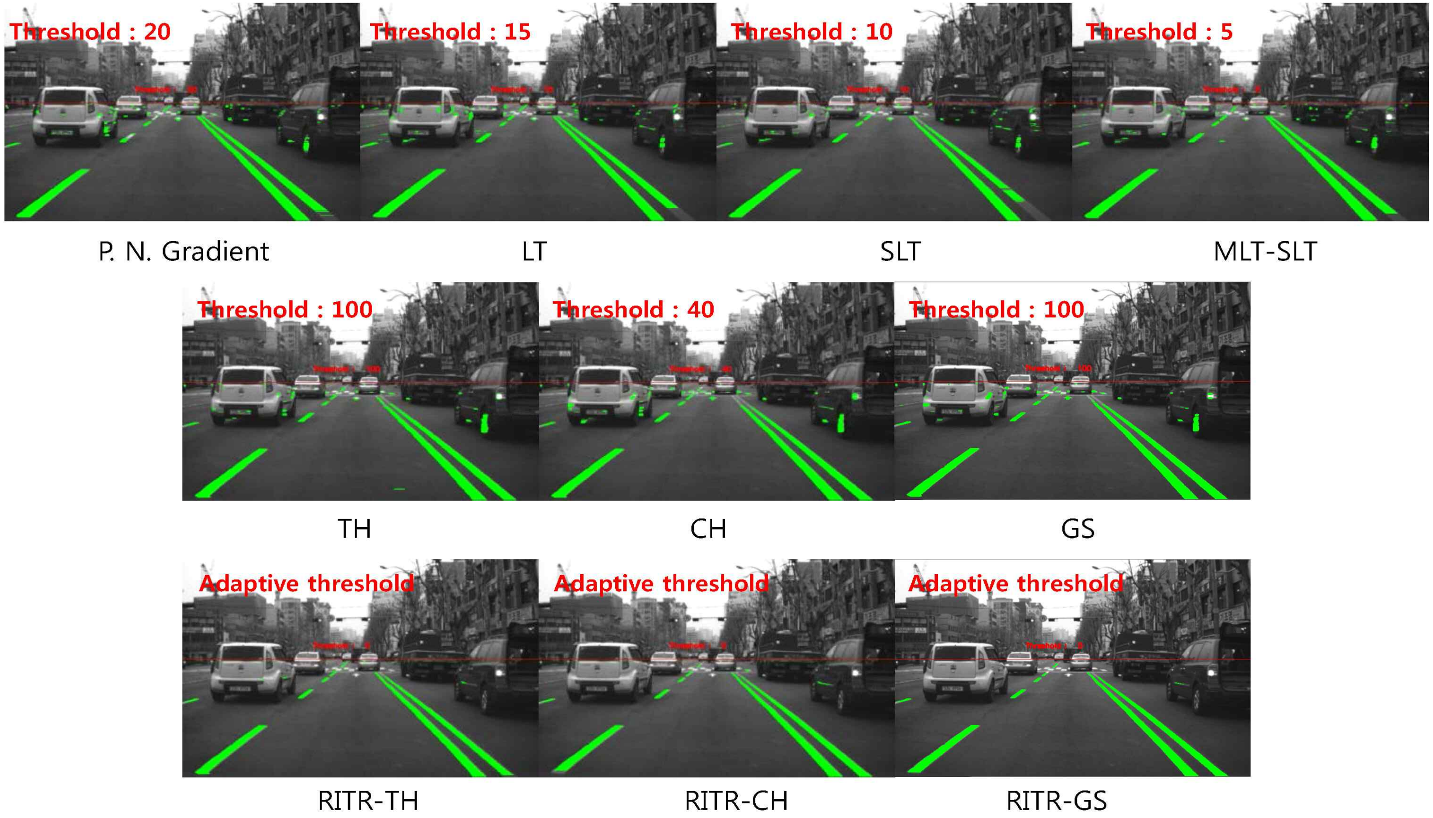


P. N. Gradient LT SLT MLT-SLT

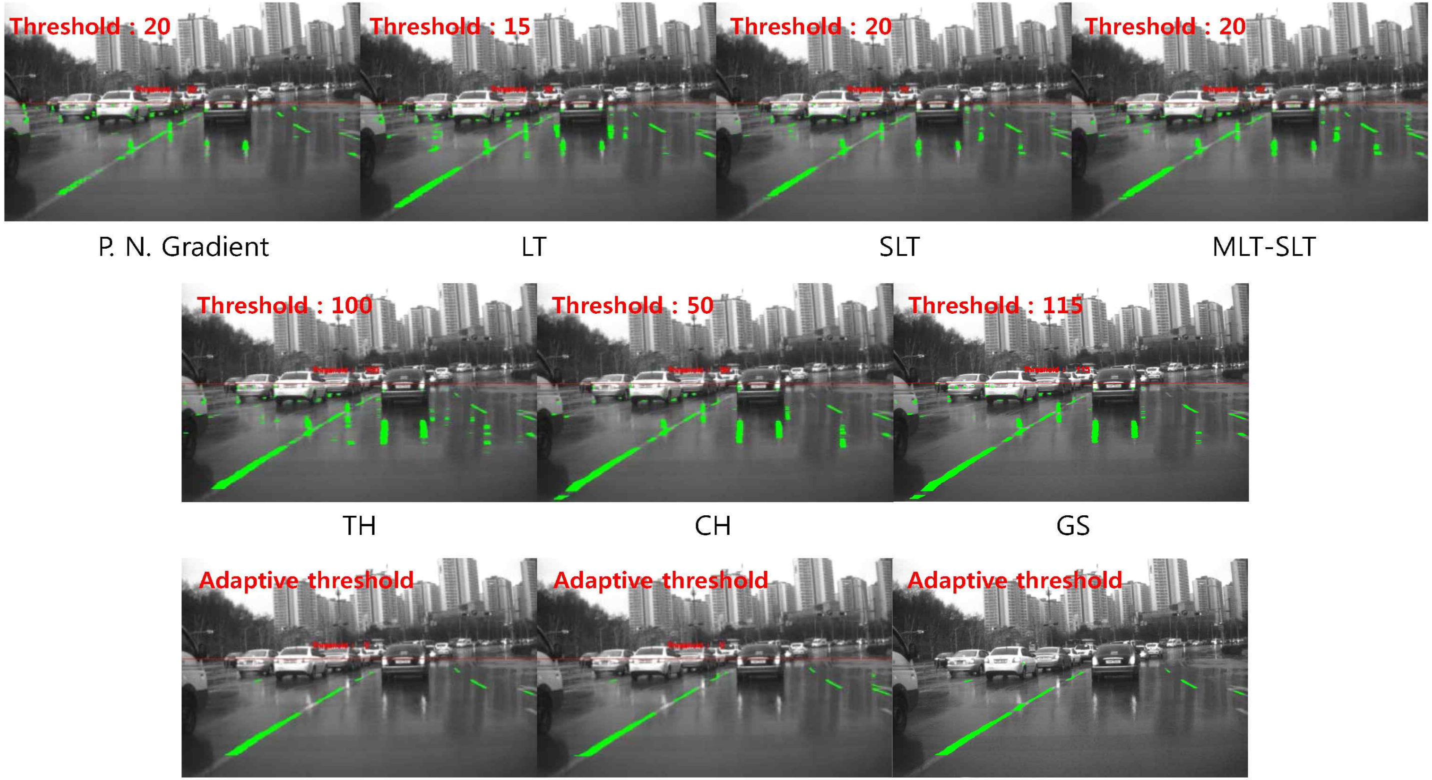
TH CH GS

RITR-TH RITR-CH RITR-GS

(d) An example of lane marking extraction results in the “Night time” scenario.



(e) An example of lane marking extraction results in the “Urban road (daytime)” scenario.

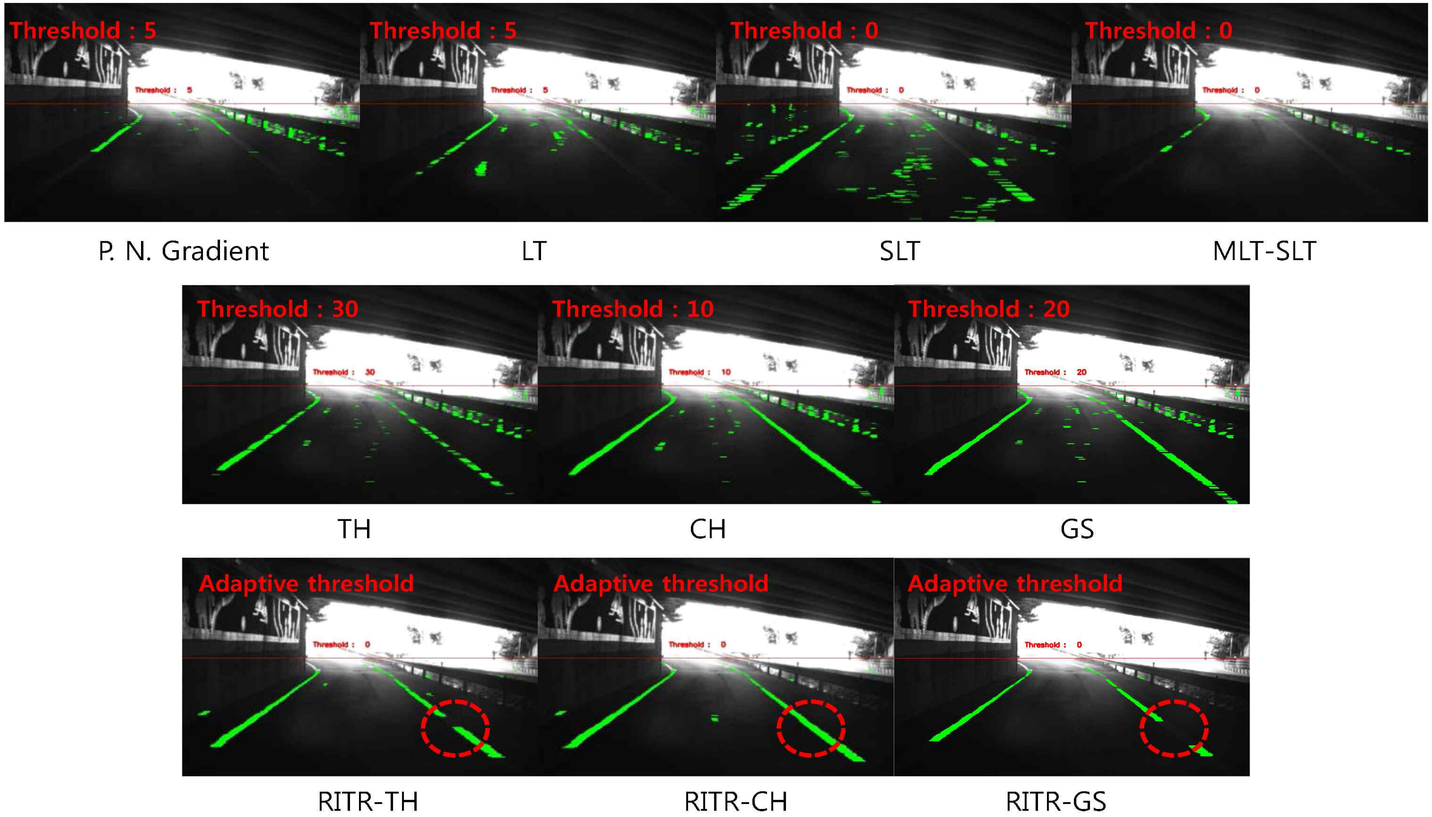


P. N. Gradient LT SLT MLT-SLT

TH CH GS

RITR-TH RITR-CH RITR-GS

(f) An example of lane marking extraction results in the “Urban road (rainy day)” scenario.

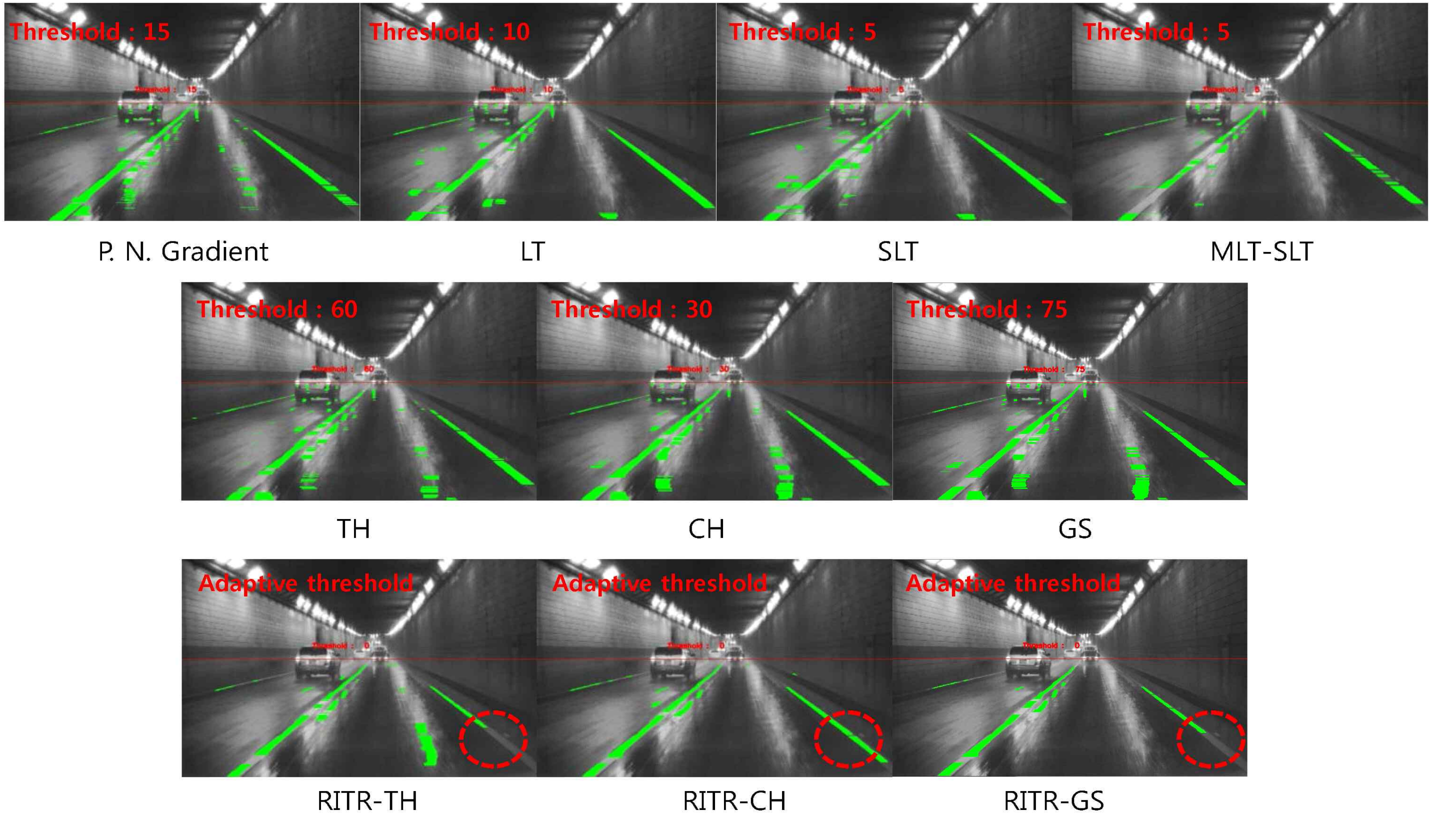


P. N. Gradient LT SLT MLT-SLT

TH CH GS

RITR-TH RITR-CH RITR-GS

(g) An example of lane marking extraction results in the “Tunnel (daytime)” scenario.



P. N. Gradient

LT

SLT

MLT-SLT

TH

CH

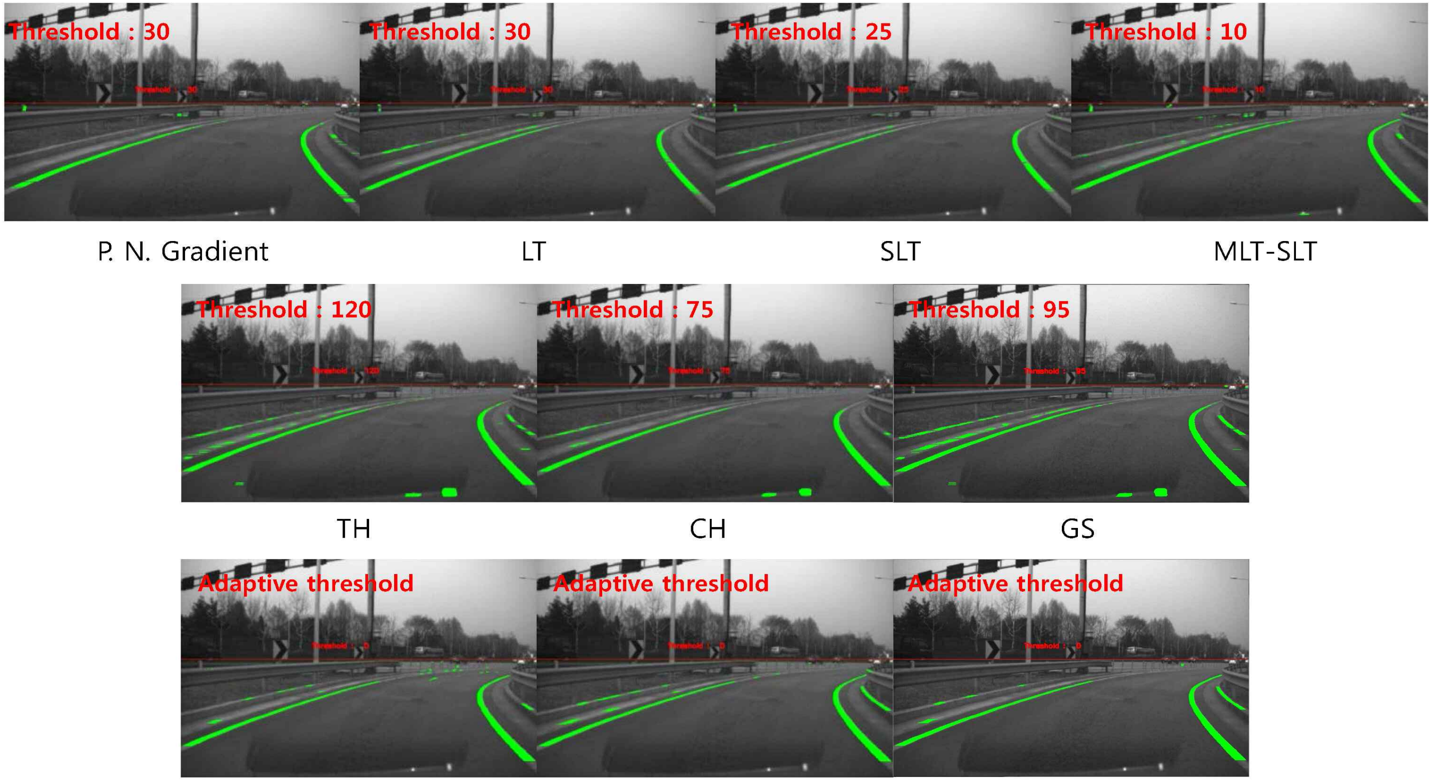
GS

RITR-TH

RITR-CH

RITR-GS

(h) An example of lane marking extraction results in the “Tunnel (rainy day)” scenario.



P. N. Gradient

LT

SLT

MLT-SLT

TH

CH

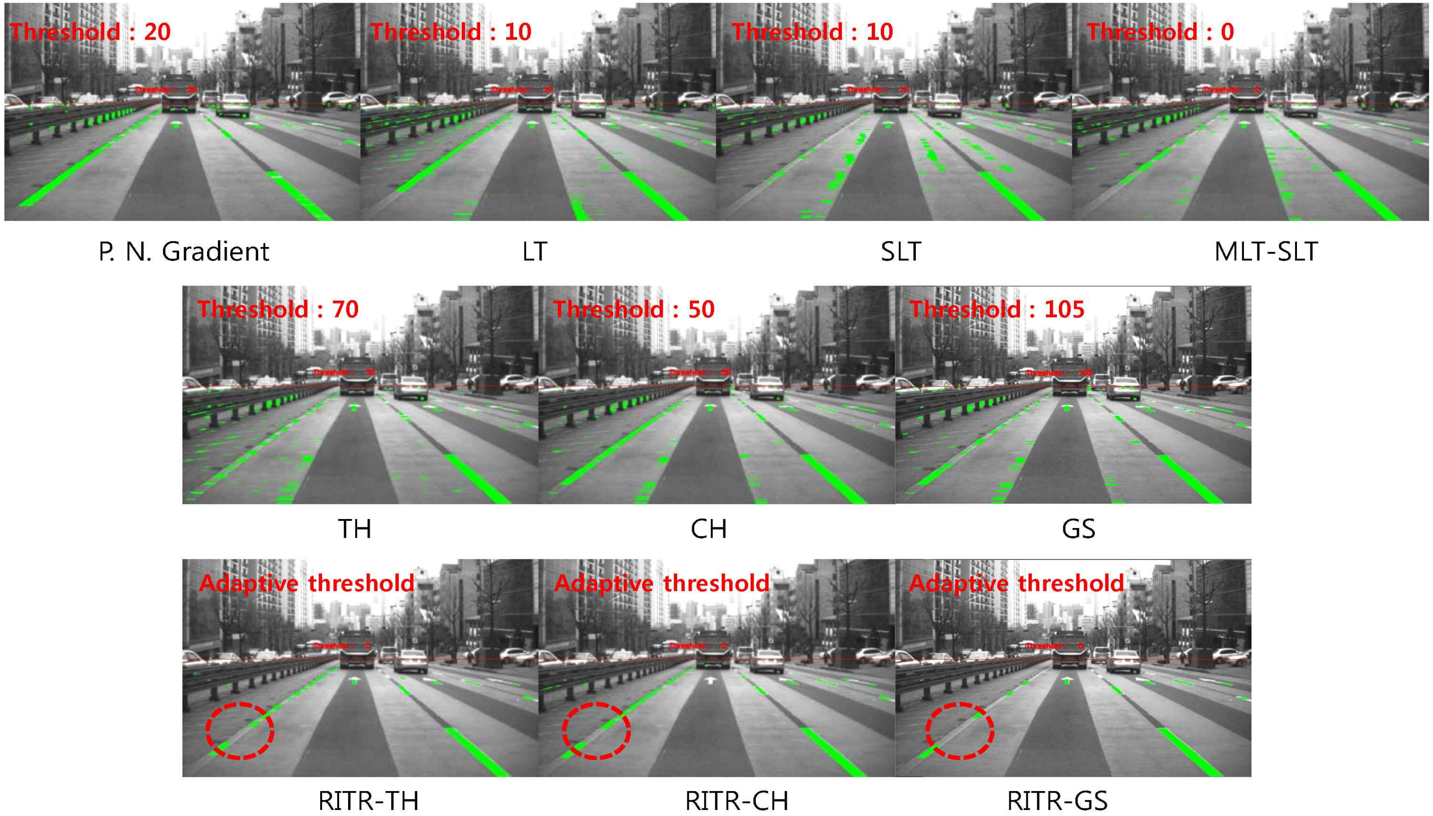
GS

RITR-TH

RITR-CH

RITR-GS

(i) An example of lane marking extraction results in the "Curve" scenario.

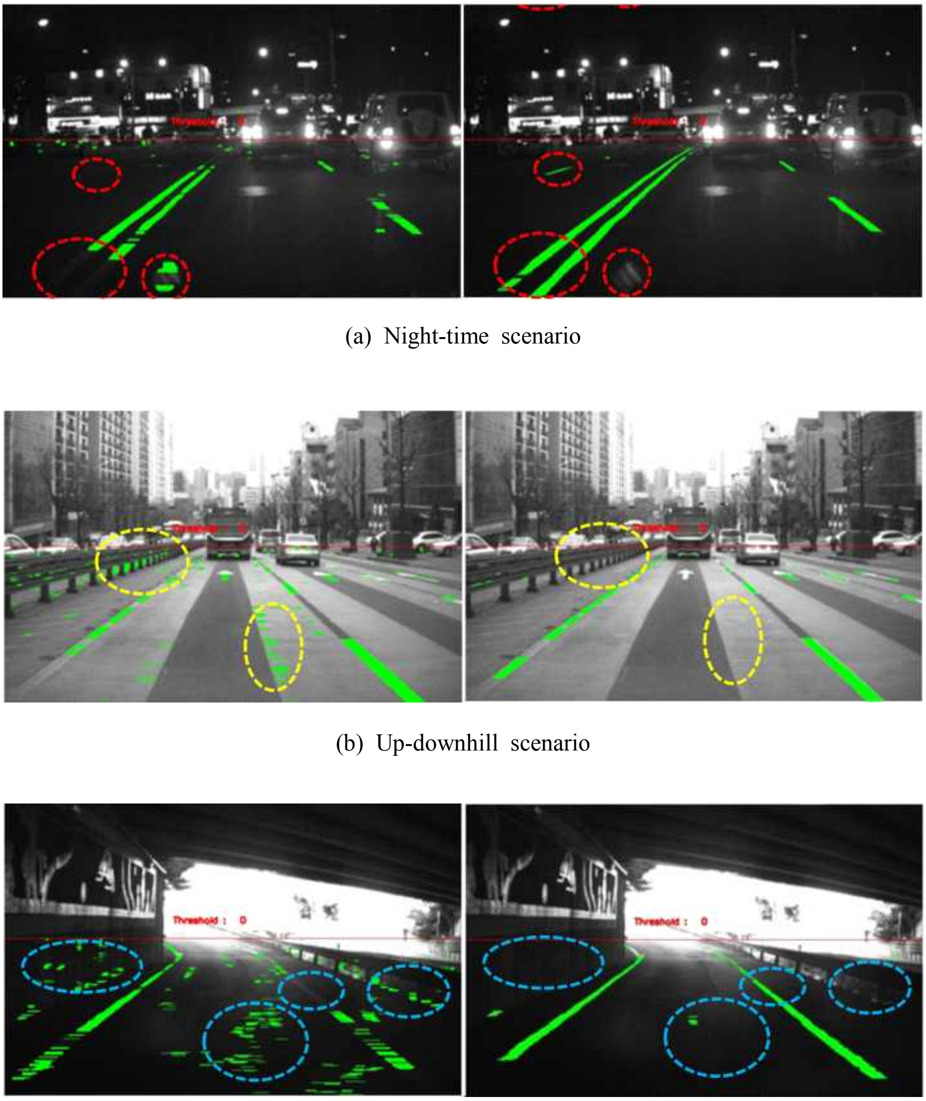


(j) An example of lane marking extraction results in the “Up-downhill” scenario.

Figure 2.14. Examples of lane marking extraction results.

The green pixels represent the lane markings extracted. Representative examples of each scenario are shown in Figure 2.15 (a-j). We selected the best extraction image of each algorithm over varying thresholds, for fair comparison. Please remember that these algorithms are not the lane detection algorithm, but lane marking candidate extraction algorithm. Taking that into consideration, most algorithms extract lane marking candidates very nicely, in most cases. However, we could observe that more false positives appear in both the worse illuminative environment such as nighttime, urban road (rainy day), tunnel (daytime) and tunnel (rainy day) scenarios (Figure 2.15 (d), (f), (g) and (h), respectively), and also show on an object, such as vehicles and barriers (Figure 2.15 (a), (c), (e) and (j), respectively).

Some reasons for FPs generation are considered that firstly, the lighting conditions, such as cluttered shadows, or reflections on the road, may produce a dark-bright-dark pattern, and it is easy to be extracted by the algorithm. Secondly, a pattern like the lane marking on an object such as vehicles, barriers and curbs, also leads to FP generation. Finally, the threshold not fully adjusted may either increase FPs (the lower threshold case), or decrease TPs (the higher threshold case). From the point of false positive cancelling, one of the most important observations here is that the false positive cancelling algorithm being applied to the reiterated extraction of the top hat (cone hat) impressively eliminates FPs. Some extraction result comparison between MLT-SLT and RITR-CH is shown in Figure 2.16.



(a) Night-time scenario

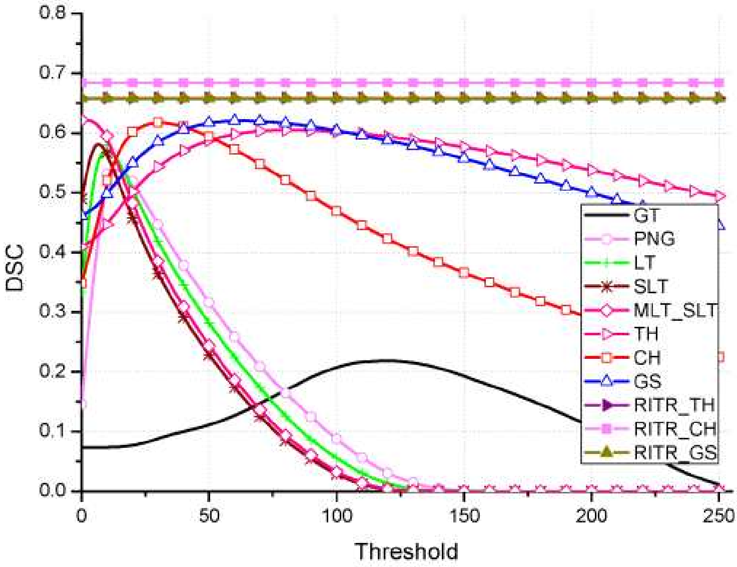
(b) Up-downhill scenario

(c) Tunnel (daytime) scenario

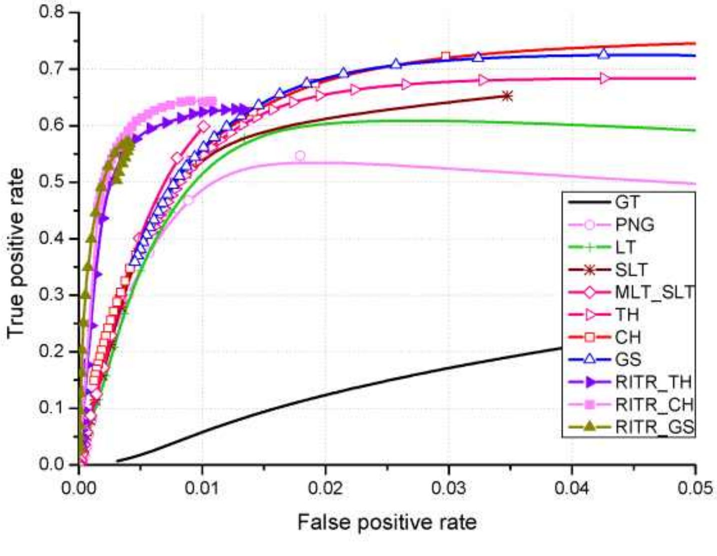
Figure 2.16. Extraction results comparison between MLT-SLT (first column) and RITR-CH (second column) in some worst cases is shown above. RITR-CH provides less FPs and more TPs than MLT-SLT.

The overall experimental results, including DSC and ROC curves, are shown in Figure 2.17. The nine algorithms in Table 2.4 were applied to all the images in the database. In accordance with the experimental result in [18], symmetry local threshold, local threshold and positive negative gradient perform well, in decreasing order. And, double extraction [12] of the symmetry local threshold shows higher performance than symmetry local threshold.

The fact that double extraction performs better may be due to the complementary relationship between the lower and higher thresholding results. In experiments, top hat performs 4.3% higher than symmetry local threshold and local threshold in DSC measure, and the score is as high as double extraction of the symmetry local threshold. This result is different from the result in Ref. [18]. We suppose the reason for the difference arises from the additional symmetry constraints, equation (6), applied for the top hat (cone hat) filter. Additionally, due to the proposed false positive cancelling function, the proposed RITR-TH and RITR-CH perform better than MLT-SLT, and in the case of RITR-CH, the peak value was observed to be 15.9% higher than SLT. Experimental results show RITR-CH has 3.9% higher performance as compared to RITR-TH. We suppose the main reason may be due to cone hat being able to extract lane markings better than top hat, when the lane marking width is not estimated well, since in terms of the shape of the filter, TH is more normalized than CH, while CH is weighted more in the centre of the lane marking. Finally, the most important experimental result is that the value in the DSC graphs of all the algorithms, except RITR-TH and RITR-CH, is changing, according to the threshold value. Therefore, the existing conventional algorithms have to find the peak value with considering the current illumination condition, while the proposed RITR-TH and RITR-CH algorithm are not



(a) DSC graph of the overall experimental results.

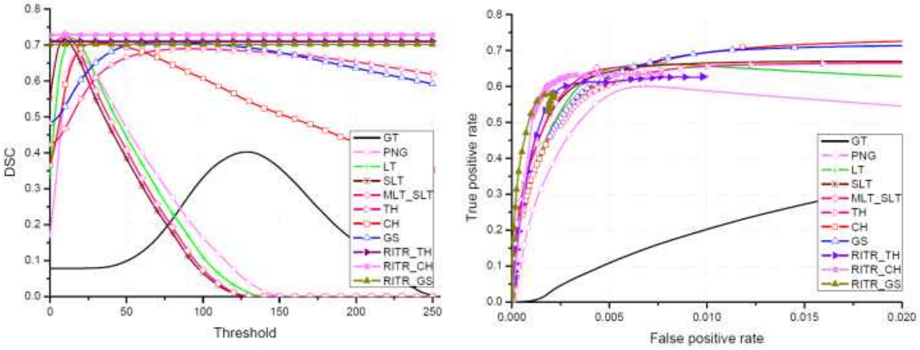


(b) ROC graph of the overall experimental results.

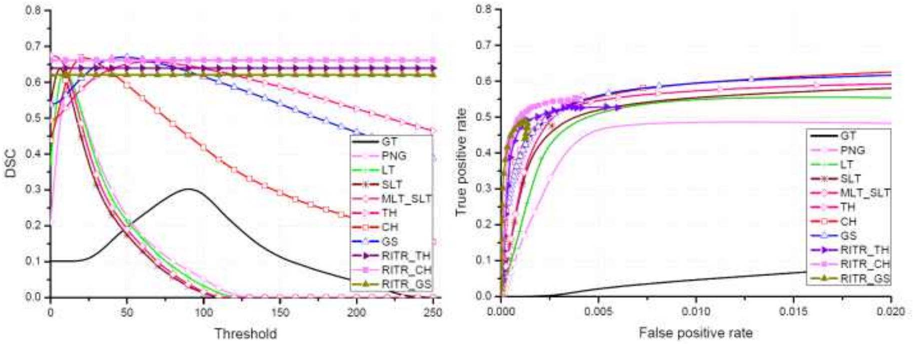
Figure 2.17. Left and right graph represent the DSC and ROC graph of the overall experimental results, respectively.

required to find the optimal threshold. This means that when the algorithm extracts lane markings against a dynamically changing illumination environment from a fast-moving vehicle, instead of depending on the difficulty of the recognition of the illumination environment and the immanent errors contained in statistics to determine the optimal threshold, we can utilize the more robust and reliable reiterated extraction algorithm for the extraction. In ROC graph, due to the false positive cancelling function, the false positive rate of the proposed algorithm is much lower than the other algorithm.

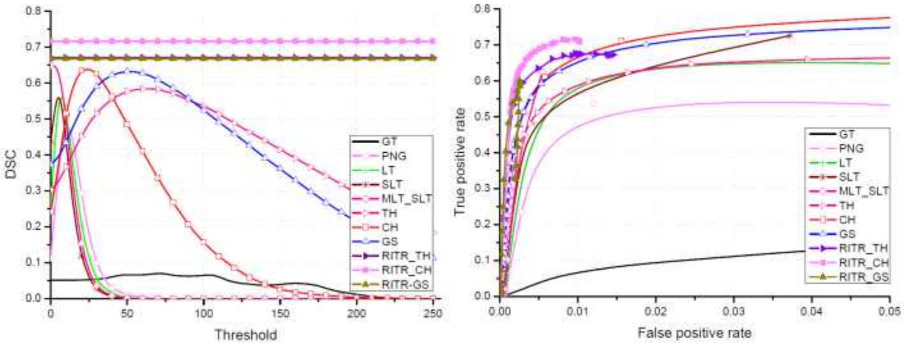
Figures 2.18 (a)-(j) show the experimental results of images in scenarios previously shown in Table 2.2. In DSC graph, we can see the peak value varies, according to each scenario. In a good illumination environment, such as the not shaded road case, most algorithms show a similar very high peak value, as shown in Figure 2.18 (a). However, as the illumination environment gets worse (Figure 2.18 (b)-(d)), we can see that the peak value is getting lower. This means that the performance of the extraction algorithm is affected by the illumination environment. But, one of the important things is that the proposed RITR-TH and RITR-CH have a standard deviation of 0.015 and 0.023, respectively, while SLT has that of 0.075, so we can clearly conclude that they are relatively illumination-tolerant, as compared to the other algorithms. Figures 2.18 (e)-(f) show how the geometrical changes (curve and up-downhill) make differences in the performance of the extraction algorithms. Though the extraction performance in curve is 10% lower than the one of the not shaded road case, the performance is acceptable, which is as high as that of the shaded road case. In addition, as compared to the performance of SLT, which is not required to estimate the lane marking width dynamically, and considering that the performance of TH, CH, RITR-TH and RITR-CH is not degraded so much, we can conclude that the estimation of lane marking width applied to the TH and CH filters works well. Figures 2.18 (g)-(j) show the performance degradation between daytime and rainy day, in urban road and tunnel scenarios. Undoubtedly, due to the reflections on the road coming from the rear lamps, and sunlight on rainy days, we can observe a performance degradation of 8.7%, as compared to the one of the daytime case in DSC graph. The important observation here is that when the weather condition or illumination environment is getting worse (e.g. rainy day, or in tunnel) the gap of performance between RITR-TH (CH) and the other algorithms shown above is getting larger, due to the false positives.



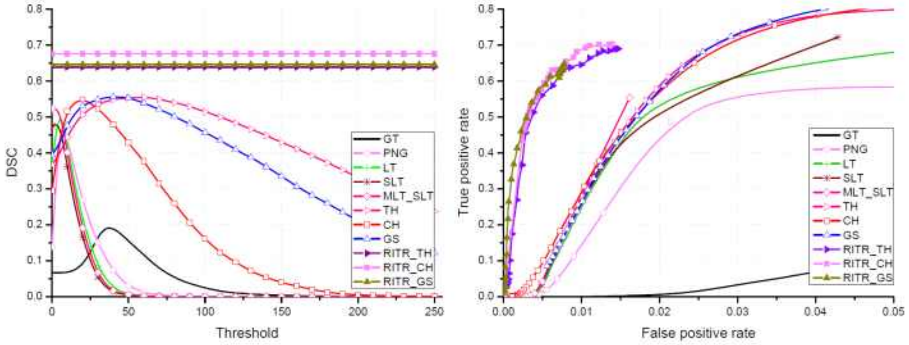
(a) DSC and ROC graph of the “Not shaded road” scenario.



(b) DSC and ROC graph of the “Shaded road” scenario.

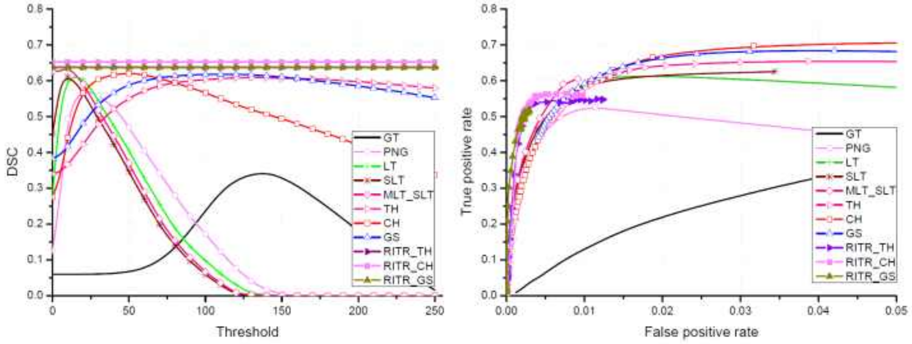


(c) DSC and ROC graph of the “Sunset” scenario.

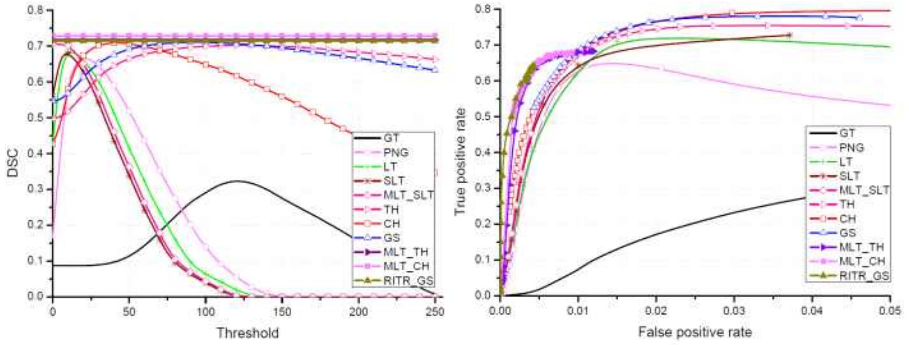


(d) DSC and ROC graph of the "Nighttime" scenario.

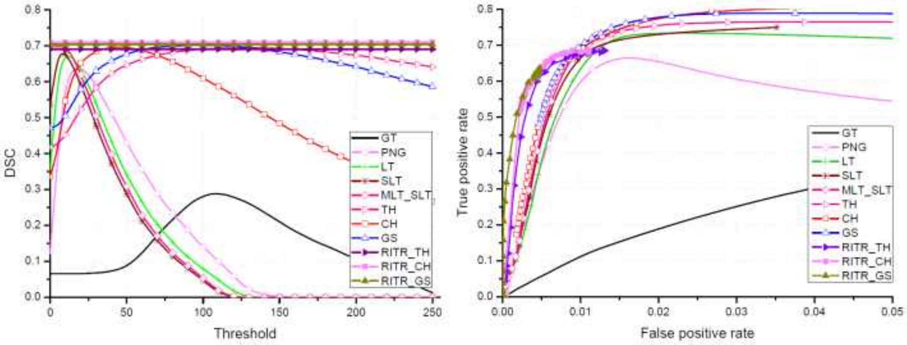
.



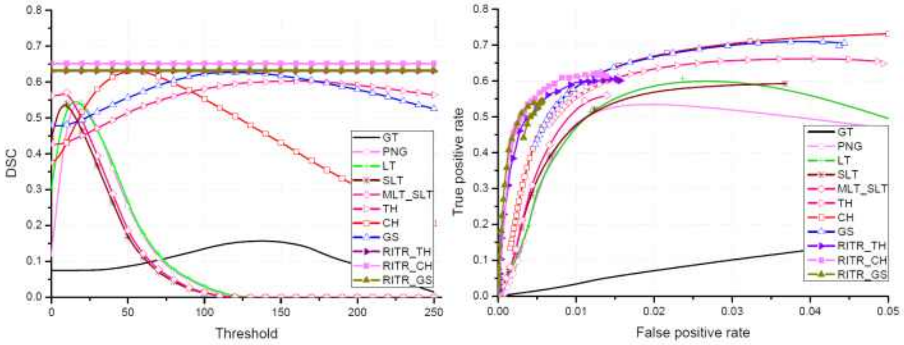
(e) DSC and ROC graph of the "Curve road" scenario.



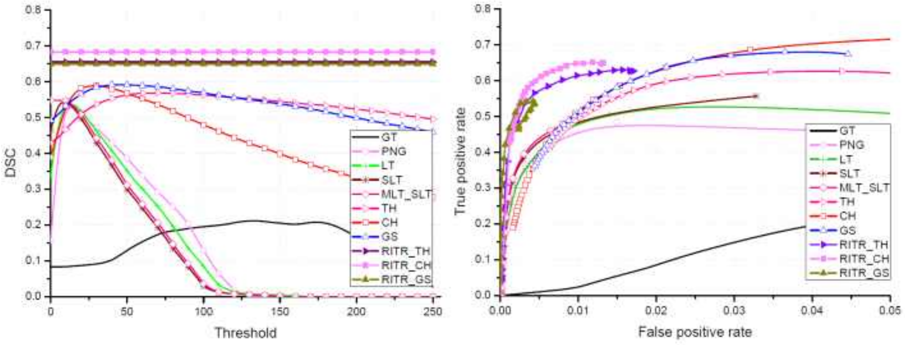
(f) DSC and ROC graph of the "Up-downhill" scenario.



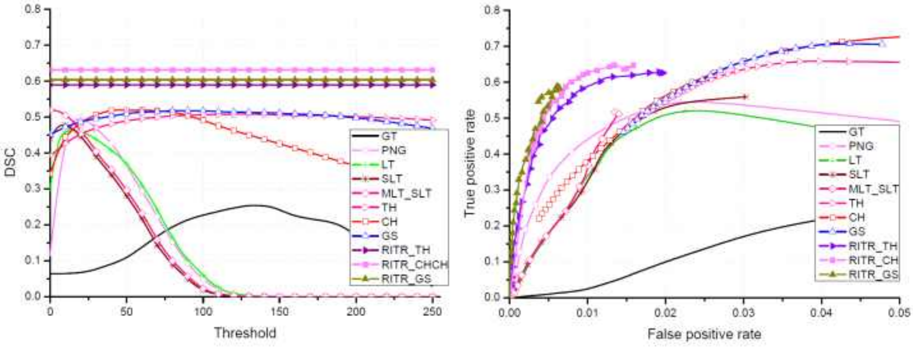
(g) DSC and ROC graph of the "Urban road (daytime)" scenario.



(h) DSC and ROC graph of the "Urban road (rainy day)" scenario.



(i) DSC and ROC graph of the "Tunnel (daytime)" scenario.



(j) DSC and ROC graph of the "Tunnel (rainy day)" scenario.

Figure 2.18. DSC and ROC graph of scenarios: (a) not shaded road, (b) shaded road, (c) sunset, (d) night-time, (e) curve, (f) up-downhill, (g) urban road (daytime), (h) urban road (rainy day), (i) tunnel (daytime), and (j) tunnel (rainy day)

In terms of DSC, we adopted the K-factor to evaluate the performance of the reiterated extraction algorithm. As shown in Figure 2.6, the reiterated extraction algorithm finds the average of the filter response values as a threshold. the K-factor is utilized to differentiate the threshold value from zero to the 2.5 times of average. The experimental result shows that the peak value of DSC is shown at the point that the K-factor is 1.0 as shown in Figure 2.19. This indicates that the average of the filter response is appropriate for the threshold of the reiterated extraction algorithm.

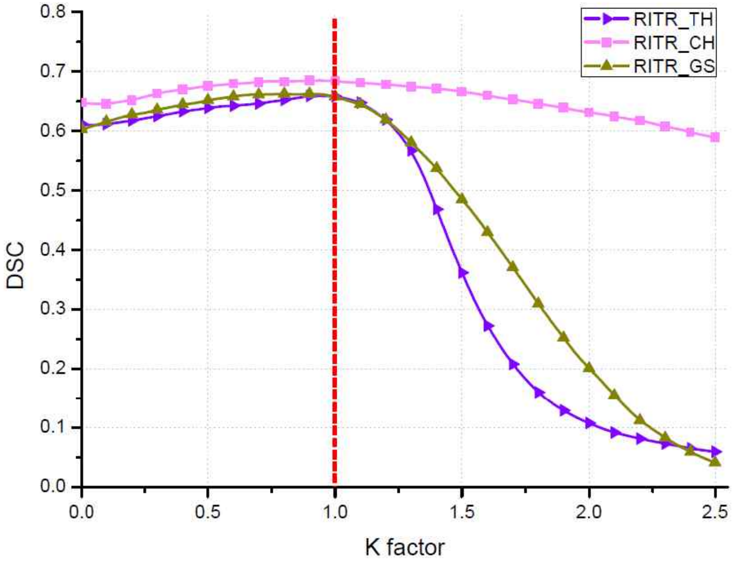


Figure 2.19. Comparison of DSC among RITR-TH, RITR-CH and RITR-GS. The peak value is shown at the point of the K factor is 1.0.

In terms of execution time, the proposed top hat (cone hat) filter is more than two times faster than LT, SLT and MLT-SLT algorithms, while providing as high a performance as MLT-SLT algorithm. In addition, the FAST RITR-CH algorithm performs about three times faster than the MLT-SLT algorithm, which is exactly the same algorithm as RITR-CH, but the size of the source image is vertically and horizontally half of the original image. It is over 40 fps in the PC environment. An overall execution time comparison is shown in Figure 2.20.

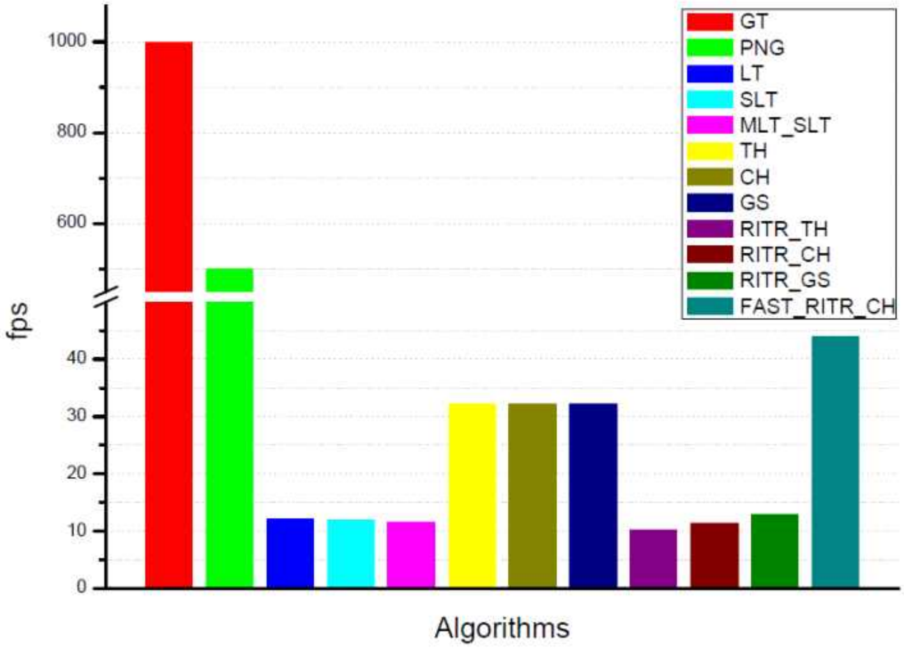


Figure 2.20. Comparison of computation delay. TH and CH are more than two times faster than SLT in our experiments. RITR-TH and RITR-CH are slightly slower than SLT. Fast RITR-TH(CH) is four times faster than SLT.

2.5 Summary

In this paper, we propose a cone hat filter that provides robust and reliable lane marking extraction and a reiterated extraction algorithm that can determine the threshold values adaptively, and filter out false positives. As a result, the proposed algorithm shows higher performance in term of DSC peak value, as compared to the five conventional lane marker extraction algorithms. For our experiments in all the scenarios, the proposed RITR-CH algorithm performed about 10-15 percent higher on average. In addition, it is notable that as the illumination conditions worsened, the gap of the DSC peak value between RITR-CH and the other algorithms became larger. This fact supports that the proposed algorithm has illumination-tolerant characteristics. Furthermore, one of the remarkable benefits of the proposed approach is the easiness of finding the optimal threshold; it is invariant in all scenarios, no matter what the illumination, or whether the condition is changed. In terms of execution time, the proposed top hat (cone hat) filter is over 30 fps, and the fast RITR-CH algorithm performs over 40 fps.

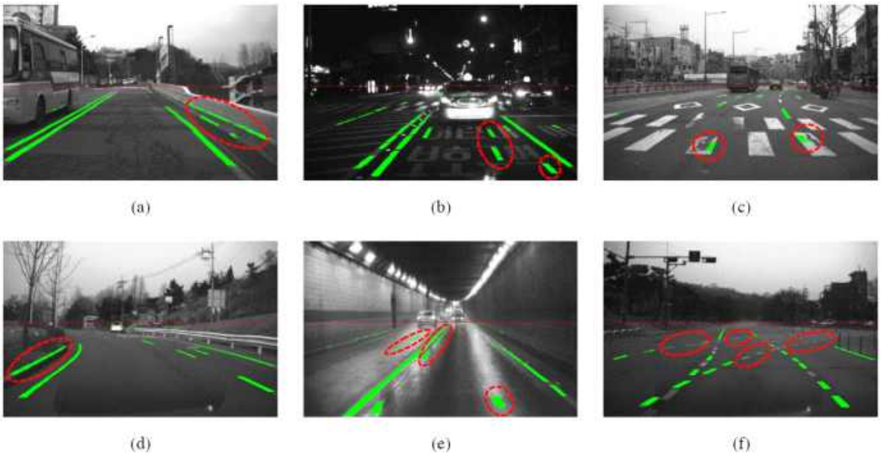


Figure 2.21. Some examples of false positives and false negatives are shown: (a) Barriers, (b) Characters, (c) Worn-out road markings, (d) Curbs, (e) Reflections, and (f) Wrong directional lane markings

Chapter 3

Fusing Road Detection and Lane Position Estimation for Robust Road Boundary Estimation

3.1 Introduction

Road detection provides information about the available free space into which a vehicle can move compared to lane detection which provides lane information. If there are no lane markings on a road, such as in an off-road environment, then road detection is a very useful tool that can replace lane detection. For this reason, road detection can be regarded as a secondary option for driving assistance when lane detection is not available to detect lane markings. However, if we integrate road detection and lane detection into a single system, the shortcomings of each algorithm can be compensated for to handle such difficulties more easily. In addition, the combination of the two different algorithms allows for the investigation of new and useful information that cannot be retrieved using either system alone.

Existing road detection algorithms can be categorized into moving camera-based and stationary camera-based algorithms. In this paper, we focus on moving camera-based algorithms. The moving camera-based approaches can be classified again based on the type of information source: (1) monocular image-based, (2) stereo disparity-based, and (3) a combination of the two algorithms. These approaches can be characterized as follows.

Among the monocular image-based approaches, the chromaticity-based approach shows robust detection results for shaded areas. It investigates the chromaticity of the road and classifies the pixels within the range of road chromaticity [22]. The road model generated by the combination of the many color models seems to be useful. They classify the road pixels using machine learning such as a support vector machine based on their own road model [23]. The chromaticity-based approach can detect the shaded road area very nicely but has difficulty detecting over-saturated and under-saturated pixels due to the lack of color information in the pixels. The color model combination-based approach seems to have difficulty with detection of shaded areas.

In the case of the stereo disparity-based approaches, the vertical disparity-based approach is the most frequently used due to its simple but strong detection capabilities [24]. However, the detectability depends on the accuracy of the disparity map, and in general, the detection distance is shorter than that of a monocular image-based algorithm.

Other useful approaches include the occupancy-grid and stixel-based approaches [25, 26]. The main goal of those two approaches is to distinguish between objects and free spaces in the disparity image. In terms of the combination approach in which both monocular and stereo algorithms are used, a combination of the chromaticity and v-disparity methods has shown good detection results [27]. Image segmentation and the v-disparity based classification method are also presented [28]. The former is effective for shaded road detection and boundary detection but the computation cost is high. The latter shows detection results depending on the segmentation results, but the segmentation is not free from shadows.

In this paper, we focus on the monocular image-based approach because of its price competitiveness when compared to the stereo disparity-based approach. Among the monocular image-based approaches, the chromaticity-based approach was chosen due to its shadow-tolerant characteristics. We propose a hybrid approach that utilizes the results of road detection and the information from lane position estimation. One of the challenges that road detection faces is correctly determining the road boundary. However, the chromaticity-based approach cannot detect over-saturated and under-saturated areas, which often leads to road boundary detection failure. Likewise, the lane position estimation algorithm can calculate the position of proximal lane markings; however, the verification of the existence of a lane marking is not easy. Combining the information gained from each algorithm can ease the difficulty created by this problem. If the initial road boundary reported by road detection shows correspondence with an estimated lane marking position, then the estimated lane marking position has a high probability of being the correct road boundary. The proposed approach shows very good results for detecting the road boundary. Moreover, thanks to the combined information, position information about the current lane can be provided since the proposed approach can provide information about the total number of lanes in the detected road area.

The remainder of this paper is organized as follows. In Section II, we introduce how to detect the road using the chromaticity approach. In Section III, we introduce the proposed lane position estimation algorithm. In Section IV, the hybridization of road detection and lane position estimation is explained, and experiments and evaluation are presented in Section V. Finally, we conclude in Section VI.

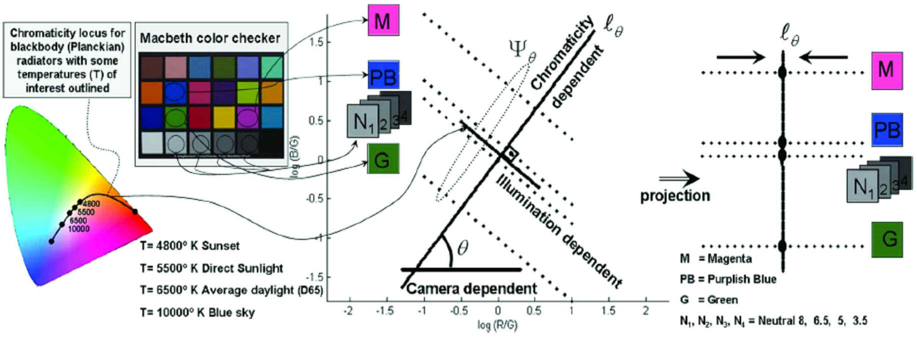


Figure 3.1. Ideal log-log chromaticity plot [22]. The straight lines associated with each line are obtained by changing the color temperature of the illuminator. As a result, patches of different chromaticities have different line associated with them. The main property of l_q is preserving differences with regard to chromaticity but removing differences due to illumination changes.

3.2 The Chromaticity-based Flood-Fill Method

3.2.1 Illuminant-Invariant Space

Under the assumptions of Lambertian reflectance, approximately Planckian lighting, and fairly narrow-band camera sensors, the RGB features of an image captured by camera could be represented as:

$$R_i = \sigma E(\lambda_i) S(\lambda_i) q_i,\ \ i = R, G, B \tag{1}$$

where, s is a Lambertian shading, $E(l_i)$ is the spectral power distribution, $S(l_i)$ is the surface spectral reflectance function, and q_i is the sensitivity of the camera. The chromaticity is defined as:

$$c_k = R_k \Big/ \sqrt[3]{\prod_{i=1}^{3} R_i}\ \ or\ \ c_k = R_k \Big/ R_{(3-k)} \tag{2}$$

where, R_k is the channel for red, green and blue and $R_{(3-k)}$ is the one channel for the other two colors. Illuminant-invariant space is defined by taking the logarithm of chromaticity as below:

$$\rho_i \equiv \log(c_k) = \log(s_k / s_{(3-k)}), \quad i = 1,2 \quad . \tag{3}$$

where, $\rho_{1,2}$ form a illuminant-invariant space as shown in Figure 3.1. The dotted lines in the middle of Figure 3.1 represent the chromaticity values varying under the changes the color temperature K. The important thing is that the values of each colors are laid along with the lines, i.e. when the illumination conditions are changed, the axis of color in the log-chromaticity space is invariant. To get this unique value, we have to calculate *q*, the angle of line l_q, with the family of parallel lines Y_q. Remember that *q* is the camera dependent intrinsic parameter and thus does not need to be recalculate again [22]. To find the correct axis *q*, Shannon's definition of entropy has been adopted as below:

$$\varepsilon_\theta = -\sum_i^L H_\theta(i) \log(H_\theta(i)) \quad . \tag{4}$$

where, e_q is the entropy of I_q and I_q is the gray scale image obtained from projecting the pixel values of the log-chromaticity space onto a line l_q. *L* is the number of bins of histogram H_q. When the entropy reaches a minimum value over the range of *q*, the angle is useful in distinguishing different colors under varying illumination changes. Figure 3.2 illustrates the scheme of the entropy calculation.

Figure 3.3 shows the results of searching the calibration axis using entropy calculation. As shown in Figure 3.3(b), the angle having the minimum entropy exists and the difference in gray scale intensity becomes minimal at the angle having the minimum entropy (refer to the red box in Figure 3.3(c), (d)).

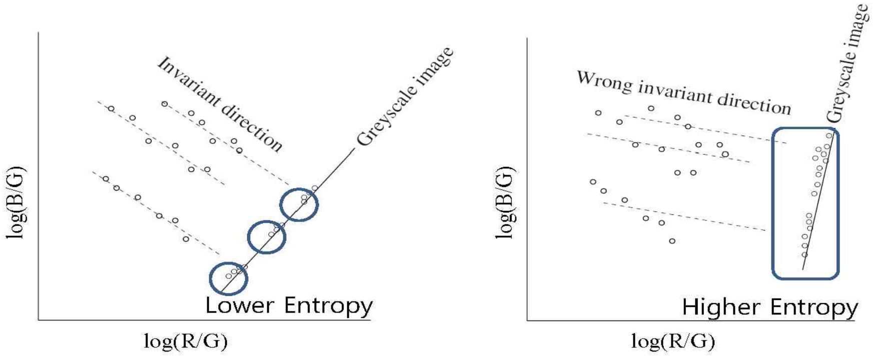
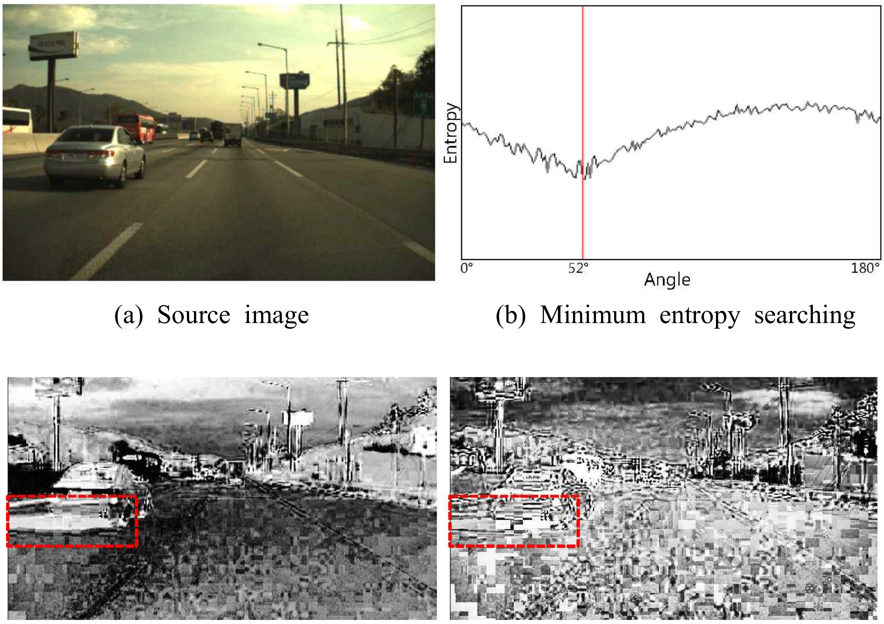


Figure 3.2. Schematic illustration of the entropy calculation.



(a) Source image

(b) Minimum entropy searching

(c) No minimum entropy result

(d) Minimum entropy result

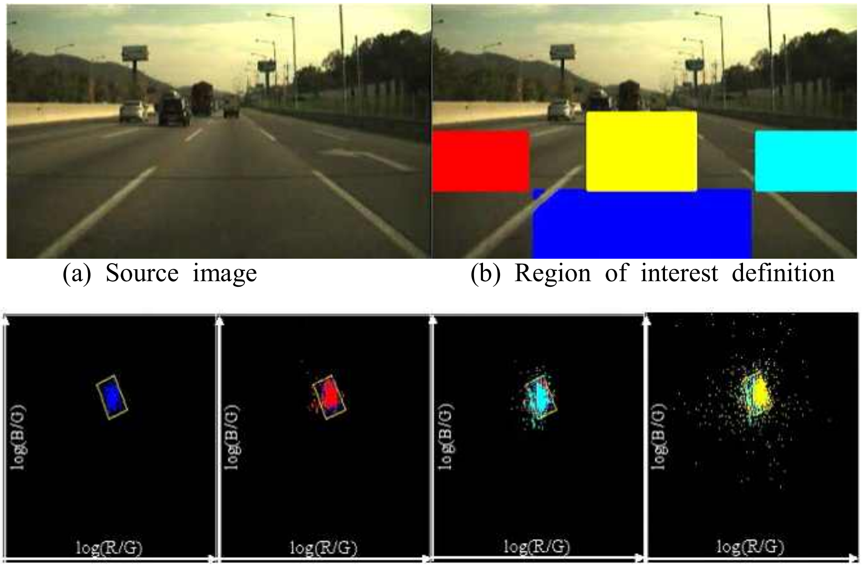
Figure 3.3. (b) Result of entropy calculation. The red line represents the minimum entropy. (c) and (d) present a snapshot of the minimum entropy searching process. The chromaticity value under the shadow came very close to that of the non-shaded road area at the projection angle having minimum entropy.

3.2.2 Road Pixel Selection

To detect the road pixels, the range of the road color must be investigated. Therefore, we converted the pixels in the pre-defined four areas from the RBG color space into the log-chromaticity space as shown in Figure 3.4. After that, we found the range of the road chromaticity. For a more accurate range selection, we dropped the furthest 10% from both the left and right ends.

3.2.3 Flood-fill Algorithm

The flood-fill algorithm [29] finds connected neighbor pixels starting from a seed pixel with a specific value. To find the road pixels, we adopted the chromaticity. The connectivity is determined by the chromaticity closeness of the neighboring pixels. The pixel at (x,y) is compared to the seed pixel as below:

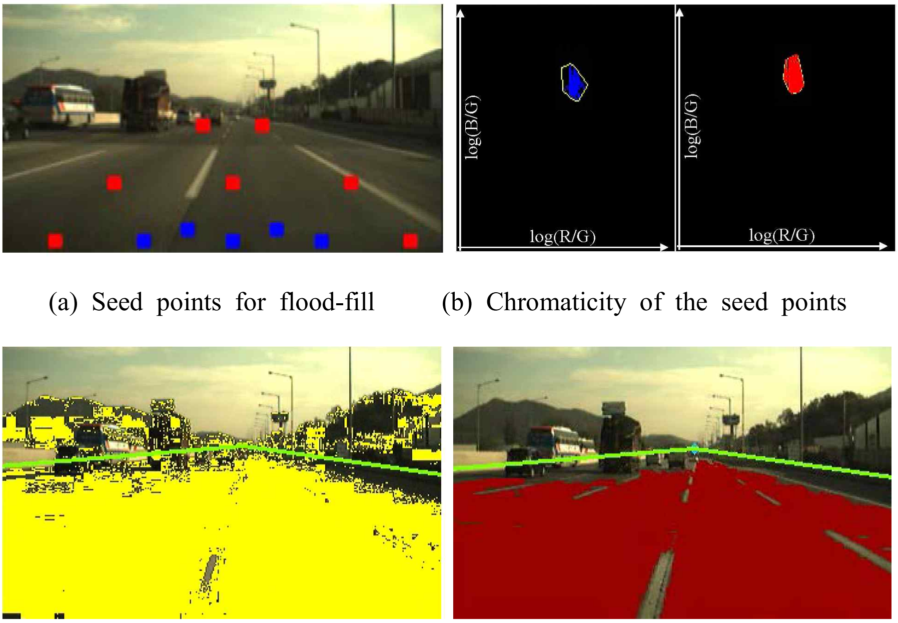


(a) Source image

(b) Region of interest definition

(c) Road chromaticity range selection in the log-chromaticity space

Figure 3.4. Snapshot of the road pixel selection. The regions of interest for the road chromaticity selection are predefined as shown in (b). The most frequent chromaticity in each region is shown in (c).



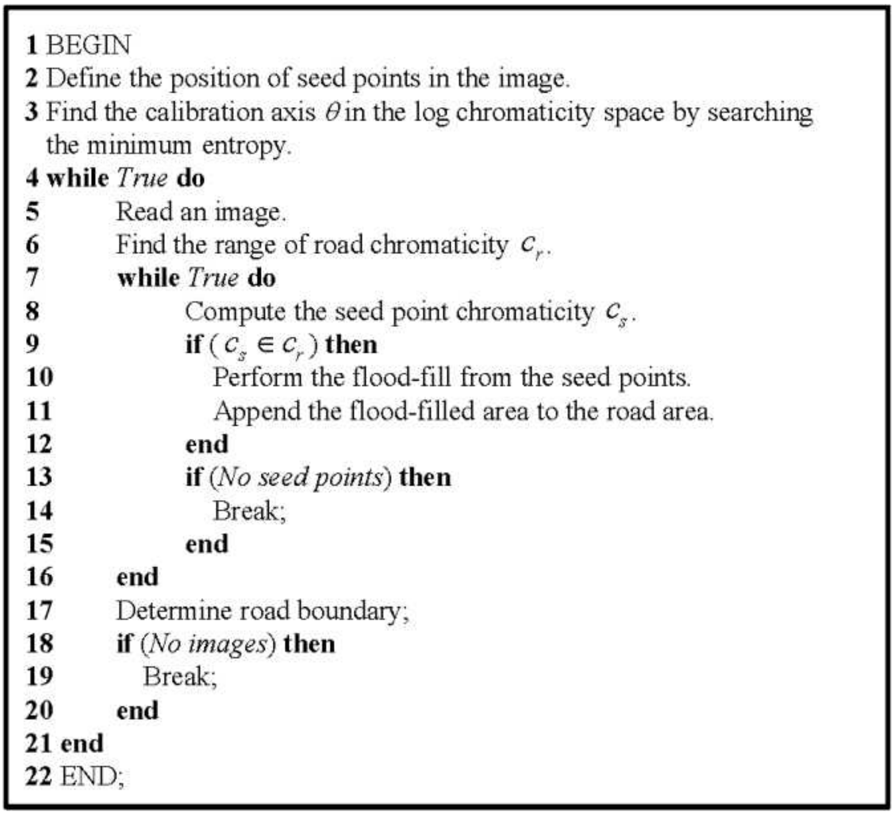
(a) Seed points for flood-fill (b) Chromaticity of the seed points

(c) Pixels within the road chromaticity range (d) Flood-fill result

Figure 3.5. Illustration of the flood-fill process. (a) Predefined seed points. The blue and red represent the inner and outer area of the seed points respectively. (b) Seed point validation by checking chromaticity values. (c) The pixels within the range of the road chromaticity. (d) The result of flood-fill based on the road chromaticity values.

$$C(x_{seed},y_{seed}) - loDiff \leq C(x,y) \leq C(x_{seed},y_{seed}) + upDiff \quad . \tag{5}$$

where $C(x_{seed},y_{seed})$ is the value of seed point. *loDiff* and *upDiff* are maximal lower and upper differences between the currently observed pixel and the seed pixel respectively. The seed points for the flood-fill are previously defined as shown in Figure 3.5(a) and the acceptance test for the seed point is conducted using the road chromaticity. Only accepted seed points can be used for flood-filling to detect the road pixels. The result of the flood-fill is shown in Figure 3.5(d) and the pixels within the range of the road chromaticity are shown in Figure 3.5(c). The pixels in the road chromaticity are used for flood-fill as the candidates for the road pixels. The chromaticity-based flood-fill algorithm for road detection is shown in Figure 3.6.



```
BEGIN
Define the position of seed points in the image.
Find the calibration axis $\theta$ in the log chromaticity space by searching
  the minimum entropy.
while True do
    Read an image.
    Find the range of road chromaticity $c_r$.
    while True do
        Compute the seed point chromaticity $c_s$.
        if ( $c_s \in c_r$ ) then
            Perform the flood-fill from the seed points.
            Append the flood-filled area to the road area.
        end
        if (No seed points) then
            Break;
        end
    end
    Determine road boundary;
    if (No images) then
        Break;
    end
end
END;
```

Figure 3.6. Road detection algorithm made using the chromaticity-based flood-fill.

3.3 Lane Position Estimation

In this section, we propose a new next lane position estimation algorithm by utilizing lane marking extraction [30] and a cross-ratio [31]. During lane position estimation, we continuously generate a homography matrix using the lane markings positions extracted from the lane marking extraction algorithm. After that, the coordinates of the lane markings are converted to bird’s-eye view coordinates using the homography matrix. The bird’s-eye view coordinates of the lanes are utilized as the parameters for the next lane position estimation based on the cross-ratio. Finally, the estimated lane position is then inversely converted to the forward-looking view coordinate.

3.3.1 Lane Marking Extraction

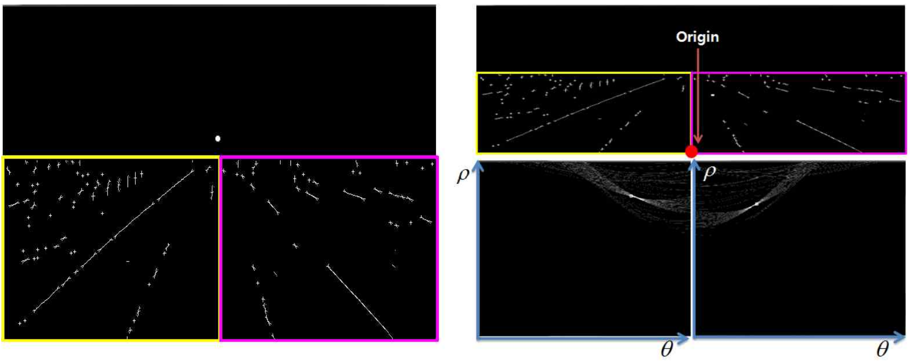
Lane marking extraction is prerequisite for lane detection and provides the set of lane marking candidates with false-positive cancelling results. Since the lane fitting process is not conducted, lane information is not provided. There are several well-known lane marking extraction algorithms, including edge/ridge detection, steerable filter, top hat, family of local threshold, and double extraction. We selected the reiterated extraction algorithm based on the cone hat filter since it is known for being illumination tolerant and false positive cancelling is provided [30].

3.3.2 Proposed Lane Position Detection Algorithm

Reiterated extraction extracts lane markings using the convolution between a cone hat filter and image intensity. The center points of the lane markings are used to compute the position of the vanishing point using the Hough transform as shown in Figure 3.7 [32]. The vanishing point is then utilized for cancelling the set of false positives based on the direction of the lane marking segments. Therefore, the accumulated convolution values from the vanishing point to the border of the image are calculated and displayed from left to right as shown in Figure 3.8. Finally, the position of the lane markings become detectable. The position of lane markings can be detected by using Equation 6 as below:

$$S(i) = \sum_{j=0}^{L_j} f(x_{ij}) * I(x_{ij}) \cdot \tag{6}$$

where, *S(i)* is the sum of filter response along with the search line and *f(x*$_{ij}$*),I(x*$_{ij}$*)* is the kernel of lane marking extraction filter and intensity at position x_{ij}, respectively. x_{ij} is the position on the searching line as shown in Figure 3.8, and *i*, *j*, and L_j are the number of search line, the number of pixels on the search line, and the total number of pixels on the search line respectively.



(a) Search region separation (b) Hough transformation

Figure 3.7. Vanishing point detection by using Hough transform.

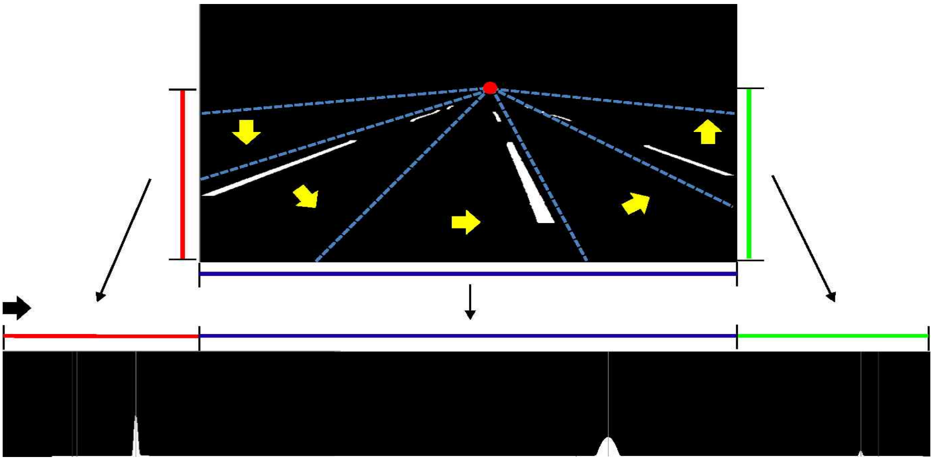


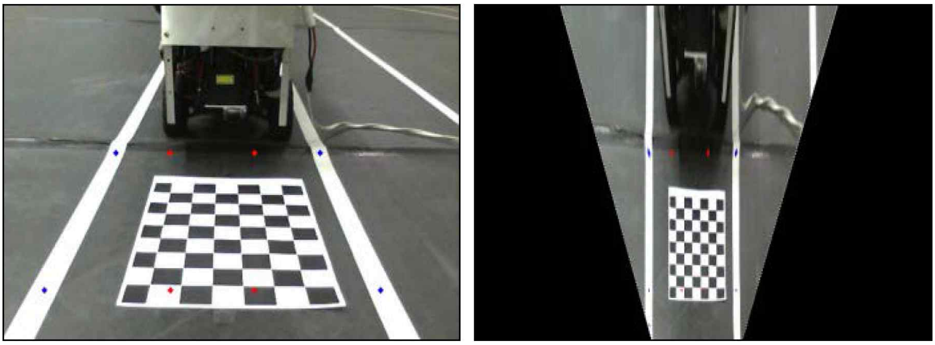
Figure 3.8. Proposed lane position detection algorithm. The blue line searches for the lane position from left to right.

3.3.3 Bird's-Eye View Transformation Using the Proposed Dynamic Homography Matrix Generation

To estimate the position of the next lane marking, we use the known positions of the lane markings. However, these positions have to be transformed into bird's-eye view coordinates. To do that, a homography matrix is necessary, but the calculation of a homography matrix is time-consuming and complex because a checker-board image is required. Additionally, whenever the direction or position of the camera changes, recalibration must be conducted. However, we propose a new homography matrix generation method that can be done on the fly as it requires no complex calculations. Since the proposed method uses the position of the lane markings, it guarantees parallel lane markings after using the bird's-eye view coordinates. The proposed homography calculation method is introduced as follows:

(1) Select four source points to the left and right of the center of the lane markings as the blue points shown in Figure 3.9(a).

(2) Select four destination points inside of the source points as the red points shown in Figure 3.9 (a). Note that the shape of the four red points have to be a rectangle.



(a) Eight-points selection (b) Bird's-eye view transform

Figure 3.9. Homography matrix generation for bird's-eye view transform. The blue points and red points represent the source point and destination point respectively.

By using the eight points, we can calculate the homography matrix as below [33]:

$$X = Hx\ . \tag{7}$$

where X is the vector of the bird's-eye view coordinates, x is the vector of forward-looking view coordinate, and H is the homography matrix. Equation 7 can be rewritten as below:

$$\begin{bmatrix} XW \\ YW \\ W \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} . \tag{8}$$

where, *W* = *gx* + *hy* + *1*, and divided Equation 8 by *W*:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \frac{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}{\begin{bmatrix} g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}} . \tag{9}$$

Rearranging Equation 9 according to X and Y produces:

$$X = \frac{ax + by + c}{gx + hy + 1}, \quad Y = \frac{dx + ey + f}{gx + hy + 1} . \tag{10}$$

$$\begin{aligned} X &= ax + by + c + 0 + 0 + 0 - Xgx - Xhy \\ Y &= 0 + 0 + 0 + dx + ey + f - Ygx - Yhy \ . \end{aligned} \tag{11}$$

Converting Equation 11 to matrix and applying the eight-points results in:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -X_1x_1 & -X_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -Y_1x_1 & -Y_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -X_2x_2 & -X_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -Y_2x_2 & -Y_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -X_3x_3 & -X_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -Y_3x_3 & -Y_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -X_4x_4 & -X_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -Y_4x_4 & -Y_4y_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \\ X_4 \\ Y_4 \end{bmatrix} . \qquad (12)$$

Equation 12 is the form of *Ax=b*, so *x* can be calculated using a pseudo-inverse as below:

$$\begin{aligned} A\lambda &= B \\ A^T A\lambda &= A^T B \\ \lambda &= (A^T A)^{-1} A^T B \end{aligned} . \qquad (13)$$

Using the homography matrix, we can transform the coordinate from the forward-looking view to the bird's-eye view as shown in Figure 3.9. In Figure 3.9(a), we see that the source points are relocated to the position of the destination points. Since the classical homography matrix generation method doesn't update periodically, the bird's-eye view transformation seems to be erroneous in the uphill/downhill. However, since the proposed method continuously updates the homography matrix, it always provides parallel lane markings on the fly.

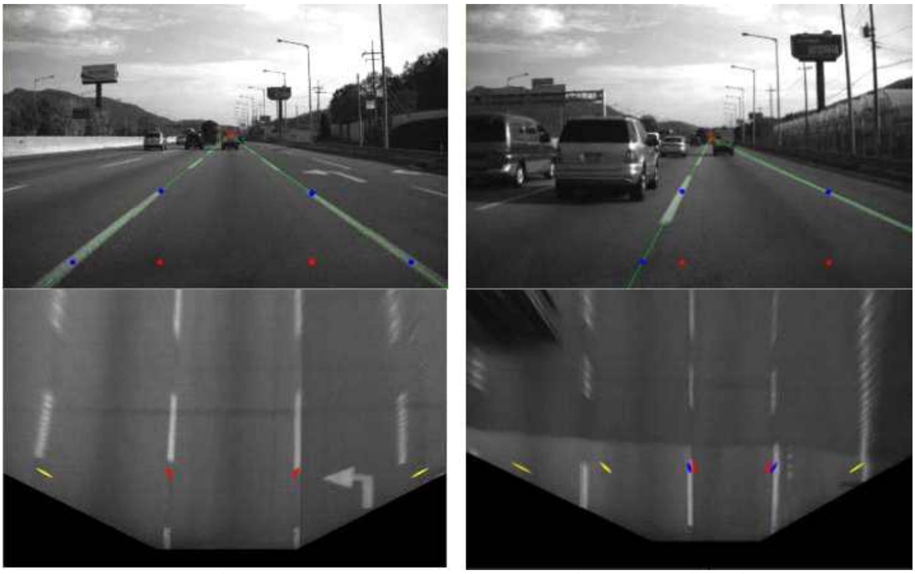
3.3.4 Next Lane Position Estimation Based on a Cross-Ratio

When generating the homography matrix, we utilize the position of the lane markings. In most cases, the lane width is identical, as shown in Figure 3.10(a). However, when the vehicle changes lanes, the width of each lane marking in the bird's-eye view becomes different since the position of the source point is not aligned symmetrically as shown in Figure 3.10(b). In this case, we can estimate the position of the lanes using a cross-ratio. Given four points x_i, the cross-ratio is defined as:

$$\mathrm{C}\left(x_1, x_2, x_3, x_4\right) = \frac{\left|\overline{x_1 x_2}\right| \left|\overline{x_3 x_4}\right|}{\left|\overline{x_1 x_3}\right| \left|\overline{x_2 x_4}\right|} . \tag{14}$$

where,

$$\left|\overline{x_i x_j}\right| = \det \begin{bmatrix} x_{i1} & x_{j1} \\ x_{i2} & x_{j2} \end{bmatrix} . \tag{15}$$



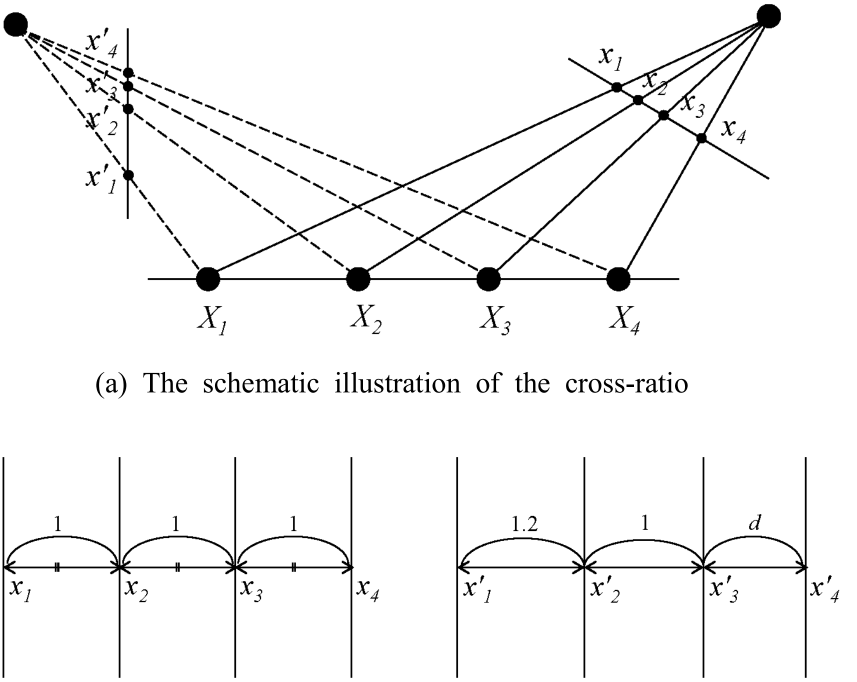
(a) Normal case (b) Lane changing case

Figure 3.10. Bird's-eye view transform results.

If each point x_i is a finite point then Equation 15 represents the signed distance from x_i to x_j. In addition, the value of the cross-ratio is invariant under any projective transformation of the line: if $x' = H_{2x2}x$ then:

$$C(x_1^{'}, x_2^{'}, x_3^{'}, x_4^{'}) = C(x_1, x_2, x_3, x_4) \quad . \tag{16}$$

Based on the assumption that the lane width is unchanging, the cross-ratio has an invariant value. When the vehicle changes lanes, the cross-ratio doesn't change as compared to previous states.



(a) The schematic illustration of the cross-ratio

(b) Finding unknown width *d* by using cross ratio

Figure 3.11. Schematic illustration of cross ratio. (a) Definition of symbols (b) An example of unknown lane width calculation using a cross-ratio.

Therefore, if we have lane markings for more than three positions, we can estimate the position of the upcoming lane markings within the same lane width. For example, if we want to know width *d* as shown in Figure 3.11(b), we can use the following calculations:

$$C(x_1, x_2, x_3, x_4) = \frac{1 \times 1}{2 \times 2} = 0.25 \quad .\tag{17}$$

$$C(x_1^{'}, x_2^{'}, x_3^{'}, x_4^{'}) = \frac{1.2 \times d}{2.2 \times (1 + d)} = 0.25 \quad .\tag{18}$$

According to Equation 16, we can calculate *d* with Equation 17 and 18 as below:

$$d = 0.84 \quad .\tag{19}$$

As shown above, by using at least three points, we can estimate the position of the next lane marking with the cross-ratio.

3.3.5 Forward-looking View Transformation

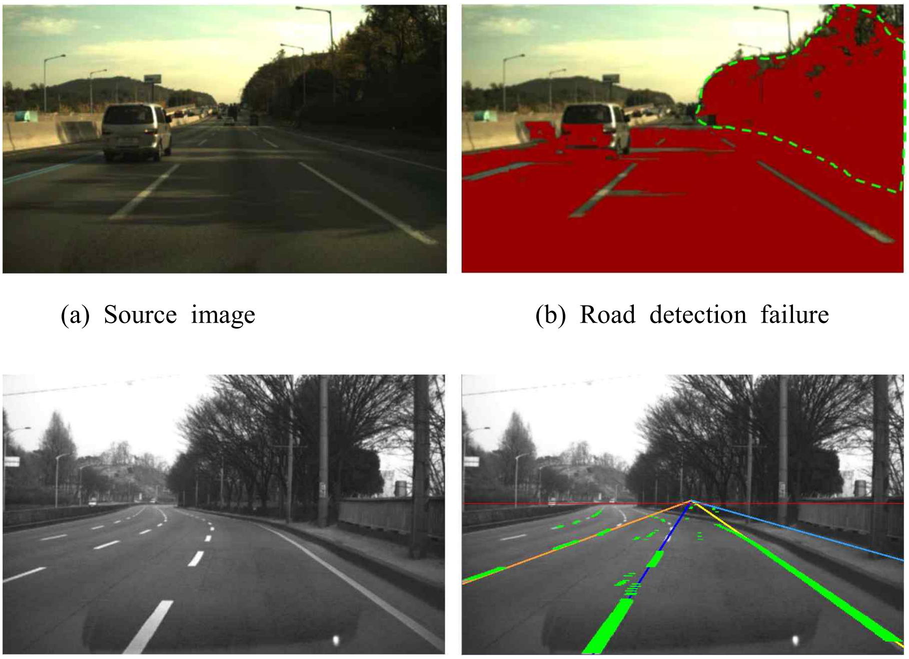
Determining the position of upcoming lane markings using the next lane position estimation can be transformed from the bird's-eye view to the forward-looking view using the inverse matrix of the homography matrix.

3.4 Combining the Information Provided by Road Detection and Lane Position Estimation

The previously introduced method of road detection and lane position estimation has some drawbacks. In this section, we discuss the drawbacks of the algorithms and how to combine the two algorithms. Finally, we introduce the benefits of the hybrid algorithm.

3.4.1 Detection Failures

With regard to the road detection algorithm, when the source image is either over-saturated or under-saturated, detection failure occurs due to the lack of color information in the pixels. This case often happens when sunlight reflects off of road surfaces or when the area outside of the road is shaded and becomes too dark. The former happens very rarely but the latter happens quite often as shown in Figure 3.12(b). In the case of lane position estimation, the position of the lane can still be estimated. However, it is hard to know for how long the lanes extend. In addition, the lane markings on the outside edge of the road are not easy to detect in general. The failure of lane position estimation is shown in Figure 3.12(d).



(a) Source image

(b) Road detection failure

(c) Source image

(d) Lane estimation failure

Figure 3.12. Case of detection failures. (b) Road detection fails due to under-saturation. (d) Lane position estimation fails due to a lack of lane existence information. Road detection needs information about the road boundaries while lane position estimation requires lane existence verification.

The information required to negate the drawbacks for the algorithms are shown as below:

- Road detection requires road boundary information to prevent the failure caused by under-saturation.
- Lane position estimation requires lane existence verification method.

Fortunately, the two algorithms cover for each other's drawbacks. Road detection can provide lane existence information since the information about the area detected can be utilized to infer the existence of outer lanes. Likewise, lane position estimation can support the estimation of the location of the road boundary since lane markings exist at the boundary of the road and the correspondence between the boundary of road detection and the position of the estimated lane marking is very useful information for resolving the boundary decision problem.

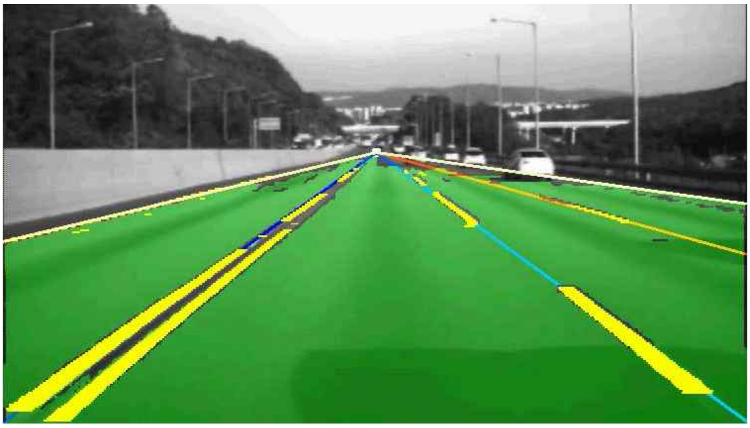


Figure 3.13. Snapshot of the combined information from road detection and lane position estimation. Road detection provides lane existence information while lane position estimation helps determine where the road boundary exists. In addition, thanks to the combined information, the two methods can provide the current lane position and number of lanes on the road.

3.4.2 Benefits of the Combined Information

The two main benefits of joining the two algorithms together are that (1) it allows for the determination of the total number of lanes on the road and (2) it facilitates the recognition of the position of the current lane. These two facts cannot be acquired without the information provided by both algorithms. The results of combining the information from both road detection and lane position estimation are shown in Fig. 13. We can see that the road boundary and lane positions are nicely detected.

3.5 Experiments and Evaluation

In order to test the proposed hybrid method, road detection and lane position estimation experiments were conducted in the experimental environments shown in Table 3.1 and 3.2. For testing in the under-saturated environments, we selected a location with both sunny and shaded road sections. In addition, for testing lane position estimation, we recorded the test video including a lane changing situation on a highway having four lanes. The general testing environment is shown in Table. 3.1.

List	***Specification***	
Camera	MOSS-10 DSP Platform	
Resolution	752 * 480	
Frame per second	30fps	
Total number of frames	1,000 frames	
Color channel	RGB	Road detection
	Gray	Lane marking extraction, Lane position estimation

Table 3.1. Test Environments.

A summary of the testing sequence statistics is shown in Table 3.2. The images consisted of shaded and non-shaded images and the ratio between them is 49% and 51% respectively. The ratio between the close vehicles and no vehicles are 7% to 93%.

List	*Shadows*	*No shadows or dim ones*	*Total*
Close vehicles	4%	3%	7%
No vehicles or further away ones	45%	48%	93%
Total	49%	51%	100%

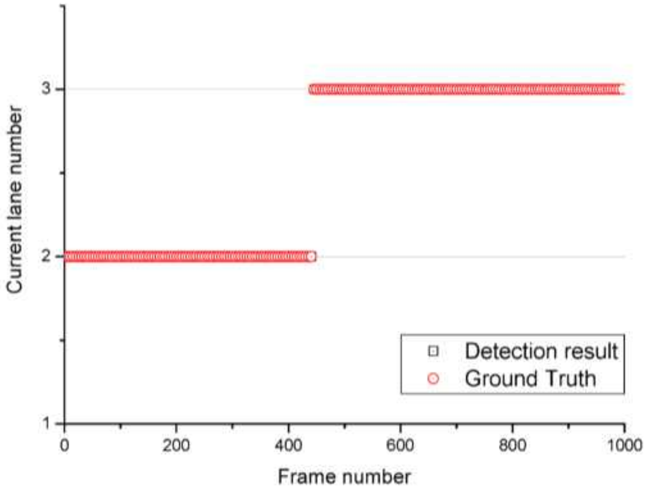
Table 3.2. Summary of Testing Sequence Statistics

For the evaluation, the boundary detection error rate is measured for road detection, and for the hybrid algorithm, we observed how much the detection error rate improved. Additionally, the lane estimation error rate is also measured to observe the performance of the lane position estimation. The experiment results, as summarized in Table 3.3, show that when road detection is conducted with lane position estimation, the boundary detection error rate is fairly high. However, the hybrid algorithm decreases the boundary detection error rate by about ten times from 24% to 1.3%. With regard to lane position estimation, the estimation error rate was slightly higher than the boundary detection error rate. This implies that these two algorithms are tightly correlated.

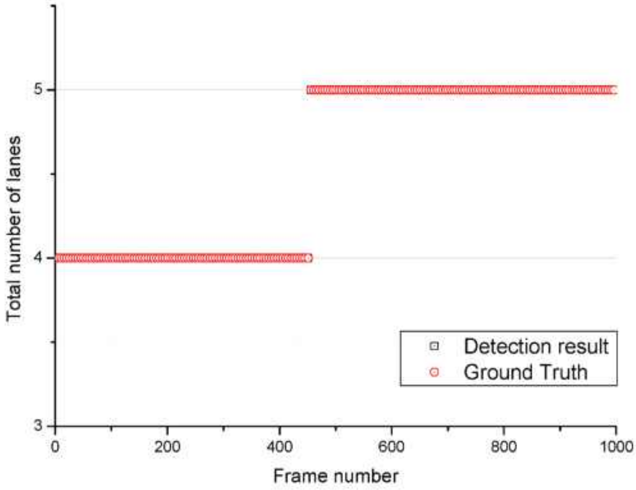
List	*Boundary detection error*	*Lane estimation error*
Road detection	24%	-
Road detection + Lane position estimation	1.3%	1.8%

Table 3.3. Experimental results

The accuracy of the new information about the current lane position and the total number of lanes is plotted as shown in Figure 3.14. The red circle indicates the ground truth while the black square indicates the detection results. The overall detection error was about 0.4%. A snapshot of the road detection results is shown in Figure 3.15.

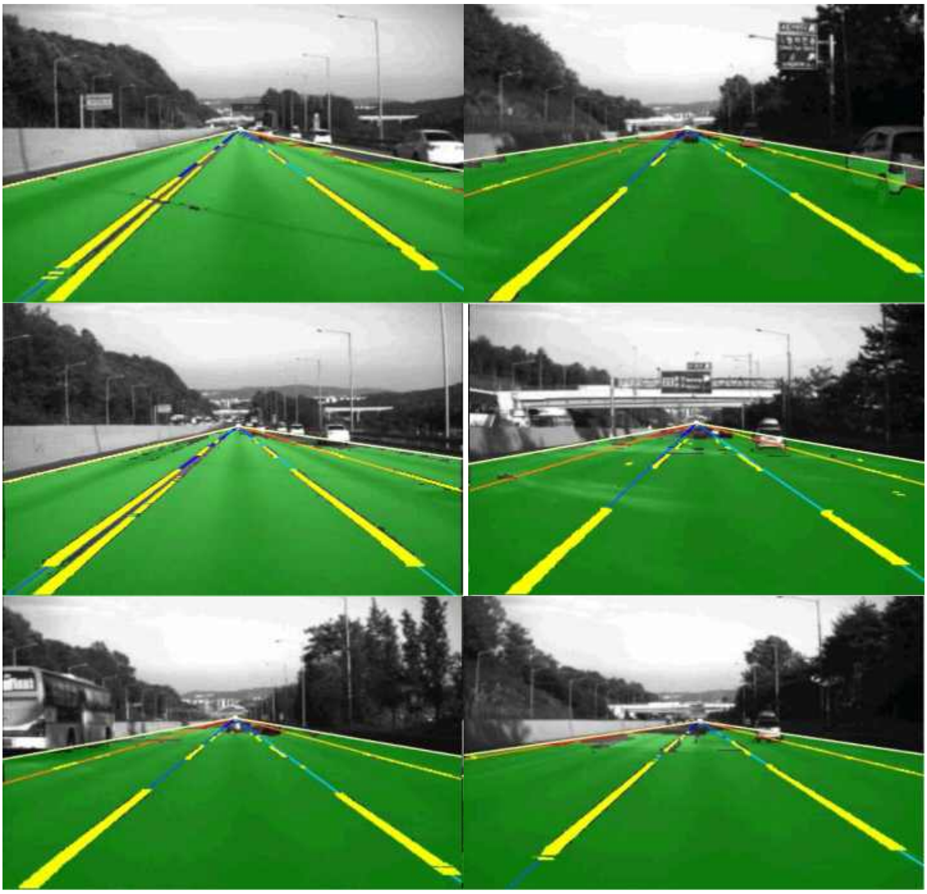


(a) Detection result of the current lane number



(b) Detection result of the total number of lanes

Figure 3.14. The detection result of the current lane position and the total number of lanes.



(a) Non-shaded area (b) Shaded area

Figure 3.15. Snapshot of the detection results based on road detection and lane position estimation.

3.6 Summary

In this paper, we proposed a hybrid algorithm that combines road detection and lane position estimation. For road detection, we adopted the chromaticity and flood-fill methods. For lane position estimation, we adopted a reiterated extraction algorithm for lane marking extraction and proposed a dynamic homography generation method and a cross-ratio-based next lane position estimation algorithm. The experiment results showed that the hybrid algorithm provides more robust road boundary detection and lane position estimation results. Additionally, the hybrid algorithm provided new useful information about the total number of lanes and the position of the current lanes.

Chapter 4

Accurate Inter-Vehicle Distance Measurement based on Monocular Camera and Line Laser

4.1 Introduction

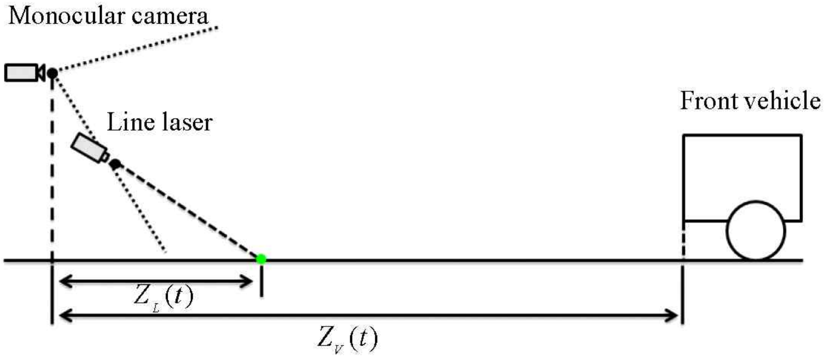
The inter-vehicle distance is regarded as the critical factor of safety for many of driving assistant systems, e.g., ACC, FCW, Collision Avoidance System (CAS), and Overtaking Assistance System (OAS) since inaccuracies in inter-vehicle distance measurements may lead to severe accidents. The sensors for vehicle distance measurement are categorized into two classes [34]: (a) active depth detection system, e.g., radar, lidar and Time Of Flight (TOF) sensors (b) passive depth detection system, e.g., CCD camera. Active sensors can measure the distance very accurately but they are very expensive and the interference from similar systems is one of the major drawbacks. Passive sensors require more computation time and very sensitive to illumination condition. Stereo camera measures depth based on the disparity and it has some drawbacks; (1) the errors in distance measurements increase in polynomial manner proportional to the distance [35], (2) the accuracy of the disparity can be degraded under the low or high light conditions. Likewise, if the texture feature is too sparse or too dense, the disparity calculation problems are very difficult to solve [36]. Monocular distance measurement typically utilizes prior knowledge [37] and conducts camera calibration [38]. This approach identifies vehicles with already known knowledge such as edges, shapes, shadows. It assumes the position and the direction of camera is fixed and estimates the distance by using pixel distance and calibration parameters. However, when the position or direction of the camera is changed, recalibration is required once again. Due to the inherent prior knowledge, e.g., when shapes are occluded or illumination conditions are changed, the performance of the algorithm can be poor [39].

In this paper, we propose a new method for measuring the distance to the front vehicle based on monocular camera and a line laser. The benefits of the proposed method are as follows: (1) Proposed method is more accurate than disparity-based method. The errors in distance measurement was less than 4m during the entire experiments. (2) It performs well even in night since both laser line and lane markings are visible in the night.

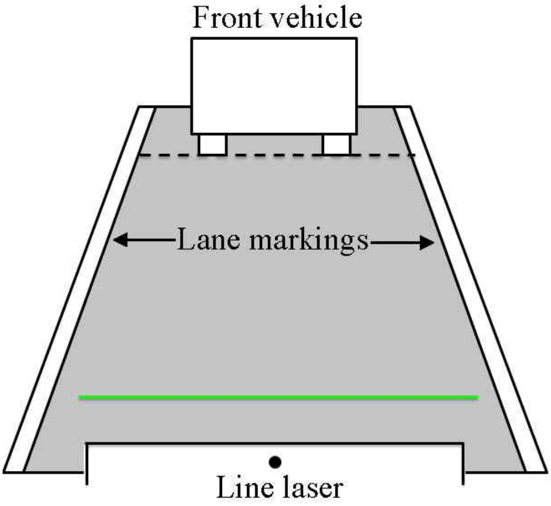
4.2 Proposed Distance Measurement Algorithm

The scheme of the proposed method is as follows: a laser line is emitted in the near front of the vehicle on the road, and a monocular camera captures a sequence of images including the scene of the laser line and the road. We have three assumptions as follows: (1) lane detection system provides the information about the lane width. (2) the width of the laser line illuminated on the road is either measured directly or calculated by using the ratio between the width of the laser line and the lane width that is already known. It is reasonable since, depending on the road type, the width of lane markings follows the traffic regulations in general. The information about the lane width can be provided by geographic information system for transportation

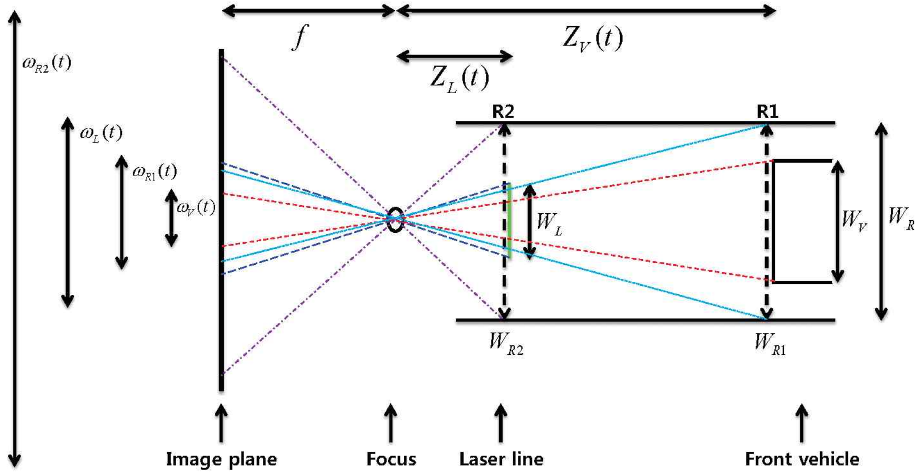
(GIS-T). (3) the lane width on the laser line is the same as the lane width on the rear-end point of leading vehicle. Figure 4.1 (a)-(b) depicts the device set-up and (c) explains the definition of the symbols and imaging geometry.



(a) Device setup (side view)



(b) Device setup (forward-looking view)



(c) The definition of symbols and imaging geometry

Figure 4.1. Schematic diagram of the imaging geometry.

The pinhole camera model gives [40]:

$$\omega_V(t) = \frac{f \cdot W_V}{Z_V(t)}, \qquad \omega_L(t) = \frac{f \cdot W_L}{Z_L(t)} \tag{1}$$

where, W_V and W_L mean the width of the vehicle and the lane, and $\omega_V(t)$ and $\omega_L(t)$ represent the width of the vehicle and the lane in the image plane, respectively. $Z_V(t)$ and $Z_L(t)$ are the distances from the camera to the front vehicle and to the line laser, and finally, f means the focal length. According to the assumption $W_{R1}=W_{R2}=W_R$, we can induce the following equations:

$$\frac{\omega_{R2}(t)}{\omega_L(t)} = \frac{W_R}{W_L}, \qquad \frac{\omega_{R1}(t)}{\omega_V(t)} = \frac{W_R}{W_V} \tag{2}$$

where, $\omega_{R1}(t)$ and $\omega_{R2}(t)$ are the lane width of the point R1 and R2 in the image plane. Following equations come from (2) and (4) is induced from (3):

$$W_R = \frac{W_L \cdot \omega_{R2}(t)}{\omega_L(t)}, \qquad W_V = \frac{W_R \cdot \omega_V(t)}{\omega_{R1}(t)} \tag{3}$$

$$W_V = W_L \cdot \frac{\omega_{R2}(t)\omega_V(t)}{\omega_{R1}(t)\omega_L(t)} \tag{4}$$

Rearranging (1) according to $Z_V(t)$ and substituting (4) in (1), we can get (5) comprised of focal length, width of the line laser, and three measurable variables from the image ($\omega_{R1}(t)$, $\omega_{R2}(t)$ and $\omega_L(t)$) as below:

$$Z_V(t) = \frac{f \cdot W_V}{\omega_V(t)} = f \cdot W_L \frac{\omega_{R2}(t)}{\omega_{R1}(t)\omega_L(t)} \tag{5}$$

To make units consistent in (5), dividing on the right-hand side of equation by pixel size (p) gives $Z_V(t)$ in meters. Finally, for the enhancement of accuracy, by multiplying the calibration parameter k we get (7).

$$Z_V(t) = k \cdot f \cdot W_L \frac{\omega_{R2}(t)}{\omega_{R1}(t)\omega_L(t)} \cdot \frac{1}{p} \tag{6}$$

$$Z_V(t) = \frac{\omega_{R2}(t) \cdot C}{\omega_{R1}(t)\omega_L(t)}, \quad C = \frac{kfW_L}{p} \tag{7}$$

Stereo camera-based distance measurement is shown as below and Figure 4.2 depicts the principle:

$$Z = B \cdot f / d \tag{8}$$

where, B represents the base line, the distance between two monocular cameras, and d means the disparity, the difference between two images

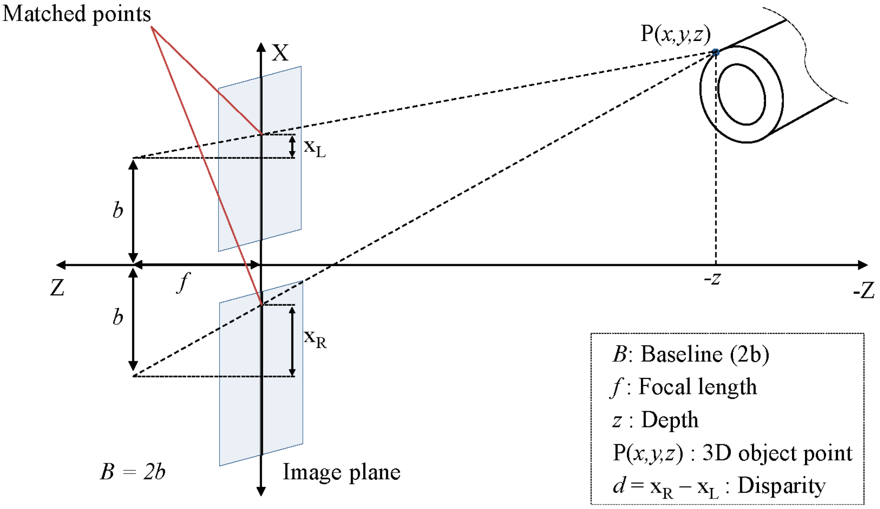


Figure 4.2. Distance calculation based on the disparity map.

in the pixel unit. To get the value of Z in meter unit, the right-hand side of equation should be divided by the pixel size.

4.3 Experiments and Evaluation

4.3.1 Experimental System Setup

For the experiment, we installed a monocular camera and a line laser in the vehicle as well as a stereo camera to evaluate of the performance as shown in Figure 4.3. We regarded the data from a lidar as ground truth, and made all the data synchronized.

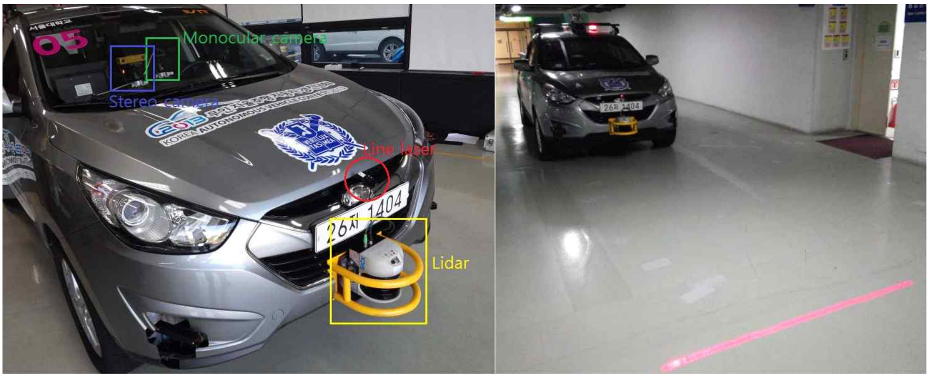


Figure 4.3. Experimental system setup.

The specification of the sensors used in the experiments is shown in Table 4.1.

Sensor	Model	Specification
Monocular camera	FL3-GE-03S2C-C (Camera) DF6HA-1B(Lenz)	648 x 488 (82fps), 7.4µm square pixel, Focal length: 6mm
Line laser1	BLSI65100A-L30	100mW, 30°, red
Line laser2	SEI53100-L50	100mW, 50°, green
Stereo camera1	BB2-08S2C-38	1032 x 776 (20fps), 3.8mm, 65-deg HFOV, 4.65µm square pixel, Baseline: 12cm
Stereo camera3	BB2-08S2C-38	640 x 480 (20fps) Same with stereo camera1
Stereo camera2	VSTC-P250	640 x 480 (20fps), 8mm, 6µm square pixel, Baseline: 26cm
Lidar	LMS511 outdoor	Measurement range : up to 65m, Field of view: 190°

Table 4.1. Sensor specification

4.3.2 Experimental Results

We collected two sets of experimental data for daytime and nighttime in the highway. For nighttime dataset, we used the stereo camera1 and line laser1 while stereo camera2 and line laser 2 were used for daytime dataset. Figure 4.4 is the snapshot of each datasets. From left to right, monocular camera with line laser image, stereo disparity image, the distance image generated by using the data from lidar. We adopted the built-in disparity map provided by the camera.

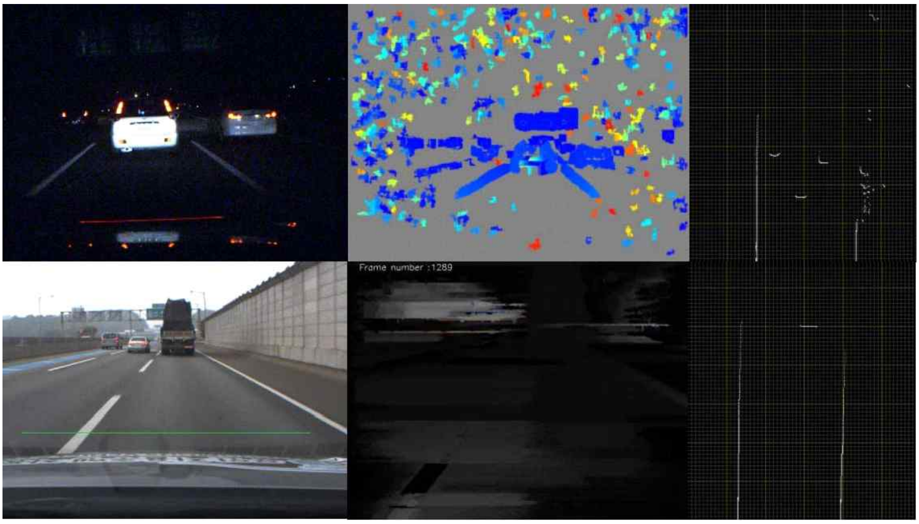
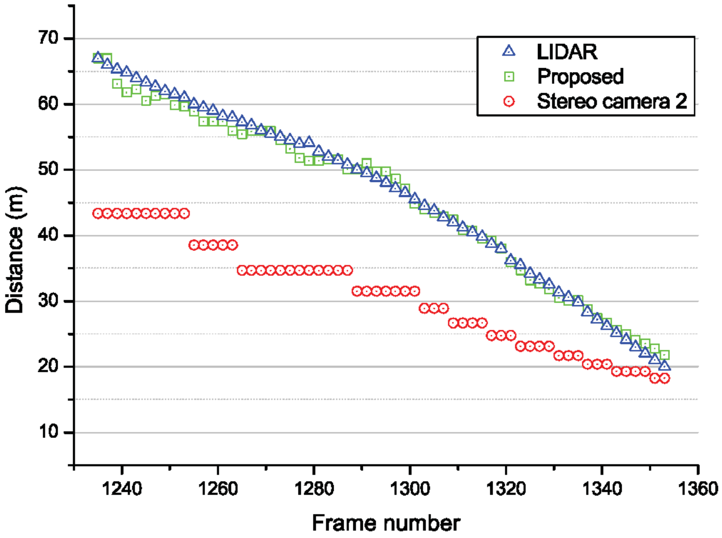
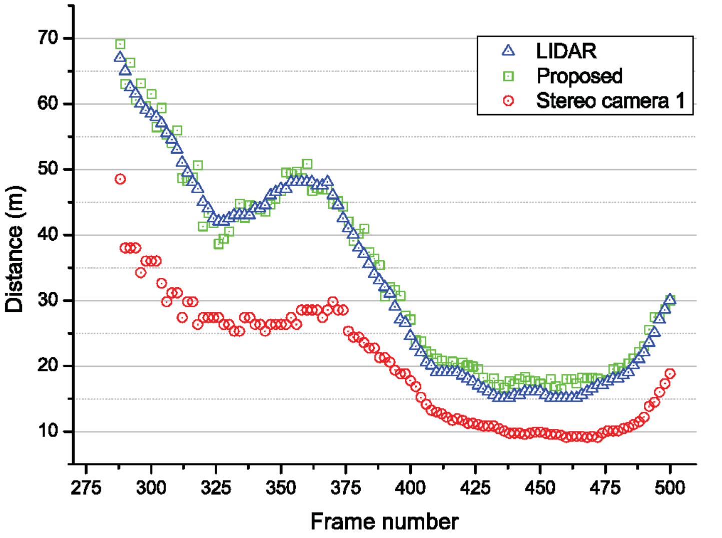


Figure 4.4. The snapshot of the dataset

According to the experimental results as shown in Fig. 5, the proposed method provides higher accuracy as compared to the stereo camera-based method. Figure 4.5 (a) depicts the sequence of distance measurements in daytime highway while Figure 4.5 (b) depicts the nighttime highway situation. The distance was not constantly decreased or increased during the experiments.



(a) Daytime distance measurement



(b) Nighttime distance measurement

Figure 4.5. Distance measurement results

Average distance measurement error of experiments is shown in Table 4.2. The results show that the proposed method is more accurate than stereo camera and the distance measurement error is less about ten times on average. The results in Table 4.2 show that the average error in experiment1 (1.05m) is lower than in experiment2 (1.79m), though the illumination conditions in experiment1 is better than in experiment2. This is due to the fact that the resolution of camera1 used in experiment1 is 1.6 times lesser than camera2 used in experiment2. Since the distance is inversely proportional to the disparity and the number of pixel is limited, lower resolution has more distance error.

	Proposed	Stereo camera
Experiment1 (daytime)	1.05m	15.11m
Experiment2 (nighttime)	1.79m	12.74m
Average	1.42m	13.92m

Table 4.2. Average error in distance measurement

The error in distance measurement-to-distance relationship is shown in Figure 4.6. For comparison study, we adopted the theoretical distance measurement error from the datasheets provided by the manufacturer. Stereo camera3 has the same specification as stereo camera1 except the resolution. But the resolution of stereo camera1 is same as monocular camera (Refer Table 4.1). Comparison results show that the distance measurement error of the proposed method is lower than the other methods. In case of lidar, the error is ±2.5cm in 1-10m range, ±3.5cm in 10-20m range and ±5cm in 20-30m range.

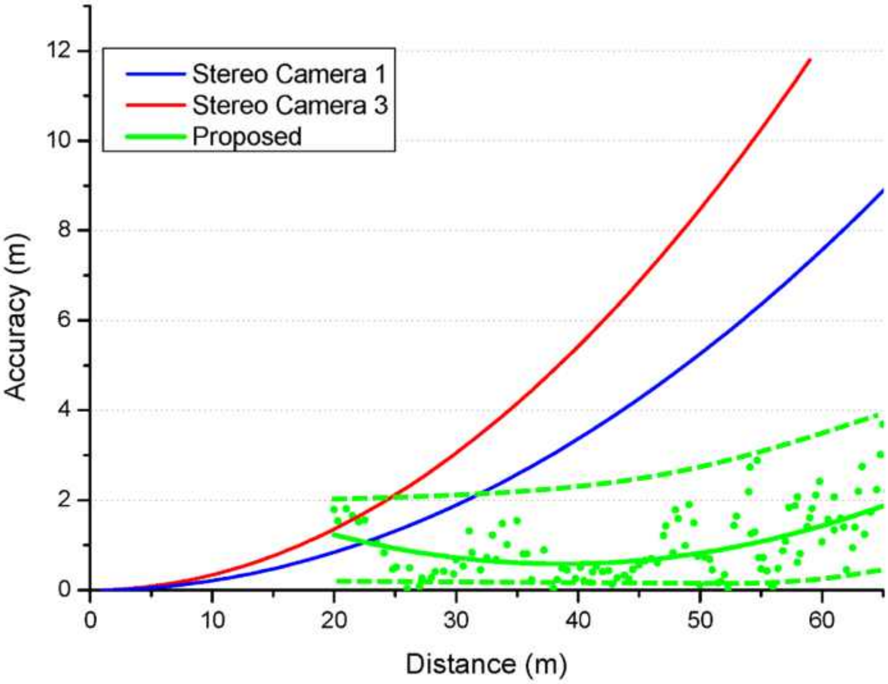


Figure 4.6. The comparison results of the error in distance measurement as compared to the lidar for the distance variations.

4.4 Summary

In this paper, we proposed a new method for the longitudinal inter-vehicle distance measurement by using monocular camera and line laser. Experimental results show that the proposed method exhibits high accuracy as compared to the stereo camera-based method. The benefits of the proposed method are in summary: (1) the proposed method still shows high accuracy in nighttime as compared to the stereo camera-based method which suffer from low illumination conditions. Since the proposed method utilized a line laser and lane markings that are visible well in night. (2) The distance measurement error of the proposed method is lesser in several meters while the stereo camera-based method increases with distance proportionally to the square of the distance.

Chapter 5

Conclusion

In this dissertation, I focused on several important issues for the monocular image recognition in autonomous environment. In Chapter 2, I proposed the illumination-tolerant lane marking extraction filter and effective false positive cancelling algorithm. Experimental results show that the proposed algorithm outperforms other algorithms in more illuminated scenarios. In terms of execution time, the proposed filter achieves over 30fps, and the fast RITR-CH algorithm performs over 40fps. In chapter 3, I proposed the fusing approach between road detection and lane position estimation. The proposed approach is able to provide the new information that is the total number of lanes on the road and the current position of lane. This localization information is very helpful for autonomous driving. In Chapter 4, I proposed a new method for measuring inter-vehicle distance by using a monocular camera and a line laser. The experimental results show that the accuracy of the proposed method very high and outperforms the disparity-based method. I wish above researches would be useful to someone who works for autonomous driving.

Bibliography

[1] W. P. Een, N. Jindapetch, L. Kuburat, and N. Suvanvorn, “A Study of the Edge Detection for Road Lane,” *Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI)*, pp. 995-998, 2012.

[2] C. QIU, “An Edge Detection Method of Lane lines Based on Mathematical Morphology and MATLAB”, *International IEEE conference on Cross Strait Quad-Regional Radio Science and Wireless Technology Conference*, pp. 1266-1269, July 2011.

[3] C. Y. Chang, and C. H. Lin, “An Efficient Method for Lane-Mark Extraction in Complex Conditions,” *International Conference on Ubiquitous Intelligence and Computing and Autonomic and Trusted Computing*, pp. 330-336, 2012.

[4] C. F. Wu, C. J. Lin, and C. Y. Lee, “Applying a Functional Neurofuzzy Network to Real-Time Lane Detection and Front-Vehicle Distance Measurement,” *IEEE Transactions on Systems, Man, and CyberneticsPart C: Applications and Reviews*, vol. 42, No. 4, pp. 577-589, July 2012.

[5] X. Liu, G. Wang, and J. Liao, B. Li, Q. He and Max Q. H. Meng, “Detection of Geometric Shape for Traffic Lane and Mark,” *IEEE International Conference on Information and Automation*, Shenyang, China, pp. 395-399, June 2012.

[6] R. K. Satzoda, S. Suchitra and T. Srikanthan, “Robust Extraction of Lane Markings Using Gradient Angle Histograms and Directional Signed Edges,” *Intelligent Vehicles Symposium*, Alcal de Henares, Spain, pp.754-759, June 2012.

[7] Y. Chen, M. He, and Y. Zhang, “Robust Lane Detection Based on Gradient Direction,”, *IEEE Conference on Industrial Electronics and Applications*, pp.1547-1552, 2011.

[8] H. Deusch, J. Wiest, S. Reuter, M. Szczot, M. Konrad and K. Dietmayer, “A Random Finite Set Approach to Multiple Lane Detection,” *International IEEE Conference on Intelligent Transportation Systems*, pp. 270-275, September 2012.

[9] A. Borkar, M. Hayes and M. T. Smith, “A Novel Lane Detection System With Efficient Ground Truth Generation,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 1, pp. 365-374, March 2012.

[10] J. Li, X. An and H. He, “Lane Detection Based on Visual Attention,”, *International IEEE Conference on Image and Graphics*, pp.570-575, 2011.

[11] E. Shang, J. Li, X. An and H. He, “Lane Detection using Steerable Filters and FPGA-based Implementation,”, *International IEEE Conference on Image and Graphics*, pp. 908-913, 2011.

[12] E. Pollard, D. Gruyer, J. P. Tarel, S. S. Ieng, A. Cord, “Lane Marking Extraction with Combination Strategy and Comparative Evaluation on Synthetic and Camera Images,” *International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA., pp. 1741-1746, October 2011.

[13] R. Danescu, S. Nedevschi, “New Results in Stereovision Based Lane Tracking,”, *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, pp. 230-235, June 2011.

[14] A. Linarth and E. Angelopoulou, “On Feature Templates for Particle Filter Based Lane Detection,” *International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA., pp. 1721-1726, October 2011.

[15] X. Liu, B. Dai, J. Song, H. He, and B. Zhang, “Real-time Long-range Lane Detection and Tracking for Intelligent Vehicle,” *International IEEE Conference on Image and Graphics*, pp. 654-659, 2011.

[16] V. Popescu, R. Danescu, and S. Nedevschi, “On-Road Position Estimation by Probabilistic Integration of Visual Cues,” *International IEEE Conference on Intelligent Vehicles Symposium (IV)*, Madrid, Spain, pp. 583-589, June 2012.

[17] T. Y. Sun, S. J. Tsai and V. Chan, “HSI Color Model Based Lane-Marking Detection,”, *International IEEE Conference on Intelligent Transportation Systems*, Toronto, Canada, pp. 1168-1172, September 2006

[18] T. Veit, J.P. Tarel, P.N. Nicolle, and P. Charbonnier, “Evaluation of Road Marking Feature Extraction,” *IEEE Conference on Intelligent Transportation Systems*, pp. 174-181, Oct. 2008.

[19] OpenCV (2003): Intel Open Source Computer Vision Library http://www.intel.com/research/mrl/research/ opencv/

[20] S. Suzuki and K. Abe, (1985) “Topological structural analysis of digital binary image by border following,” *Computer Vision, Graphics, and Image Processing*, Volume 30, Issue 1, pp. 32-46, April 1985.

[21] D. Aubert, K. Kluge, and C, “Autonomous navigation of structured city roads,” *in SPIE Mobile Robots*, 1990.

[22] J. M. Alvarez, and A. M. Lopez. "Road detection based on illuminant invariance." *Intelligent Transportation Systems, IEEE Transactions on* 12.1, pp. 184-193, 2011.

[23] J. M. Alvarez, M. Salzmann, and N. Barnes. "Learning Appearance Models for Road Detection," IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, June 23-26, 2013.

[24] R. Labayrade, D. Aubert, and J-P. Tarel. "Real time obstacle detection in stereovision on non flat road geometry through." *Intelligent Vehicle Symposium, IEEE*. vol. 2, 2002.

[25] S. Thrun, "Learning occupancy grid maps with forward sensor models." *Autonomous robots* 15.2, pp. 111-127, 2003.

[26] H. Badino, U. Franke, and D. Pfeiffer. "The stixel world-a compact medium level representation of the 3d-world." *Pattern Recognition*. Springer Berlin Heidelberg, pp. 51-60, 2009.

[27] B. Wang, and V. Frémont. "Fast road detection from color images." *Intelligent Vehicles Symposium (IV), IEEE*, pp. 1209-1214, 2013.

[28] G. B. Vitor, D. A. Lima1, A. C. Victorino, and J. V. Ferreira "A 2D/3D Vision Based Approach Applied to Road Detection in Urban Environments." *Intelligent Vehicles Symposium (IV), IEEE*, pp. 952-957, 2013.

[29] A. Asundi, and Z. Wensen. "Fast phase-unwrapping algorithm based on a gray-scale mask and flood fill." *Applied optics* 37.23, pp. 5416-5420, 1998.

[30] S.N. Kang, J.H. Hur, M.E. Choi, and S.W. Seo “Illumination-tolerant lane marker extraction for multi-lane detection.”, IDIS 2013.

[31] R. Hartley, and A. Zisserman. Multiple view geometry in computer vision, vol. 2. Cambridge, 2000.

[32] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes." *Pattern recognition* 13.2 pp. 111-122, 1981.

[33] A. Criminisi, I. Reid, and A. Zisserman. "A plane measuring device." *Image and Vision Computing* 17.8. pp. 625-634, 1999.

[34] D.O. Cualain, et al, "Distance detection systems for the automotive environment: a review." *Irish Signals and Systems Conf*, Sep. 2007.

[35] D. Gallup, et al, "Variable baseline/resolution stereo." *Computer Vision and Pattern Recognition,* 2008, pp 1-8.

[36] N. Jojic, et al, “Detection and Estimation of Pointing Gestures in Dense Disparity Maps,” *Automatic Face and Gesture Recognition*, Mar. 2000, pp. 468-475.

[37] G.P. Stein, M. Ofer, and S. Amnon, "Vision-based ACC with a single camera: bounds on range and range rate accuracy." *Intelligent vehicles symposium*, 2003, pp.120-125.

[38] L. Zhiying, C. Han, and L. Lin, "Research on Automobile Distance Measurement System on Highway Based on Vehicle-mounted Camera." *Computer Application and System Modeling*, 2012, pp1060-1064.

[39] S. B. Gokturk, H. Yalcin, and C. Bamji, “A time-of-flight depth sensor – system description, issues and solutions”. *IEEE Computer Vision and Pattern Recognition*, 2004.

[40] D. Muller, et al, "Time to contact estimation using interest points." *Intelligent Transportation Systems,* 2009, pp.1-6.

국문 초록

“호모 파베르(Homo Faber),” 도구를 사용하는 인간을 지칭하는 말이다. 태초로부터 인간은 편리함을 추구하기 위해 도구를 만들어 왔다. 빠르게 이동하고 싶은 인간의 욕구는 말을 타고, 수레를 만들고, 자동차를 만들게 했다. 자동차는 사람을 빠르고 편리하게 이동할 수 있도록 해 주었지만, 사람은 완벽하지 못한 존재이기에 그 동안 교통사고로 많은 사람들이 죽어갔고, 지금 이 순간도 죽어가고 있다. 지능형 자동차는 인간의 안전 욕구를 충족시키기 위한 최선의 대안으로서 연구되어지기 시작했고, 머지않은 미래에 무인 자동차의 꿈은 실현되어 질 것이 분명하다.

무인 자동차를 현실화하기 위해 다양한 분야의 기술이 요구되지만 그 중 차량 주변 환경에 대한 인식은 가장 기본적이고 중요한 문제 중의 하나임에 틀림없다. 주변에 무엇이 어디에 존재하는지를 알기 위해 다양한 센서들을 활용할 수 있지만, 그 중 가장 많은 정보를 수집할 수 있고, 다양한 용도로 활용할 수 있으면서도 비싸지 않아 다양한 차종에 활용될 수 있는 단안 카메라를 이용한 인식방법에 대한 연구가 실생활에 가장 유용하게 활용될 것으로 기대한다.

본 논문집에서는 차량 환경에서 단안카메라를 이용해 자율 주행에 반드시 필요한 다음에 소개 될 몇 가지 인식방법에 대한 연구결과를 다루었다. 첫 번째 문제는, 차량이 도로를 주행하기 위해서 차선을 반드시 지켜야 하는데 차선을 다양한 조명환경 변화 속에서 탐지하는 것은 어려운 일이지만 반드시 해결해야 할 문제이다. 첫 번째 연구주제는 바로 시간과 날씨, 장소에 따라 다양하게 변화하는 조명환경 속에서도 안정적으로 차선을 탐지할 수 있는 차선 마킹 탐지 기법에 대한 연구이다. 차선 마킹이 온전치 않더라도 탐지해 낼 수 있는 차선 마킹 탐지 전용 필터와 허위 탐지 신호를 제거하는 방법을 고안하여 문제를 해결하였다. 두 번째 문제는, 만약 차선이 없는 도로에서는 어떻게

할 것인가에 대한 고민과 도로 위 많은 차선 중에 지금 몇 번째 차선으로 달리고 있는지를 알아야 잠시 후 어떤 차선으로 달려야 하는지를 결정할 수 있게 되는데, 이 문제를 해결하기 위해 수행한 연구이며 문제 해결을 위해 도로 탐지와 차선 위치 추정 방법을 융합하여 해결하였다. 마지막 문제는, 주행 중 전방 차량과 일정거리를 유지해야 안전하게 주행할 수 있고, 거리에 따라 안전장치들이 적절하게 작동되어야 더욱 안전한 주행이 가능한데 이때 필요한 거리정보를 좀 더 정확하고 정밀하게 측정할 수 없을지에 대한 고민을 해결하기 위해 수행한 연구이며, 문제 해결을 위해 차량 전방에 직선 모양의 레이저를 비추어 거리를 측정하는 방법을 고안하여 문제를 해결하였다.

이렇듯 무인 자동차 구현을 위해 해결해야 할 다양한 문제가 존재하고 이 문제들을 해결하기 위해 단안 카메라를 활용한 적절한 해결방법을 제시하였다. 본 연구결과를 포함하여 향후 더욱 다양한 분야에서 활발한 연구가 진행되어 사고 없이 편안하고 빠르게 이동할 수 있는 시대가 다가오길 기대한다.

주요어 : 차선 마킹 탐지, 도로 탐지, 차선 위치 예측, 차량간 거리 탐지

학 번 : 2010-30798

감사의 글

뭐라고 운을 떼야할지 모르겠습니다. 지난 4년간 졸업을 위해 뛰어왔던 시간들을 되뇌어 보지만 한마디로 정의하긴 어려울 것 같습니다. 지금 이 순간 감사의 글을 쓸 수 있는 것만으로도 크나큰 축복이 아닐 수 없습니다. 군인이기 때문에 라는 불편한 수식어가 제게 붙지 않도록 많은 애를 썼던 것 같습니다. 제가 박사 학위 기간 중에 연구했던 많은 것들이 그 동안 군 생활하며 해오던 것과는 분명 많이 달랐고, 대부분의 것들이 순간순간 제게는 큰 도전이었습니다. 하지만 지금 다 지나고 보니 잘 해냈던 것 같습니다. 군인정신만으로 해낼 수 있었다고 생각하진 않습니다. 정말 행운인 것은 제 곁에 많은 좋은 분들이 있었다는 것입니다. 제가 학교에 처음 왔을 때 그저 누구보다 열심히 잘 하겠노라고 말씀드렸던 것으로 저를 뽑아 주셨던 교수님께 감사드리고, 그 동안 많이 부족했던 저를 수없이 많은 가르침으로 바로 세워주시고, 또 제게 큰 믿음을 주셔서 더욱 열정적으로 연구할 수 있게 많은 기회를 주셨던 것에 대해 진심으로 감사드립니다. 더 멋진 삶으로서 보답할 수 있었으면 좋겠습니다. 더불어, 부족한 제 논문을 지도 편달해주신 최진영 교수님, 조남익 교수님, 김현진 교수님, 김창수 교수님께 감사의 말씀을 드립니다. 지도교수님의 명성에 누가되지 않도록 앞으로 연구할 수 있는 곳에 가게 되면 더 열심히 연구에 매진하겠습니다.

함께 땀 흘리며, 머리 맞대고 연구했던 먼저 졸업하신 선배님들과 멋진 후배님들께도 감사의 말씀을 드립니다. 정한유 교수님을 볼 때면 "어찌 그리 사람 됨됨이가 바른지 모르겠다"는 느낌이 항상 들었습니다. 함께 연구해 볼 수 있는 기회가 있었으면 좋겠다는 생각이 들었지만 연구 분야가 달라 기회가 없었던게 조금 아쉽습니다. 유지훈 교수님, 최윤호 교수님 연구실 오가며 어려운 일 없는지 여쭤봐 주시고 걱정해 주셔서 감사드립니다. 좋은 연구 하

셔서 더 멋진 교수님 되시기를 기대하겠습니다. 박민호 교수님, 하필 제가 다리를 다치는 바람에 함께 자전거 트레킹을 못 한게 정말 아쉽습니다. 국토 종단을 하자셔도 함께 할 자신있었는데, 다음에 다시 한번 기회가 있었으면 좋겠습니다. 교수님과 학교 주도로 자전거 시합 한 번 해야 하는데 결국 못하고 졸업하네요. 김성우 박사님, 함께 연구실 생활하며 많이 보고 배웠고, 또 많이 닮고 싶었습니다. 함께 연구했던 기간이 너무 짧아 많이 아쉽기는 하지만 어느 멋진 곳에서 다시 멋진 모습으로 뵐 것으로 기대하고 있습니다. 종호형, 항상 뭘 물어봐도 성심껏 답변해 주시고, 연구 장소가 달라 많은 얘길 나누진 못 했지만 예전에 말씀하셨던 그 상무님처럼 승승장구하시길 바랄께요. 문영아, 연구할 때 톡톡 튀는 아이디어로 좋은 의견 내고 하는 모습 너무 보기 좋았었다. 회사 잘 다니고, 결혼식 때 하필 교육 결과보고일자랑 겹쳐서 못 가서 미안해. 나도 너처럼 전방 십자 인대가 파열 될 줄은 미처 몰랐다. 승호야, 연구실에 필요한 바른 소리, 쓴 소리 가장 많이 했던, 그 역할을 지금 해주는 사람이 없는게 아쉽구나. 네가 했던 좋은 연구들 잘 이어나가서 좋은 곳에서 승승장구 하길 바란다. 영훈아, 참 많은 모습을 갖고 있어서 많이 놀랐고, 재밌었다. 주변에 함께 있었던 사람들에게 좋은 영향을 더 많이 주길 바라고, 더 많은 것들 해보면서 나도 너처럼 자기 일 잘 하면서 좀 더 신나게 살았으면 좋겠다. 동현아, 연구실 놀러 온다더니 졸업하고 많이 못 봐서 아쉽고 딱 부러지는 성격이니 어딜가서든 잘 하리라 믿는다. 푸니탄 형, 제가 형을 도울 수 있는 시기가 왔는데 형도 취직하고 저도 해군 복위하게 돼서 정말 아쉬워요. 친 동생처럼 잘 챙겨주고, 살갑게 대해줘서 참 감사해요. 몸 건강하시고 형수님과 행복하시고, 인도에서의 꿈 꼭 이루시길 빌어요. 준식이형, 일하면서 공부하는 건 정말 어려운 일인데, 시간을 내어 병행하시는 걸 보면서 대단하다는 생각을 했어요. 힘드시겠지만 끝까지 마무리 잘 하시길 바랍니다. 은동아, 덕분에 이동물체탐지하는 연구를 함께 할 수 있어서 참 좋았다. 어려울 때

항상 함께 얘기할 수 있는 벗을 함께 둔다는게 참 큰 힘이 되지. 이제 물고를 텄으니 나머지 기간 마무리 잘 하길 빌께. 믿음아, 연구실에 큰 형처럼 후배들 실질적으로 잘 챙기고, 연구성과도 가장 좋고. 니가 후배들 잘 끝까지 이끌어 주길 바란다. 고민하던 것들 너무 염려치 말고, 막상 닥치면 별거 아닌거 너도 알잖니? 기풍아, 지금쯤 훈련 받으며 추위와 배고픔에 시달리고 있겠지. 훈련소 생활, 마치고 보면 그리 힘들지 않더라. 복귀해서 결혼생활, 연구 잘 해서 좋은 성과 내고, 후배들이 너를 잘 따르니 후배들 잘 챙겨서 더 멋진 선배로 기억되길 바란다. 명옥아, 너랑 함께 입학해서 가장 많이 나와 같이 고생하고 그랬는데, 막상 졸업하고 떠나려니 아쉽기만 하다. 더 할 수 있을 때 열심히 하는게 맞는 것 같다. 넌 실력도 있지만, 운도 잘 따르니 열심히 하면 큰 성과 낼 거라 믿는다. 나중에 좋은 모습으로 다시 보자. 승탁아, 항상 곁에서 보면 듬직하단 생각이 들었다. 힘들어도 별로 티내지 않고 좋은 방법 찾아서 해결해가는 모습이 보기 좋았다. 연구실 잘 이끌고, 좋은 연구 많이 해라. 인섭아, 나를 형처럼 잘 따라줘서 정도 참 많이 들었는데 떠나려니 아쉽다. 항상 성실히 잘 하는 모습 좋았고, 조금 잘 못 되어도 좋으니 조금 더 적극적으로 추진력 있게 하면 더 좋을 것 같다. 술은 못마셔도 좋으니 술자리는 같이 하면서 이런 저런 맘속 얘길 하는게 너한테 도움이 될거라 본다. 한슬아, 네가 힘든일 도맡아서 해주는 모습 보면서 참 좋았다. 역시 형이라 좀 다르더라, 생각도 깊고. 동생들 아직 어리니 잘 챙겨가며 이끌어 주길 바란다. 연구방향도 잘 잡아서 시행착오 없이 좋은 연구 하길 바란다. 보경아, 같이 비전팀 하면서 내가 시간을 좀 더 내서 얘기도 더 많이 하고 그랬으면 좋았을 텐데 그러지 못 해 미안하다. 박사 과정 하면서도 명옥이랑, 인섭이랑 팀웍 잘 맞춰가면서 연구하면 좋은 성과 있으리라 본다. 준화야, 넌 똑똑해서 참 잘 받아들이고, 이해하고. 적응도 빠르고 잘 하더라. 부디 어딜가서든 지금처럼 열심히 연구해서 좋은 논문 많이 써서 멋진 모습으로 돌아와라. 규민아, 좋

아하는 거 더 잘해서 그 분야 최고가 되라. 지금하는 벤처하면서 많이 느끼고, 항상 현시점에서 다음에 뭐가 필요할지를 생각해서 미리 준비하면 지금보다 더 크게 쓰일거라 믿는다. 두산아, 지금 벤처팀에 쏟아 붓는 열정처럼 나중에 무슨 일을 하게 되는 열정적으로 해라. 상황판단이 빠르고, 명석하니 뭘해도 잘 해낼 거라 믿는다. 나중에 잘 되면 연구실 연구비 걱정 좀 안하게 해주라. 수목아, 난 너 없었으면 졸업 못 할 뻔 했다고 본다. 그 만큼 많이 도와줘서 감사하게 생각하고 너는 배우려고 하는 자세가 되있으니 앞으로 1~2년 안에 크게 될거라 믿는다. 지금처럼 열심히하고 서로 도와가며 재밌게 일해라. 연준아, 기풍이 도와서 지금 하던 것처럼 연구실에 필요한 일 잘 도맡아 해주길 빈다. 믿음이 따라서 최적화를 해도 좋을 것 같고, 기풍이처럼 제어해도 넌 잘 할 수 있을 것 같더라. 연구 열심히 해서 연구실을 빛내주라. 박일형 선배, 선배님 보고 참 많이 배웠어요. 정말 열심히 잘하시는 모습에 저도 더 열심히 할 수 밖에 없었던 것 같습니다. 나중에 실무에서 더 오래 뵐 수 있었으면 좋겠습니다. 토마스, 넌 정말 멋진 녀석이었어. 연구도 잘하고, 연애도 잘하고, 취미활동도 잘하고. 참 균형잡힌 생활을 하는 진정한 능력자가 아니었나 싶다. 정한아, 함께 있을 때 내가 많은 도움을 주진 못 했지만 지금 하고 있는 일 잘 하고 어딜가서든 건승하길 빈다. 지희씨, 정인씨, 수민씨, 그 동안 물심양면으로 참 많이 지원해주고 응원해주셔서 감사드려요. 우리 연구실 연구생들 앞으로도 많이 챙겨주세요.

끝으로, 집에 일찍 들어가지도 못하는 매정한 남편을 이해해주고 응원해준, 보경이, 민주 엄마, 사랑하는 아내 미미에게 감사의 마음을 전합니다. 일생을 두고 받은 사랑만큼 더 사랑하며 살아갈게요. 항상 아들을 응원해주신 우리 부모님, 먼 곳에서 항상 강서방 챙겨주시는 장인 장모님 감사드립니다. 끝까지 효도하며 즐겁게 해드리며 살겠습니다.