Road Area Extraction Based on Vanishing Point Detection and Texture Orientation Estimation

Bui Trung Hieu

Kyushu Institute of Technology
Ph.D. Thesis
March 2013

Acknowledgements

I would like to thank all people I met at Kyushu Institute of Technology (KIT) and those who have made a comfortable environment for me during my graduate time.

First and foremost, I would like to express my heartfelt thanks and sincere gratitude to Professor Eitaku Nobuyama, my dear advisor and mentor, for being ultimately supportive, understanding and helpful during my study at KIT. Professor Nobuyama is one of the kindest, gentlest and smartest people I have ever known, and I found that I was very lucky to work with him. This thesis and all of my research papers would have not been possible without or with less support and guidance from my advisor. His immense knowledge, perpetual enthusiasm for science and research, and thoughtful insights have helped me get on the right track for my research and kept inspiring me. As an excellent scientist with rich experience, he has not only complemented my academic knowledge, but also enlarged my vision and improved my research skills. "independent thought" and "freedom in research" are what he always encourages me to perform, without any constrain in idea. At the same time, he was always there for me to consult and discuss with, or check out my papers even during his weekend or on vacation. I believe that if I leave KIT, I would miss his dedication and kindness so much.

Special thanks are due to Professor T. Ejima, Professor H. Noda, Assoc. Professor S. Enokida, and Assoc. Professor T. Saitoh, for serving on the examination committee. Their valuable comments and insightful suggestions have greatly improved my work.

I would like to thank Assoc. Professor T. Saitoh for his constant supports and useful discussions during my study at KIT.

I also would like to thank Professor K. Tanaka for introducing me to KIT, especially

to my current supervisor. I would have not started my Doctoral course at KIT without his recommendation and supports.

I would like to express my profound appreciation to people in my lovely laboratory who were always there to help me to complete my research, especially Mr. Nakamura and Mr. Hisano who helped me a lot with software setups, translation of paperwork, and chatting in Japanese.

I wish to extend my sincere gratitude to the Shiki Masakazu Asia Scholarship, the 100^{th} Anniversary Memorial Scholarship of KIT, the Fukuoka International Exchange Foundation Scholarship, and the Rotary Yoneyama Memorial Scholarship for their financial support during my study in Japan. Without these important resources, I would not have totally focused on completion of my research.

I would like to thank Mrs. Horio, Mr. and Mrs. Fujita for their constant helps and supports during my living time in Japan. I also would like to thank Doctor Nawata for giving the free medical check-ups to my family for the past three years.

I am greatly indebted to my wife, Huyen. She has sacrificed many weekends due to my research works but even though she relentlessly encouraged me. I am grateful to her as she believed in me, helped me, and supported me.

Finally, I state my greatest pleasure for having such wonderful parents. Their presence in my life is the strongest inspiration behind my every work.

Abstract

Road area extraction plays an important role in different areas of computer vision such as autonomous driving, car collision warning and pedestrian crossing detection. Detecting road areas from a single road image is a challenging problem as the detection algorithm must be able to deal with continuously changing backgrounds, different environment (urban, high ways, off-road), different road types (shape, color), and different imaging conditions (varying illumination, different viewpoints and changing weather conditions).

Over the past few decades, numerous road area extraction methods have been widely published for urban and highway roads, structured roads, and unstructured roads. In all of these studies, estimating a location of vanishing point (VP) is a key requirement. A set of parallel lines in the world space by perspective projection converges to a common point in image space known as the VP. If the VP can be located correctly, then it is more likely to detect the road area properly. State-of-the-art VP-based road detection methods can be mainly grouped into three categories: edge-based methods, prior-based methods, and texture-based methods. Most edge-based methods take advantage of computational efficiency, and they can be applied to real time systems. However, the disadvantage is that they only work well for structured roads with clear painted lines or distinct road boundaries, while they usually fail in unstructured roads lacking sharply defined, smoothly curving edge.

In order to overcome the limitation of these edge-based methods, prior-based methods and texture-based methods for road detection have been widely proposed. Most prior-based methods are robust to varying imaging conditions, road types and surrounding environments. However, they often require a large-scale image or video

training database and also manual works for labeling VPs for the training stage. Hence, such prior-based methods are inapplicable to detect the road region from a single road image. In contrast, texture-based methods are very effective at detecting road areas for both structured roads and unstructured roads. These methods first search for local oriented textures and then make them vote for the location of the roads VP. In order to segment the road area, a VP-constrained group of dominant edges are detected, and two most dominant edges are selected as the road borders. In general, the disadvantages of these texture-based methods are: i) the computational cost of VP detection process is high, and ii) an estimation error of VP detection which may affect the performance of road area extraction is obtained in some images.

In this thesis, our main objectives are: i) to reduce the computational cost and improve the performance of VP detection algorithm, and ii) to propose a new VP-based road area extraction method from a single road image. For this purpose, this thesis has been divided into four chapters.

In Chapter 1, the background and some related works are introduced.

Chapter 2 is in accordance with the first objective mentioned above. In this chapter, our novel texture-based local soft voting method for VP detection is explained. Firstly, Gabor filters and confidence level function are introduced. Gabor filters are used to calculate the texture orientation at every pixel of the road image, and the confidence level function is used to determine the remaining voters which are useful for the VP voting process by checking the reliability of the obtained texture orientations. After that, a novel local soft voting method is proposed, in which the number of scanning pixels is much reduced to reduce the computational cost, and a new VP candidate region is introduced to improve the estimation accuracy. The proposed VP detection method has been implemented and tested on 1000 road images which contain large variations in color, texture, lighting condition and surrounding environment. The experimental results demonstrate that this new voting method is both efficient and effective in detecting the VP from a single road image and requires much less computational time when compared to a previous voting method.

Chapter 3 is in accordance with the second objective stated above. In this chapter,

our new VP-constrained road area extraction method is described. The goal of this method is to detect the most immediate straight road part in the direction of the optical axis of the forwarding looking cameras based on estimating two lines (road boundaries) going from the VP and below the VP in the road image. In this method, in order to achieve a one degree level of precision for road boundaries detection, a histogram of 180 angles corresponding to angles of 180 lines going from the detected VP is used. When generating the histogram, the color information of the road image is combined to improve the estimation performance. The proposed road area extraction method has been implemented and tested on 1000 road images which are same as ones used in Chapter 2. The experimental results show that our proposed method performs well in challenging conditions.

Finally, Chapter 4 presents our conclusions and future work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In recent years, one of the most important innovations in transportation has been the application of advanced sensor, computer, electronics and communications technologies to the operation of the transportation system. These applications, known as intelligent transportation systems (ITS), are primarily intended to improve both the safety and efficiency of travel flows on the transportation system.

The type of ITS applications that are found in many parts of the world including regional multimodal traveler information systems, navigation positional aids to motor vehicles and ships, coordinated traffic control systems, transit management systems, and electronic toll/fare and freight tagging systems. A major benefit of ITS applications is that information on the current performance of the transportation system can be used to inform system users of bottlenecks and alternate routes or of model option. ITS technologies can be used as well by the operators of modal networks to respond to incidents or other events needing attention.

Road area extraction (see some examples in Fig. 1.1) is one of the major research topics which contributes to safety driving assistance systems (SDAS) and autonomous driving systems (ADS) which are two of the most important parts of ITS. The aim of road area extraction is to give information to on-board intelligent computing equip-

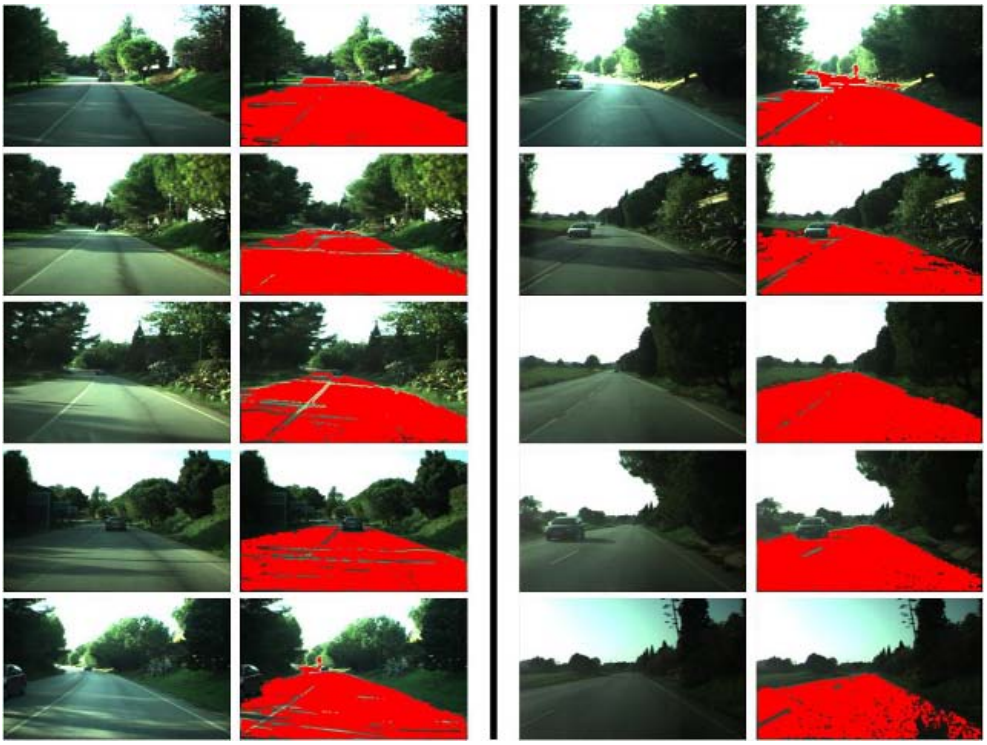
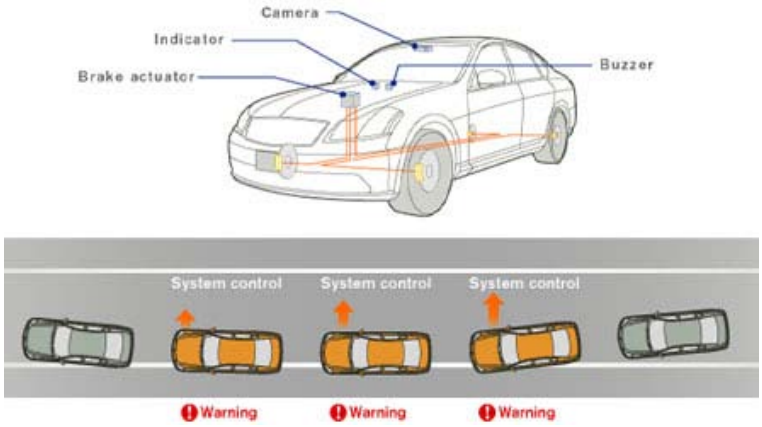


Figure 1.1: Examples of road area extraction (red regions) by Alvarez et al. [1].

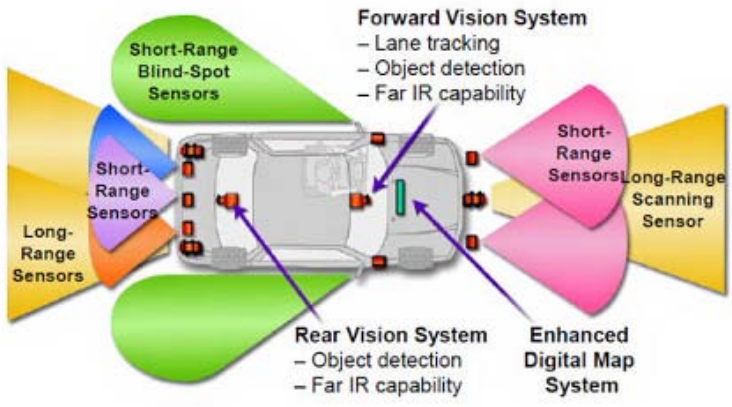
ments as a major knowledge of driving environment.

SDAS support drivers with particular aspects of driving by providing information. They can assist in critical situations with an intervention to carry out the driver's commands more securely, while at the same time not relieving the driver of his responsibility. If necessary, they will take emergency measures and help the driver to stay in control. Figure 1.2 (a) shows an example of lane departure warning system [4] which is one important component of SDAS. The lane departure warning system checks that the vehicle is in its lane by examining the lane markings on the road. If the driver changes lanes without using the turn indicator, an alert signal warns the driver.

ADS sense the world with such techniques as radar, lidar, GPS and computer vision. Advanced control systems interpret the information to identify appropriate navigation paths, as well as obstacles and relevant signage. Autonomous vehicles typically update their maps based on sensory input, such that they can navigate through uncharted environments. Figure 1.2 (b) shows an example of autonomous



(a) Lane departure warning system.



(b) Autonomous car.

Figure 1.2: Examples of ITS.

car [5] which includes a lane road detection system.

Generally, a road image can be classified into a structured (e.g., a road in urban areas) or an unstructured one (e.g., a road in rural area or desert). For structured roads (see Fig. 1.3), the localization of road borders or road markings is one of the most commonly used approach. Color cue [6], [7], [8], Hough transform [9], [10], steerable filters [11], [12], Spline Model [13], [14] etc. have been utilized to find the road boundaries or markings. The limitation of these methods is that they only consistently work for structured roads which have noticeable markings or borders. Methods based on segmenting the road using the color cue have also been proposed, but they do not work well for general road image, especially when the road have little difference

Figure 1.3: Examples of structured road with clear painted lines.

in colors between their surface and the environment. In addition, laser [15], radar [16], and stereovision [17] have also been used for structured-road detection.

For unstructured roads or structured roads without remarkable boundaries and markings (see Fig. 1.4), Alon et al. [18] have combined the Adaboost-based region segmentation and the boundary detection constrained by geometric projection to find the drivable road area. However, it needs many different types of roads images to train a region classifier, which might be onerous. Stereo cameras [19] are also used to determine terrain traversability. When there is little different in color between the road and off-road areas, it is hard to find strong intensity change to delimit them. The one characteristic that seems to define the road in such situation is texture.

The previous approaches [2], [3], [20], [21], [22] have attempted to define the forward drivable image region by utilizing the texture cue. They compute the texture orientation for each pixel, then seek the vanishing point of the road by a voting scheme. In order to segment the road area, a VP-constrained group of dominant edges are fre-

Figure 1.4: Examples of unstructured road.

quently detected, and two most dominant edges are selected as the road borders. Our road area extraction method proposed in this thesis belongs to this line of research.

1.2 Problems statement and objectives

As stated in the previous section, numerous image-based road detection algorithms have emerged as one of the components of ITS. Most of early systems focused on following the well-paved road that is readily separated from its surrounding. More recently, many algorithms have been proposed to handle off-road conditions. In particular, the previous approaches [2], [3], [20], [21], [22] have attempted to define the forward drivable image region by utilizing the texture cue. These methods first search for local oriented textures and then make them vote for the location of the road's VP. In order to segment the road area, a VP-constrained group of dominant edges are frequently detected, and two most dominant edges are selected as the road borders.

In general, the disadvantages of the above mentioned texture-based methods are:

i) the computational cost of VP detection process is high, and ii) an estimation error of VP detection which may affect the performance of road area extraction is obtained in some images.

In this thesis, our main objectives are: i) to reduce the computational cost and improve the performance of VP detection algorithm, and ii) to propose a new VP-based road area extraction method from a single road image. In order to reduce the computational cost of VP detection algorithm, the number of scanning pixels is much reduced. On the other hand, in order to improve the estimation performance of VP detection algorithm, a new VP candidate region and a new soft voting function are proposed. The road area can be detected using the histograms. These histograms are generated based on the detected VP and texture orientations estimation.

1.3 Outline of the thesis

The rest of this thesis is organized as follow.

Chapter 2 is in accordance with the first objective mentioned in the previous section (reducing the computational cost and improving the performance of VP detection algorithm). In Chapter 2, our novel texture-based local soft voting method for VP detection is explained. In this chapter, background and problems statement are described in Section **2.1**. After that, some related researches is reviewed in Section **2.2**. In Section **2.3**, Gabor filters and confidence level function are introduced. In this section, Gabor filters with 5 scales and 36 orientations are used to calculate the texture orientation at every pixel of the road image, and the confidence level function is used to determine the remaining voters which are useful for the VP voting process by checking the reliability of the obtained texture orientations. A larger threshold value for confidence level function is introduced and a new algorithm is proposed in order to reduce the number of remaining voters. After that, our novel local soft voting method for VP detection is proposed in Section **2.4**, in which the number of scanning pixels

is much reduced in order to reduce the computational cost, and a new VP candidate region is introduced in order to improve the estimation accuracy of the VP detection algorithm. Next, the proposed method is summarized in Section **2.5**. The proposed VP detection method has been implemented and tested on 1000 road images which contain large variations in color, texture, lighting condition and surrounding environment. The experimental results are demonstrated to show the effectiveness of the proposed method in Section **2.6**. Section **2.7** presents our conclusions.

Chapter 3 is in accordance with the second objective stated in the last section (proposing a new VP-based road area extraction method from a single road image).In Chapter 3, our new VP-constrained road area extraction method is described. The goal of this method is to detect the most immediate straight road part in the direction of the optical axis of the forwarding looking cameras based on estimating two lines (road boundaries) going from the VP and below the VP in the road image. In this method, in order to achieve a one degree level of precision for road boundaries detection, a histogram of 180 angles corresponding to angels of 180 lines going from the detected VP is used. When generating the histogram, the color information of the road image is combined to improve the estimation performance. The proposed road area extraction method has been implemented and tested on 1000 road images which are same as ones used in Chapter 2. The experimental results show that our proposed method performs well in challenging conditions. In this Chapter, background and problems statement are described in Section **3.1**. After that, some related researches is reviewed in Section **3.2**. The road area extraction method is proposed in Section **3.3**. The experimental results are demonstrated to show the effectiveness of the proposed road area extraction method in Section **3.4**. Section **3.5** presents our conclusions.

Chapter 4 presents our conclusions and future work.

Chapter 2

Vanishing point detection

2.1 Introduction

Using computer vision techniques to detect drivable road areas plays a very important role in navigating autonomous vehicle systems. Recently, numerous interesting lane and road detection algorithms have been widely published for urban and highway roads [8], [23], [24] structured roads [14], [25], [26], [27], [28] and unstructured roads [2], [18], [20], [21]. In all of these studies, estimating the vanishing point (VP) is a key requirement because the correctly obtained VP provides a strong clue to the localization of the road region.

State-of-the-art vision-based VP detection methods can be mainly grouped into three categories: edge-based methods [14], [24], [26], [29], [30], prior-based methods [23], [28] and texture-based methods [2], [3], [20], [21], [31], [32].

Most edge-based methods often include three steps: i) extract edge pixels by an edge detector ii) detect straight lines by a linear transformation and iii) obtain the VP by a voting algorithm. For instance, in [14], edge pixels are extracted by the Canny detector [33], and then straight lines are detected by the Hough transform, finally the intersections of any pair of lines vote for VPs on another Hough space. In general, these edge-based methods can be applied to real-time systems due to their computational

efficiency. However, the disadvantage is that they only work well for structured roads with clear painted lines or distinct borders, while they usually fail in unstructured roads lacking sharply defined, smoothly curving edges.

In order to overcome the limitation of these edge-based methods, prior-based methods and texture-based methods for VP detection have been proposed recently. For instance, the prior-based method proposed in [28] is robust to varying imaging conditions, road types and scenarios by integrating contextual three-dimensional information with low-level cues. This contextual information includes horizon lines estimated by the method in [34], three-dimensional scene layout computed by the method in [35], three-dimensional road geometry inferred by the method in [36], and so on. From the viewpoint of computational cost, the method in [28] is time-consuming cause of integrating several different techniques. On the other hand, the global perspective structure matching method proposed by Wu et al. [23] requires a large-scale image or video training database and also manual works for labeling the VPs for the training stage. Therefore, such prior-based methods are inapplicable to detect the VP from a single road image.

In contrast, texture-based methods for VP detection [2], [20], [21] are very effective for both structured and unstructured roads by utilizing the texture information from a single road image. All of these studies consist of three steps: i) calculate the texture orientations by applying a directional filter bank ii) determine the voters and VP candidates and iii) vote for obtaining the VP by a voting algorithm. For instance, Rasmussen [20], [21], as well as Kong et al. [2] uses the same Gabor wavelet filters introduced in [37] to compute dominant texture orientations before applying voting algorithms to obtain the VP. A global voting method for detecting VPs was proposed by Rasmussen [20], [21]. However, the computational cost of this method is very high due to a large number of scanning voters and VP candidates. Moreover, as pointed out in [2], this global voting tends to favor VPs that are high in the image, leading sometimes to large estimation errors, especially when the true VP is in the lower part

of the road image.

In order to overcome the above problems, Kong et al. [2] proposed an effective VP detection method, in which a "confidence level" function and a local adaptive soft voting (LASV) method were proposed. The confidence function is used to determine the voters by checking the reliability of the obtained texture orientations. Before applying the LASV method, a half-disk voting region is created for each VP candidate, and only voters within this half-disk vote for the VP candidate.

The above mentioned LASV method for VP detection performs well in general road images, especially in unstructured road images (see the images of Fig. 2.1(a)). In addition, this method is more effective and faster than previous texture-based methods [20], [21]. However, the computational cost of the LASV method is still high due to a large number of scanning pixels. Besides, this method yields an estimation error in some images (see the images of Fig. 2.1(b)), in which most voters in the lower part of the image cannot vote for the true VP because the radius of the proposed half-disk voting region is not large enough.

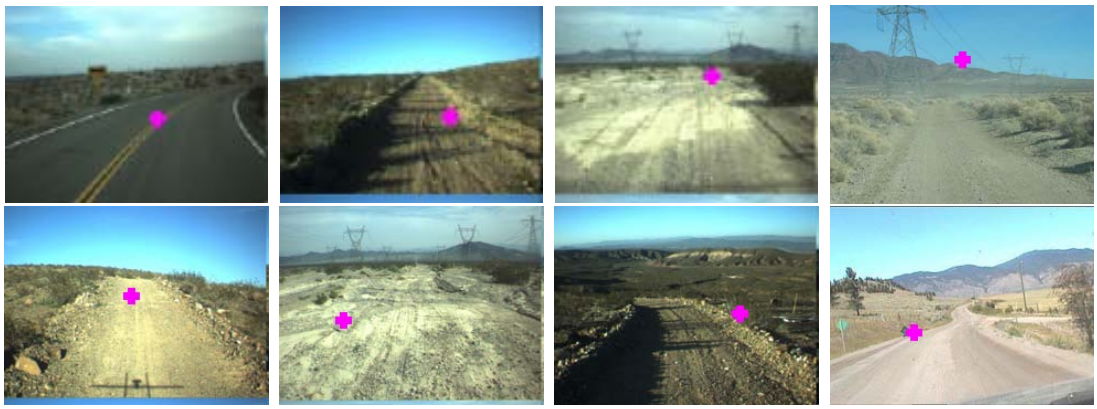
Learning both advantages and disadvantages of current texture-based methods encourages us to propose a new lower-computational-cost and higher-accuracy local soft voting method to detect the VP from a single road image.

The basic concepts of the proposed method are: i) to reduce the number of confidential voters and ii) to scan the voters instead of the VP candidates (note that the number of voters is much lower than the number of VP candidates).

In order to reduce the number of voters, the threshold for the confidence levels is set to be higher than that in the LASV method. In the voting process, a new VP candidate region is defined for each voter, and a new local soft voting function is proposed. Each voter votes for all the pixels in its VP candidate region, with the voting score calculated by a local soft voting function proposed in this thesis.

The proposed method has been implemented and tested on 1000 road images which contain large variations in color, texture, lighting condition and surrounding environ-

(a) Proper detected VPs.



(b) Improper detected VPs.

Figure 2.1: Examples of VP detection (pink crosses) by the LASV method.

ment. The experimental results demonstrate that this new method is both efficient and effective in detecting the VP and requires less computational cost when compared to the LASV method [2].

The remainder of this chapter is organized as follows. Related research is reviewed in Section **2.2**, and the Gabor filters and the confidence level function introduced in [2] are explained in Section **2.3**. A VP candidate region for each voter and a new local soft voting method are proposed in Section **2.4**, and the proposed method is summarized in Section **2.5**. In Section **2.6**, experimental results are demonstrated to show the effectiveness of the proposed method. Section **2.7** presents our conclusions.

2.2 Related research

As stated above, previous texture-based methods [2], [20], [21] have attempted to detect the VP based on texture orientation calculation. In all of these studies, Gabor filters are applied in order to compute the texture orientation at each pixel of a road image. A VP is then detected using a voting algorithm.

A global hard voting method is proposed in [20], [21] as the voting algorithm. In this algorithm, all the pixels of the image can be VP candidates, and the voting region of a VP candidate is defined as the entire image below the VP candidate.

The left-hand figure in Fig. 2.2 shows an example of the voting region, where V is a VP candidate and V_R is the voting region of V. (Note that the gray frame around the image is a region in which the convolution with Gabor filters cannot be calculated. In the present case, the width of the region is eight pixels.) A pixel P in V_R votes for V with a fixed voting score if the angle $\gamma = \angle(\overrightarrow{PV}, \overrightarrow{O_P})$ is below a certain threshold, where the vector $\overrightarrow{O_P}$ denotes the texture orientation at P and γ denotes the angle between the $\overrightarrow{PV}$ and $\overrightarrow{O_P}$ directions. A VP is detected as the pixel having the highest voting score.

The disadvantages of this method are: i) the computational cost is very high because a large voting region is scanned for each VP candidate and ii) improper VPs are detected in several cases because a voting score is fixed irrespective of the distance between P and V.

In order to overcome these disadvantages, a half-disk voting region and a soft voting method, referred to as local adaptive soft voting (LASV), were introduced in [2]. In this previous paper, Gabor filters are also used to compute the texture orientation at every pixel of the road image. Moreover, in [2], a confidence level function is introduced in order to discard the pixels for which the estimated texture orientations are not reliable. If the confidence level exceeds a certain threshold the pixel is kept as a voter; otherwise the pixel is discarded. The remaining pixels are referred to as the remaining voters, which are used to detect a VP. In this method, a VP candidate V is searched

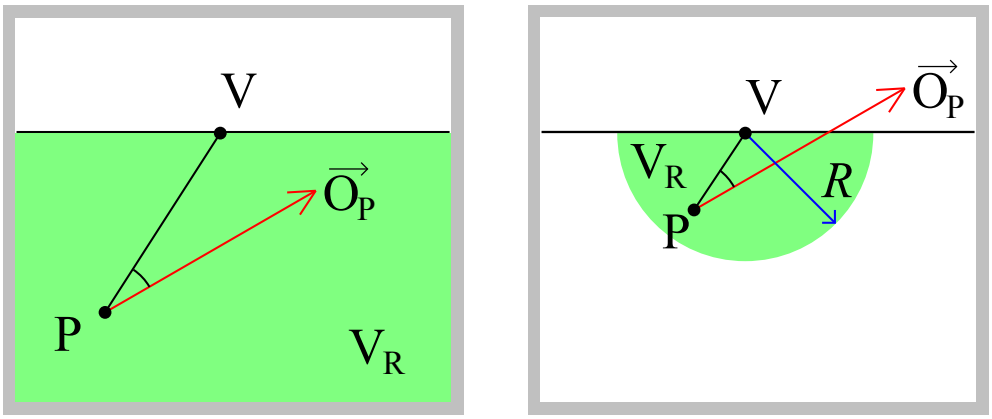


Figure 2.2: Global voting method and LASV method.

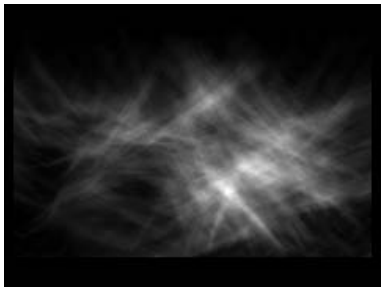
for in the uppermost 90% of pixels of the entire image, and the voting region V_R of V is defined as a half-disk below V centered at V (see the right-hand figure of Fig. 2.2). The radius R of this half-disk is set to $0.35 \times H$, where H is the height of the image. Each remaining voter P inside V_R votes for V with a voting score calculated by a voting function

$$\text{Vote}(\text{P}, \text{V}) = \begin{cases} \frac{1}{1+[\gamma \times d(\text{P},\text{V})]^2} & \text{if } \gamma \leq \frac{5}{1+2d(\text{P},\text{V})} \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where $d(\text{P}, \text{V})$ denotes the distance between P and V divided by the diagonal length of the input image. The pixel having the highest voting score is selected as a VP.

As mentioned in Introduction, the above LASV method is more effective and faster than previous global voting methods [20], [21]. However, the computational cost of the LASV method is still high, and this method yields an estimation error in some images, in which the remaining voters far from the true VP cannot vote for the true VP because R is not large enough.

Our experimental results demonstrate that, although using the half-disk voting region with a larger value of R may improve the estimation performance of the LASV method, it also increases the computational cost of the algorithm. Figure 2.3 shows an example of VP by LASV method with different values of R. It can be seen that



Voting map

VP

(a) VP detection by the LASV method with $R = 0.35 \times H$

Voting map

VP

(b) VP detection by the LASV method with $R = 0.65 \times H$

Figure 2.3: A larger R improves the estimation performance of VP detection.

by using $R := 0.65 \times H$, a better VP detection result is obtained. These experimental results will be described in detail in **2.6.3**.

2.3 Texture orientation and confidential level

The dominant texture orientation $\theta(z)$ at pixel $z = (x, y)$ of an image is the direction that describes the strongest local parallel structure or texture flow. This is of course a scale-dependent measure. Precise estimates of the dominant texture orientations are crucial in order to obtain sharp peaks in the voting objective function and hence accurately localize the VP.

There is a considerable body of work on estimating dominant texture orientations. For example, we may apply a bank of multi-scale, oriented filter such as steerable

filters [11] and analyze the maximum responses. Another approach is to generate a Gaussian pyramid of the image, use principal components analysis on the set of gradients within a small window to obtain a consensus direction at each scale, and then interpolate [38].

Our texture orientation estimation relies on Gabor filters since they have been widely used in a number of previous methods [2], [20], [21].

In this section, the texture orientation calculation method and the confidence level function introduced in [2] are briefly explained.

Gabor filters are used to calculate the texture orientation at each pixel of a road image. For a scale ω and an orientation ϕ, the Gabor filter is defined as follows:

$$\Psi_{\omega,\phi}(x,y) = \frac{\omega}{\sqrt{2\pi}c} e^{-\omega^2(4a^2+b^2)/(8c^2)}(e^{ia\omega} - e^{-c^2/2}), \tag{2.2}$$

where $a = x\cos\phi + y\sin\phi$, $b = -x\sin\phi + y\cos\phi$, and c is a constant. As in [2], we will use the Gabor filters with 36 orientations, five scales, and $c = 2.2$, i.e., ϕ is chosen to be from $0°$ to $175°$ with an angle interval of $5°$, and ω is chosen to be from 1 to 5 with a scale interval of 1.

Figure 2.4 shows the real filters and the imaginary filters of the Gabor filters with 36 orientations and five scales, where each filter consists of 17×17 pixels.

For the gray level value of a road image $I(z)$ at $z = (x, y)$ the convolution of the

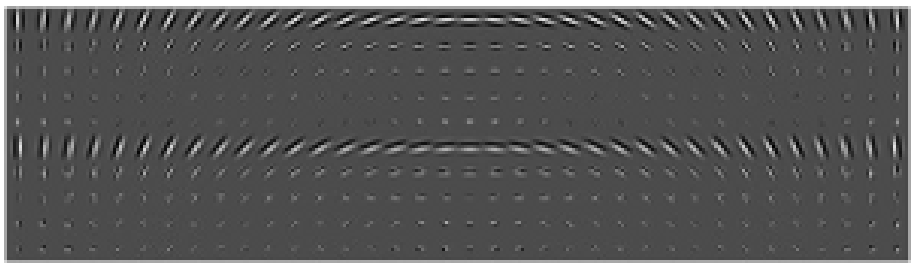


Figure 2.4: Gabor filters with 36 orientations and five scales.

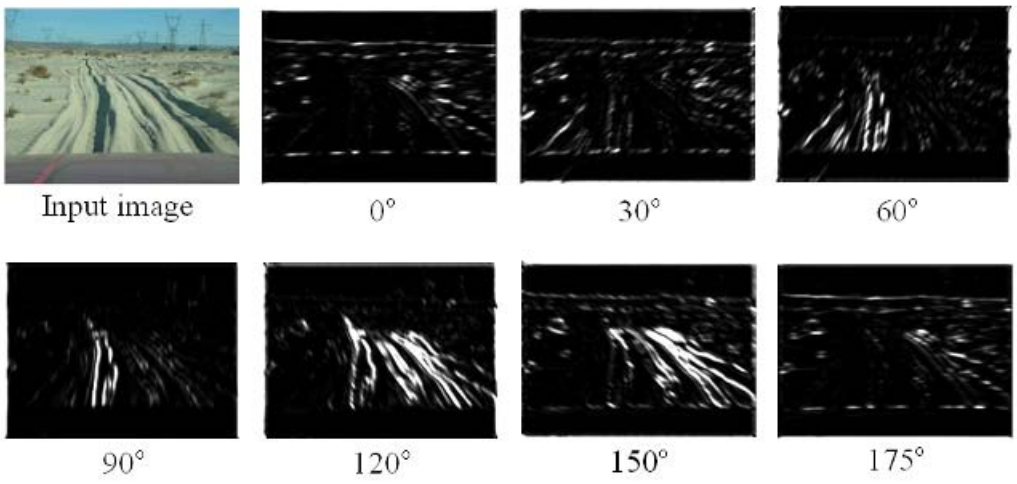


Figure 2.5: Examples of convoluted images by Gabor filters.

image and a Gabor filter is defined by

$$G_{\omega,\phi}(z) = I(z) \otimes \Psi_{\omega,\phi}(z), \tag{2.3}$$

and the response image $R_\phi(z)$ for the orientation ϕ is calculated as the average of the square norm of $G_{\omega,\phi}$ at different scales, as follows:

$$R_\phi(z) = \text{average}_\omega \left\{ (\text{Re}(G_{\omega,\phi}))^2 + (\text{Im}(G_{\omega,\phi}))^2 \right\}. \tag{2.4}$$

Then the texture orientation angle $\theta(z)$ at z is defined in terms of the maximum average response as follows:

$$\theta(z) = \text{argmax}_\phi R_\phi(z). \tag{2.5}$$

Figure 2.5 shows examples of some convoluted images by Gabor filters in different orientations. Figure 2.6 shows an image overlaid with sampled texture orientations estimated using Gabor filters.

Although the above solution for texture orientation estimation has been used by previous researchers, the estimated texture orientation in this way is not guaranteed

to be correct. The author in [2] proposed a confidence level function to provide a confidence level to the texture orientation $\theta(z)$at pixel $z = (x, y)$. They seek to a way which evaluates how peaky the function $\phi \longmapsto R_\phi(z)$ is near the optimum angle $\theta(z)$.

In order to define a confidence level, let $r_1(z) > \cdots > r_{36}(z)$ be the ordered values of the Gabor response for the 36 considered orientations (in particular, $r_1(z) = R_{\theta(z)}(z)$). If the global maximum response is significant different from the other local maximum responses, the texture orientation estimation is reliable, otherwise, it is not.

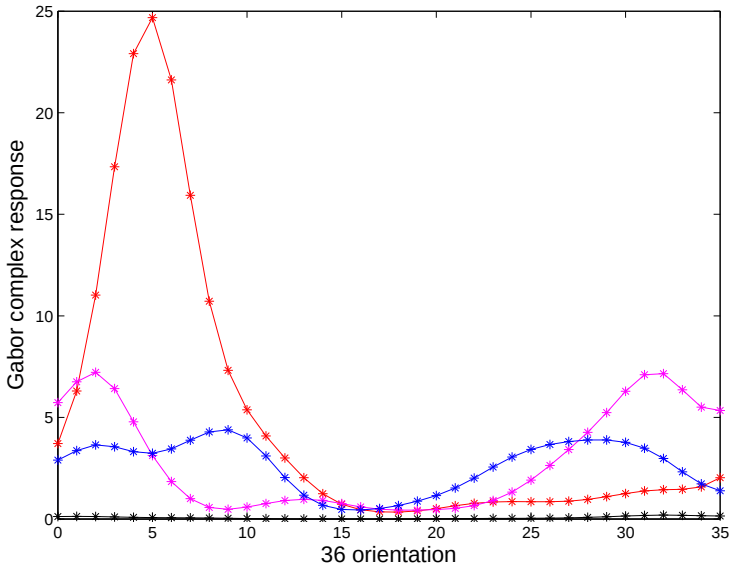
Figure 2.7 (a) shows four points on which the Gabor complex responses are evaluated. In the figure, the red point is a pixel in a texture of the road. The blue and pink points are outside the road, and the black point is in the sky. Among these four points, only the red points are needed for the VP detection process. Figure 2.7 (b) shows the contributions of complex responses for those four points in 36 orientations. From Fig. 2.7 (b), it can be seen that r_2, r_3, and r_4 corresponds to similar angles to the optimum r_1. As pointed out by the author of [2], the local maximum responses usually fall between r_5 and r_{15}. Hence, the average of the responses from r_5 and r_{15} is chosen as the mean of the local maximum responses [2].

The confidence level in the orientation $\theta(z)$ is given by

$$\mathrm{Conf}(z) = 1 - \frac{\mathrm{average}\{r_5(z), \ldots, r_{15}(z)\}}{r_1(z)}. \tag{2.6}$$

Figure 2.6: Sampled texture orientations (red segments) by Gabor filters.

(a) Four points on which Gabor complex responses are evaluated.



(b) The Gabor complex responses for the four points.

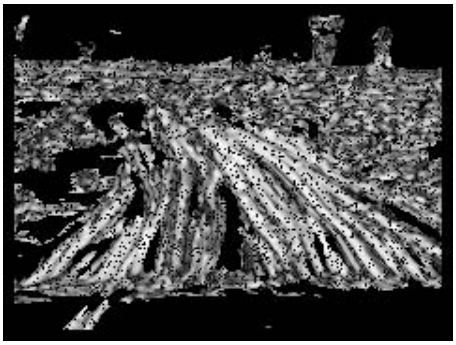
Figure 2.7: Example of Gabor complex responses of image pixels.

In [2], Conf is normalized throughout the image to the range of 0 to 1. After that, all the pixels that have confidence levels that are smaller than

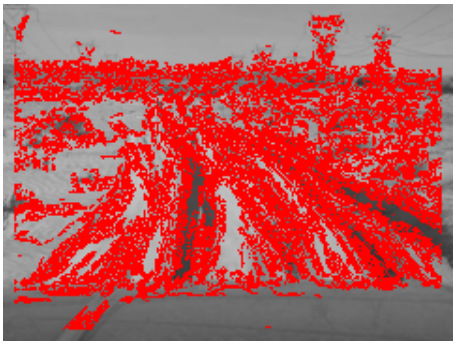
$$\delta \times \left(\max_z \mathrm{Conf}(z) - \min_z \mathrm{Conf}(z) \right), \tag{2.7}$$

with $\delta = 0.3$ are discarded. The remaining pixels become voters that are referred to as the remaining voters in the voting process.

Figure 2.8 (a) shows an example of confidence map for the texture orientation



(a) Confidence map for texture orientation estimation.



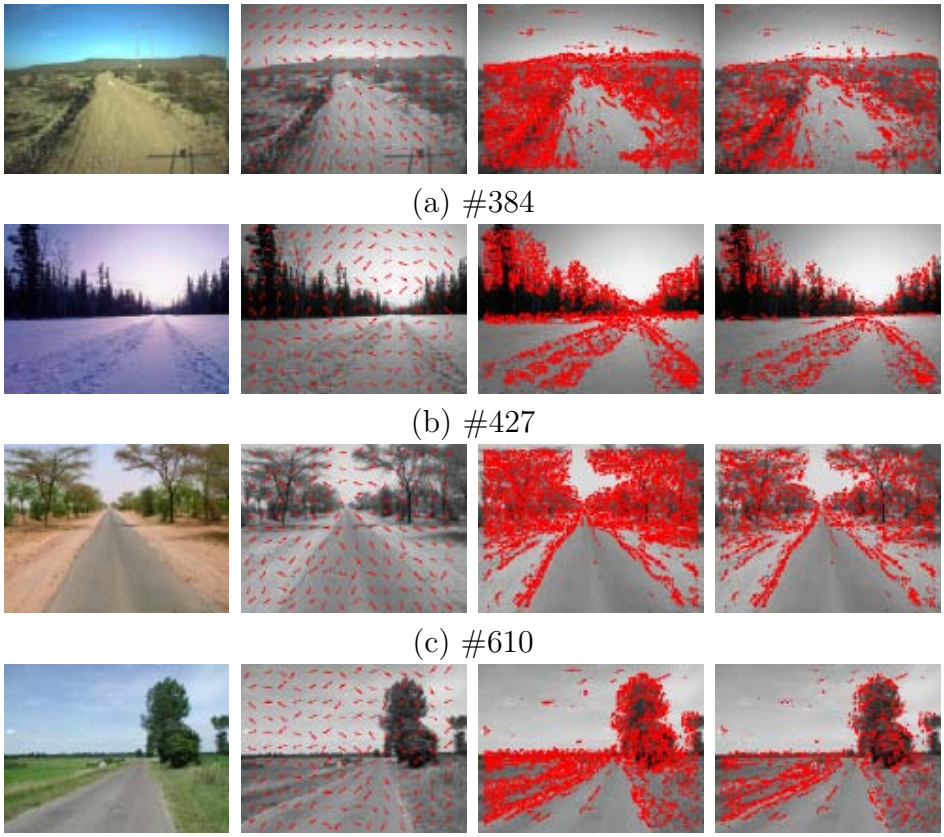
(b) Pixels with confidence level larger than 0.3.

Figure 2.8: Example of confidence map and high confidence level pixels.

estimation. A brighter the pixel indicates a higher confidence level. Figure 2.8 (b) shows the pixels (red points) with a confidence level larger than 0.3.

The experimental results of the present study demonstrate that too many off-road pixels remain when using the threshold $\delta = 0.3$. Since a high number of remaining voters increases the computational cost in the voting process, we use $\delta = 0.5$ in order to reduce the number of remaining voters.

In the examples shown in Fig. 2.9, the red points in the third column indicate the remaining voters for $\delta = 0.3$ and those in the last columns indicate the remaining voters for $\delta = 0.5$. The numbers of the remaining voters are listed in Table 2.1. The number of remaining voters for $\delta = 0.5$ is reduced by approximately 38$\sim$46% compared to that for $\delta = 0.3$. The experimental results reveal that by reducing the number of remaining voters, the total computational time of the algorithm can be reduced approximately 11.48%.



(a) #384

(b) #427

(c) #610

(d) #626

Figure 2.9: Remaining voters for the LASV method and the proposed method (original images, orientations computed by Gabor filters, remaining voters determined by the confidence function of the LASV method, remaining voters determined by the confidence function of the proposed method).

It is obviously that the remaining voters above the true VP are useless for VP detection process. The computational cost of the VP voting algorithm can be reduced if those unnecessary remaining voters can be discarded. Many research works have pointed out that for unstructured roads, edge-based VP estimation methods may have some problems with edge detection because there are no apparent boundaries in unstructured roads scenes. But for our own research experience, a VP estimated by the edge-based method can be used to discard the remaining voters in very top of image (i.e. above the true VP).

In our method, the remaining voters above a temporary VP which is detected based on a Hough transformation are discarded. The Hough transformation method (same

Table 2.1: Number of remaining voters.

	#384	#427	#610	#626
$\delta = 0.3$	13,332	10,118	14,595	10,346
$\delta = 0.5$	8,155	6,074	9,072	5,568

with one proposed in [14]) for detecting the temporary VP is performed as follows:

Step 1 Edge pixels are extracted by the Canny detector.

Step 2 Straight lines are detected by the Hough transform.

Step 3 The intersections of any pair of lines vote for the VP on another Hough space. VP and two borders are detected based on the maximum values in the Hough space.

Figure 2.10 shows an example of VP detection by the Hough transform in a structured road. The remaining voters above this temporary VP will be discarded. For unstructured roads, the temporary VP estimated by Hough transform usually falls on the bottom part of the image (see some examples in Fig. 2.11). Therefore, for those images, only remaining voters in the upper part of the image are discarded. In our method, the remaining voters are determined as follows:

Step 1 Detect a temporary VP by the Hough transform as mentioned above.

Step 2 If the temporary VP is in 70% the upper part of the image, remove all remaining voters above this VP. Otherwise, only remove those remaining voters in 40% the upper part of the image.

Figure. 2.12 visually give a comparison of the number of remaining voters in three cases: i) $\delta = 0.3$, ii) $\delta = 0.5$, and iii) $\delta = 0.5$ with using the temporary VP estimated by the Hough transform. It can be seen that by using $\delta = 0.5$ and the temporary VP, the number of remaining voters is considerably reduced. The effect of using the temporary VP on VP estimation performance will be discussed later in Section **2.6**.

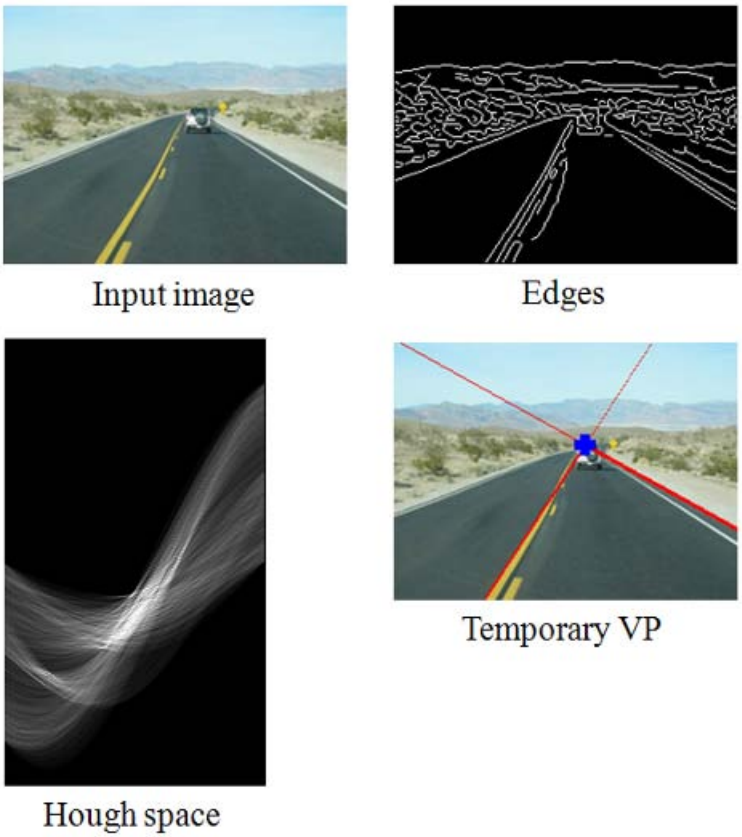


Figure 2.10: Example of temporary VP estimated by Hough transform.

2.4 Proposed local soft voting method

In this section, a VP candidate region is introduced, as well as a new local soft voting method is proposed in order to improve the estimation performance and reduce the computational cost of the algorithm.

The LASV method [2] is performed as follows: i) scan the VP candidates as the uppermost 90% of the pixels in the image ii) create a half-disk voting region for each VP candidate iii) calculate the voting score received by each VP candidate from the remaining voters in its half-disk voting region and iv) obtain the VP as the VP candidate having the largest voting score. As stated in Introduction, the computational cost of the LASV method is high due to a large number of scanning pixels, and this method yields an estimation error in some images.

In our method, the basic idea for reducing the computational cost is to scan the remaining voters instead of the VP candidates.

In order to construct the VP candidate region for each remaining voter, two conditions are considered: i) the VP candidates of a remaining voter are always above it in the image and ii) the angle between the direction from a remaining voter to a VP candidate and the texture orientation at that remaining voter is smaller than a certain threshold. The remaining voter votes for the pixels in its VP candidate region.

Figure 2.13 shows examples of the VP candidate region: a circular sector in the left-hand figure and a triangular region in the right-hand figure, where P is a remaining voter, $\overrightarrow{\mathrm{O_P}}$ is its texture orientation, and ϵ is the angle tolerance.

Initially, the circular sector is more intuitive than the triangular region for use in the proposed method. Note that in the LASV method every remaining voter votes for the pixels whose distance from the voter is less than R with $\gamma \leq \frac{5}{1+2d(\mathrm{P,V})}$, which means the VP candidate region of a voter in the LASV method is included in the circular sector depicted in the left side of Fig. 2.13.

However, from the viewpoint of computational cost, scanning the inside of the triangular region is much simpler. Thus, we use the triangular region [39]. Note that once R (:= |PQ|, the length of the line segment PQ) and ϵ are given, the vertices I and J are determined automatically.

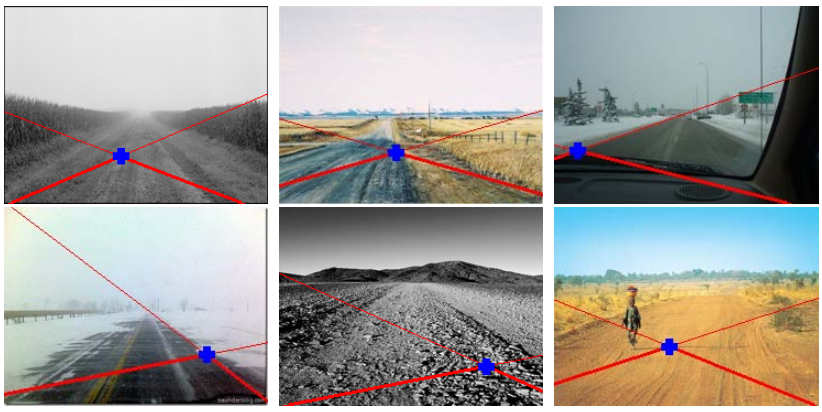
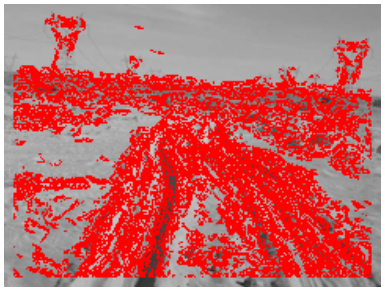
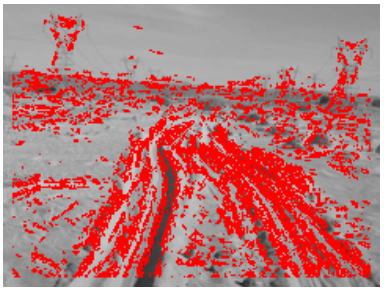


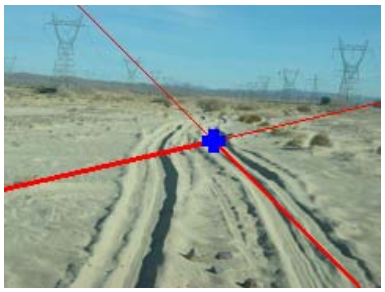
Figure 2.11: Examples of temporary VP by Hough transform in unstructured roads.



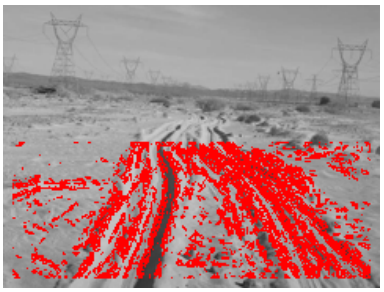
$\delta = 0.3$



$\delta = 0.5$



Temporary VP



$\delta = 0.5$ + Hough transform

Figure 2.12: Comparison of number of remaining voters (red points).

As a voting process, a local soft voting method is adopted in which the remaining voter votes for the VP candidate in its VP candidate region all the more as it is close to the VP candidate, and the angle between the orientation of its texture and the direction from it to the VP candidate is close to zero. This indicates that the voting scores of Q should be smaller than that of L and larger than that of I or J (see the right-hand figure of Fig. 2.13).

The proposed voting score function is defined as follows:

$$\text{Vote}(\text{P}, \text{V}) = \frac{\exp(-\alpha/\beta)}{1 + d(\text{P}, \text{V})^2}, \tag{2.8}$$

where V is a pixel (a VP candidate) in the VP candidate region of P, α is the distance from V to the pixel on PQ with the same y coordinate of V, and β is a constant parameter. Note that the voting score decreases as the distance $d(\text{P}, \text{V})$ or α increases, and the voting scores of pixels near I and J are approximately equal to zero. Hence,

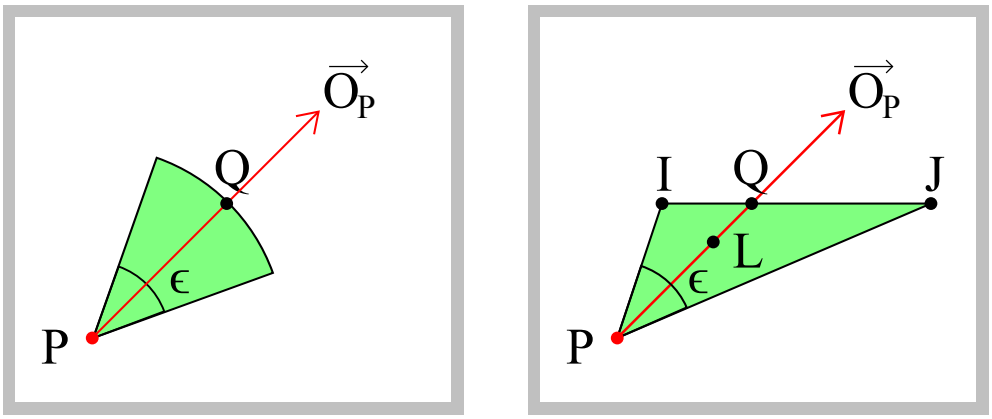


Figure 2.13: VP candidate regions.

the pixels near I and J of the triangular region are useless for the soft voting process. Based on these observations, the VP candidate region is modified from the triangular region to a shape that is a combination of a triangle and a parallelogram [40], as shown in Fig. 2.14.

The fundamental strategy to construct this modified VP candidate region is to reduce the computational time. This modified VP candidate region can be drawn if PK and PQ are determined. In our method, we use $|\mathrm{PK}| = 0.50 \times H$ and $|\mathrm{PQ}| = 0.65 \times H$, which yields the best performance in all our experiments.

The constant parameter β is selected to satisfy that the voting score of the pixel very near Q is approximately equal to the voting score of Q, and the voting score of the pixel very near K is approximately equal to the voting score of K (see Fig. 2.14). For instance, the voting score of Q (with α=0) should be approximately equal to the voting score of S (with α=1) in Fig. 2.14 (note that Q and S have the same y coordinate).

Figure 2.15 shows an example of the voting score of Q (with α=0) and S (with α=1) in the case that the texture orientation at P is 45°. In the figure, the black dashed line indicates the voting score of Q, and the red line indicates the voting score of S. We see that, when β increases, the voting score of S becomes approximately equal to that of Q. Our experimental results reveal that when the texture orientation at P varies from 0° to 175°, by using β=180, we obtain the best performance of VP

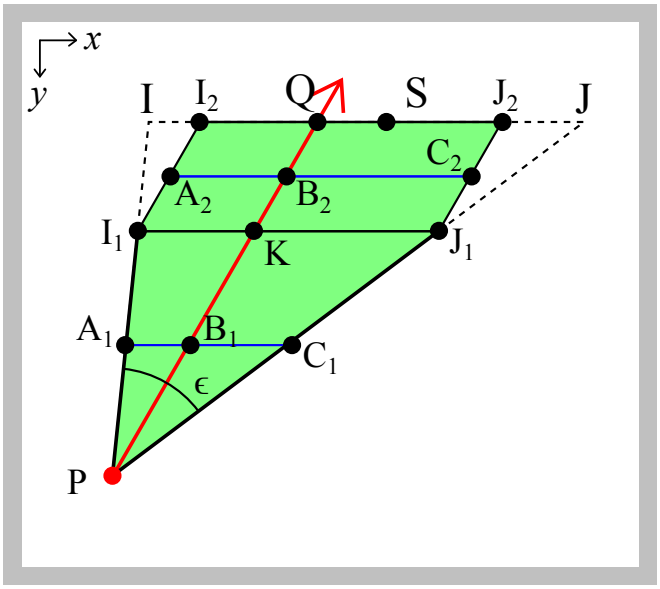


Figure 2.14: Modified VP candidate region.

detection.

Next, we summarize the proposed local soft voting method.

Step 1 Let M be a two-dimensional matrix of the same size as the road image, and set all the elements of M to zero. Let $R_1 := 0.50 \times H$ and $R_2 := 0.65 \times H$.

Step 2 For each remaining voter $\mathrm{P}(x_0, y_0)$, calculate the coordinates of the points $\mathrm{K}(x_1, y_1)$ and $\mathrm{Q}(x_2, y_2)$ as follows:

$$x_1 := x_0 - \sin(\theta) \times R_1, \tag{2.9}$$
$$y_1 := y_0 + \cos(\theta) \times R_1, \tag{2.10}$$
$$x_2 := x_0 - \sin(\theta) \times R_2, \tag{2.11}$$
$$y_2 := y_0 + \cos(\theta) \times R_2, \tag{2.12}$$

where θ is the texture orientation angle at P, and repeat procedures (a) and (b) below:

(a) (Calculation of voting scores in the triangle $\mathrm{PI_1J_1}$) For $x := x_1$ to x_0, calculate

$$y_{10} := y_0 + (x - x_0)/\tan(\theta), \tag{2.13}$$
$$y_{11} := y_0 + (x - x_0)/\tan(\theta + \epsilon), \tag{2.14}$$
$$y_{12} := y_0 + (x - x_0)/\tan(\theta - \epsilon), \tag{2.15}$$

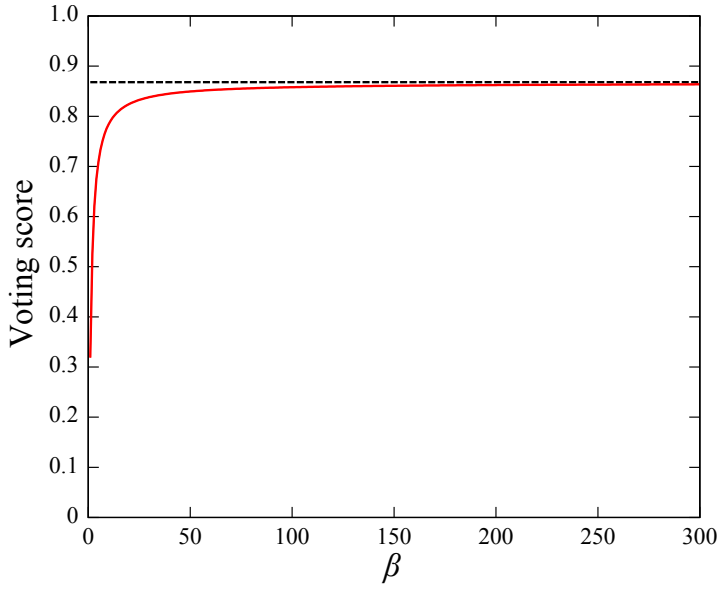


Figure 2.15: An example of voting score for Q (α=0) and S (α=1).

where $\mathrm{B}_1(x, y_{10})$, $\mathrm{A}_1(x, y_{11})$, and $\mathrm{C}_1(x, y_{12})$ (see Fig. 2.14), and calculate the voting score repeatedly for $y := y_{11}$ to y_{12}

$$M(x, y) := M(x, y) + \frac{\exp(-|y - y_{10}|/\beta)}{1 + d(\mathrm{P}, \mathrm{V})^2}, \tag{2.16}$$

$$d(\mathrm{P}, \mathrm{V})^2 = ((x - x_0)^2 + (y - y_0)^2)/\mathrm{Diag}^2, \tag{2.17}$$

where $\mathrm{V}(x, y)$, β=180, and Diag denotes the diagonal length of the input image.

(b) (Calculation of voting scores in the parallelogram $\mathrm{I}_1\mathrm{J}_1\mathrm{J}_2\mathrm{I}_2$) For $x := x_2$ to $x_1 - 1$, calculate

$$y_{20} := y_0 + (x - x_0)/\tan(\theta), \tag{2.18}$$

$$y_{21} := y_{20} - |\mathrm{KI}_1|, \tag{2.19}$$

$$y_{22} := y_{20} + |\mathrm{KJ}_1|, \tag{2.20}$$

where $\mathrm{B}_2(x, y_{20})$, $\mathrm{A}_2(x, y_{21})$, and $\mathrm{C}_2(x, y_{22})$ (see Fig. 2.14), and calculate the voting score repeatedly for $y = y_{21}$ to y_{22}

$$M(x, y) := M(x, y) + \frac{\exp(-|y - y_{20}|/\beta)}{1 + d(\mathrm{P}, \mathrm{V})^2}. \tag{2.21}$$

Step 3 Find the element of M that has the largest value, and let its index be the coordinate of the VP.

2.5 Algorithm summary

To obtain the VP, the proposed method is performed as follows:

Step 1 Calculate the texture orientation at every pixel of the road image using Gabor filters with five scales and 36 orientations (Section **2.3**).

Step 2 Keep the pixels having confidence levels that exceed the threshold (2.7), with $\delta = 0.5$ (Section **2.3**) and below a temporary VP estimated by using the Hough transform as remaining voters.

Step 3 Perform the proposed local soft voting method to obtain a VP (Section **2.4**).

2.6 Experimental results

2.6.1 Image dataset

The proposed method is compared to the LASV method using numerical examples. Most of these images have been used by Kong et al. in [2]; the remainder were downloaded from the Internet by using Google Image. Among them, about 600 images are unstructured roads, and about 400 images are structured roads. These road images contain large variations in color, texture, lighting condition and surrounding environment without any prior known camera parameters, some of them are shown in Fig. 2.16.

Since these images are of very different size, all images are normalized to the same size (height: 180 pixels, width: 240 pixels) by using the bicubic image interpolation method [41]. To assess the algorithm's performance versus human perception of the VP location, we invited 7 persons to manually mark the VP location in each image in this collection after they are trained to know the vanishing point concept.

Since the marked VPs in each image are very close, we defined the center of these marked locations as the ground truth VP location. Figure 2.17 illustrates how the ground truth locations of VPs are determined. For each image in the figure, the green points are the marked points by 7 persons, and the red point is computed as the ground truth of VP.

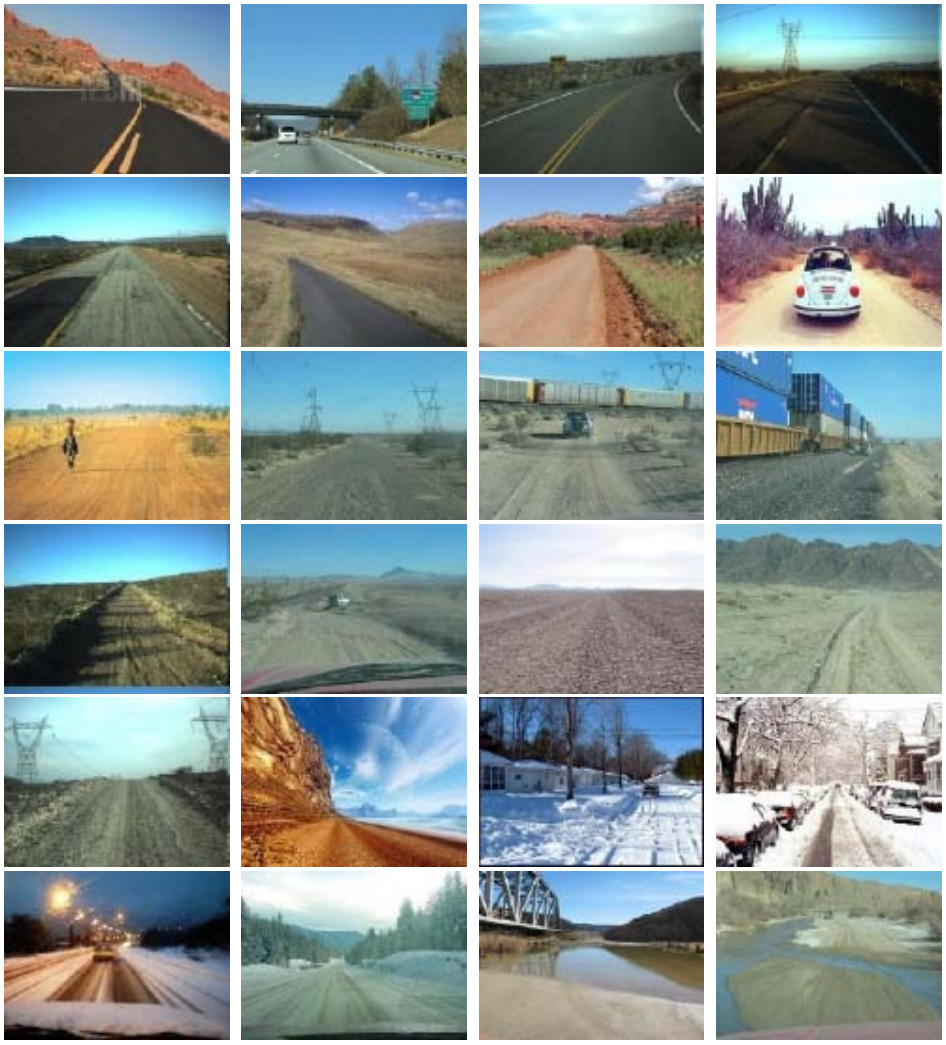


Figure 2.16: Different road types with varying colors, textures, and illumination conditions.

In order to measure the accuracy of VP estimation algorithm, we use the normalized Euclidean distance, where the Euclidean distance between the estimated VP and the ground truth is normalized by the diagonal length of the road image as follows:

$$\text{NormDist} = \frac{\sqrt{(x_e - x_g)^2 + (y_e - y_g)^2}}{\text{Diag}}, \tag{2.22}$$

where (x_e, y_e) is the estimated VP position, and (x_g, y_g) is the marked ground truth position (see Fig. 2.18). Since the input image is normalized to 180×240 pixels, the normalized Euclidean distance of 0.1 in (2.22) means that the location of the estimated

Figure 2.17: Examples of marked ground truth VP.

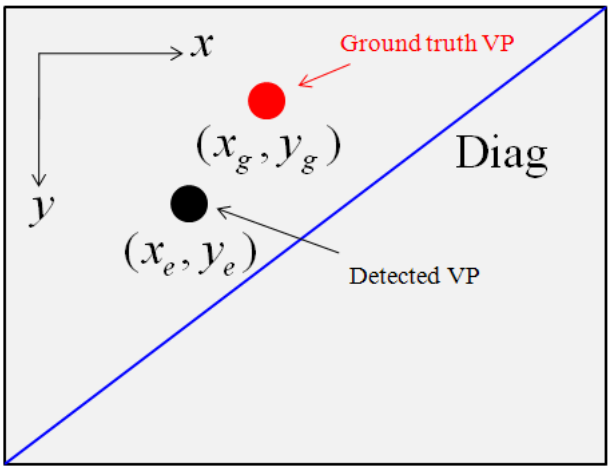


Figure 2.18: An Illustration of the ground truth VP and detected VP.

VP is about 30 pixels away from that of the marked ground truth.

2.6.2 Comparing experimental results

In order to assert the effectiveness of the proposed local soft voting method and the new VP candidate region, we compare the performances of six VP detection algorithms. Table 2.2 and Table 2.4 show the experimental results for 1000 tested images without using the Hough transform to calculate the temporary VP.

In Table 2.2, the four methods (with $\delta = 0.5$) refer to the method described in Section **2.4**. In particular, the "Soft+Modified" denotes the proposed method, and the "Soft+Triangle" denotes the method using a triangular VP candidate re-

Table 2.2: Performances of the proposed method (Soft+Modified) and the other three methods.

	Methods			
Average	Soft+Modified	Soft+Triangle	Hard+Modified	Hard+Triangle
NormDist	0.0729	0.0737	0.0739	0.0735
Total time (s)	5.213	5.245	4.388	4.400
Prep. time (s)	0.012	0.012	0.012	0.012
Voter determination time (s)	3.966	3.966	3.966	3.966
Voting time (s)	1.235	1.267	0.410	0.422

gion. The "Hard+Modified" denotes the method using a hard voting strategy, and the "Hard+Triangle" denotes the method using a triangular VP candidate region and a hard voting strategy. (Note that the hard voting strategy is performed by replacing (2.8) with $\mathrm{Vote}(\mathrm{P}, \mathrm{V}) = 1$.)

In Table 2.4, the "LASV0" denotes the LASV method with the radius R of the half-disk voting region being set to $0.35 \times H$ and $\delta = 0.3$ as proposed in [2]. The "LASV1" denotes the LASV method with R being set to $0.65 \times H$ and $\delta = 0.6$ (note that this "LASV1" method yields the best estimation performance for the LASV method in all our experiments).

In the both tables, the "NormDist" denotes the average normalized Euclidean distance (note that a smaller value means the estimated VP location is closer to the location of the ground truth), and the "Total time" denotes the average computational time for each method. The "Prep. time" denotes the average preprocessing time, the "Voter determination time" denotes the average time for calculating the Gabor convolution and confidence level estimation, and the "Voting time" denotes the average time for the voting process for each method.

Firstly, we compare the performances of the proposed method (Soft+Modified) and the other three methods (Soft+Triangle, Hard+Modified, and Hard+Triangle). From Table 2.2, it can be seen that the proposed method yields the best estimation result among the four methods.

Table 2.3 shows the estimation performance (NormDist) of the proposed method with changing the values of |PK| and |PQ| in Fig. 2.14. In the table, |PK| is varied from $0.45 \times H$ to $0.60 \times H$ with an interval being set to $0.05 \times H$, and |PQ| is varied from $0.60 \times H$ to $0.70 \times H$ with an interval being set to $0.05 \times H$. From the table, it can be seen that the proposed method with $|\mathrm{PK}| = 0.50 \times H$ and $|\mathrm{PQ}| = 0.65 \times H$ yields the best performance in all experiments.

Secondly, we compare the estimation performances of the proposed method, the LASV0 method, and the LASV1 method. The experimental results in Table 2.4 reveal that the estimation performance of the proposed method is considerably better than that of the LASV0 method, and almost the same as the LASV1 method (the difference between the NormDist values of the proposed method and the LASV1 method is 0.0020 which represents a value less than one pixel). In order to investigate the details of these experimental results, we evaluate the VP estimation performance while changing the threshold for Euclidean distances.

Figure 2.19 shows a comparison of VP estimation performance between the three methods. In the figure, the horizontal axis represents the Euclidean distances in pixels, while the vertical axis represents the number of images whose VP estimation error is less than the corresponding Euclidean distance. From this figure, it can be seen that the estimation performance of the proposed method is better than that of the LASV0 method, and almost the same as the LASV1 method.

Next, the effect of using Hough transform for estimating the temporary VP in order to reduce the number of off-road voters is discussed. Table 2.5 shows the experimental

Table 2.3: Performance of the proposed method with changing the values of |PK| and |PQ| in Fig. 2.14.

	$\|\mathrm{PQ}\| = 0.60 \times H$	$\|\mathrm{PQ}\| = 0.65 \times H$	$\|\mathrm{PQ}\| = 0.70 \times H$
$\|\mathrm{PK}\| = 0.45 \times H$	0.0749	0.0740	0.0735
$\|\mathrm{PK}\| = 0.50 \times H$	0.0746	0.0729	0.0755
$\|\mathrm{PK}\| = 0.55 \times H$	0.0745	0.0741	0.0741
$\|\mathrm{PK}\| = 0.60 \times H$	0.0742	0.0739	0.0745

Table 2.4: Performances of the proposed method, LASV0 and LASV1 methods without using the Hough transform.

	Methods		
Average	Soft+Modified	LASV0	LASV1
NormDist	0.0729	0.0948	0.0709
Total time (s)	5.213	33.820	24.815
Prep. time (s)	0.012	0.117	0.117
Voter determination time (s)	3.966	3.966	3.966
Voting time (s)	1.235	29.737	20.732

results for 1000 tested images with using the Hough transform to calculate the temporary VP for the proposed method, the LASV0 method, and the LASV1 method. We also evaluate the estimation performance of VP detected by the Hough transform. In particular, this method is denoted as "Hough" which is shown in the last column of Table 2.5. From the table, it can be seen that the estimation performance of temporary VP detected by the Hough transform is quite low. However, by using the temporary VP, the estimation performance of the proposed method is improved (compared with ones in Table 2.4).

In addition, we also evaluated the performance of the proposed method without using the temporary VP detected by the Hough transform, but pixels in the uppermost part of each image are discarded. In particular, we did the experiments for the proposed method with cutting 30%, 40%, 50%, and 60% the uppermost pixels of each image (this also means that the remaining voters in those areas are also discarded). The experimental results reveal that all the estimation performances are worse than that of the proposed method with using the Hough transform (the NormDist values for the four cases mentioned above are 0.0682, 0.0635, 0.0633, and 0.0726, respectively).

Next, the computational times of the three methods are discussed. From Table 2.4 and Table 2.5, it can be seen that the average computational time of the proposed method is considerably less than that of the LASV0 method and the LASV1 method. In particular, in Table 2.4, the "Total time" and the "Voting time" of the proposed method are approximately five times and 17 times less than those of the LASV1

Table 2.5: Performances of the proposed method, LASV0, and LASV1 methods with using the Hough transform.

	Methods			
Average	Soft+Modified	LASV0	LASV1	Hough
NormDist	0.0630	0.1084	0.0620	0.1460
Total time (s)	5.265	23.031	21.641	0.116
Prep. time (s)	0.128	0.233	0.233	
Voter determination time (s)	3.966	3.966	3.966	
Voting time (s)	1.171	18.832	17.442	

method, respectively. The "Total time" and the "Voting time" of the proposed method are approximately 6.5 times and 24 times less than those of the LASV0 method, respectively. These results show that the proposed voting strategy requires much less computational cost than the LASV method.

Note that the "Prep. time" of the proposed method is slightly less than that of the LASV method due to the difference in the number of preprocessing steps in which a median filter is applied in both methods and in addition a vertical edge elimination method is applied in the LASV method (our experimental results turn out that the proposed method without using the vertical edge elimination method is better and faster than the proposed method with using the vertical edge elimination method, in which the "NormDist" and the "Total time" are 0.0732 and 5.311(s), respectively. Hence, the vertical edge elimination method is not used in the proposed method).

These numerical examples are performed using Matlab, running on a Core 2 Duo (3.5-GB RAM) machine, and the Matlab m-files for the LASV method were provided by the author of [2].

Figure 2.20 shows some examples of VP detection on some sample images, in which the VPs detected by the LASV0 method, the LASV1 method and the proposed method almost coincide with the corresponding ground truth VPs. Fig. 2.21 shows some examples of VP detection on some sample images, in which the VPs detected by the LASV1 method and the proposed method are better than that of the LASV0 method.

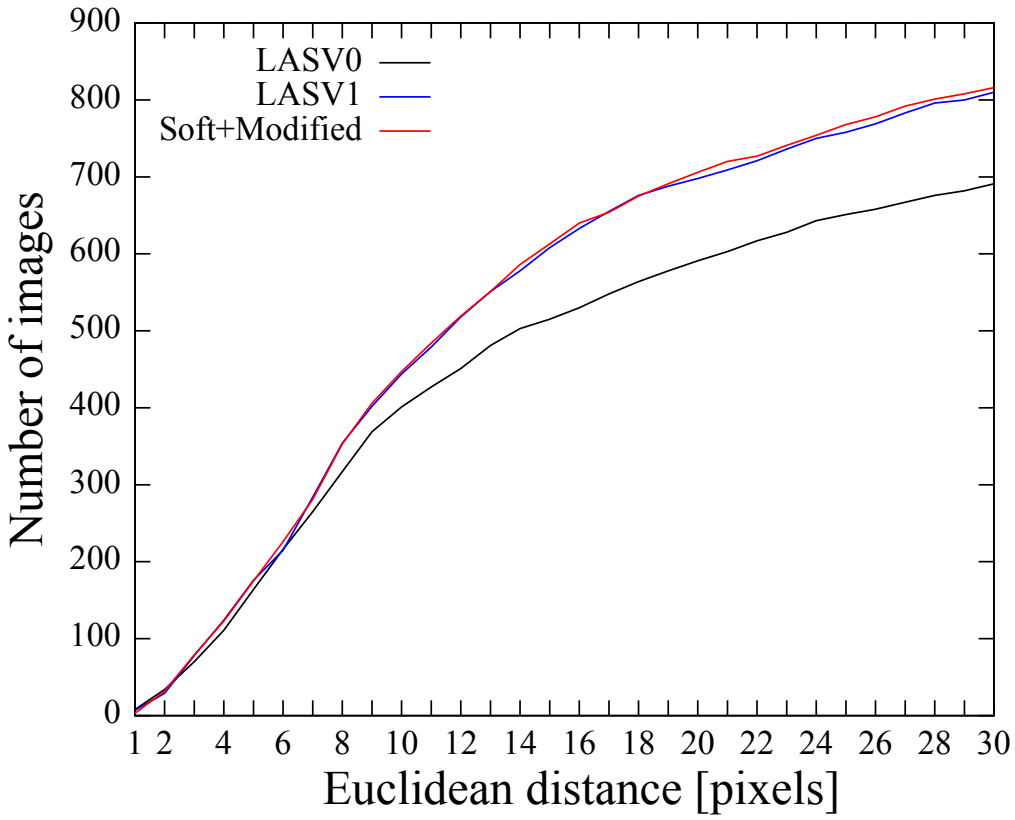


Figure 2.19: Comparison of VP estimation performance.

In order to assert the effectiveness of the proposed method for general road image, the image dataset is divided into six types of roads: desert roads, structured roads, snow-covered roads, rural roads, dark roads, and highway roads.

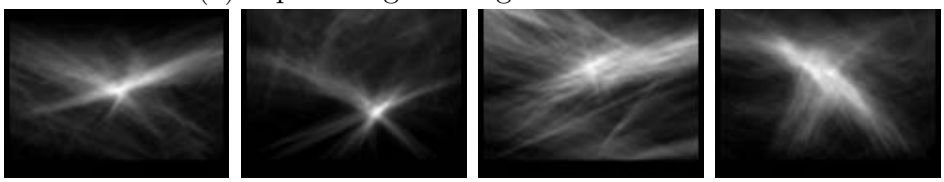
Figure 2.22, Fig. 2.23, Fig. 2.24, Fig. 2.25, Fig. 2.26, and Fig. 2.27 show some examples of VP detection for each type of roads. From those figures, it can be seen that the proposed VP detection method performs well for general road images.

2.6.3 Effect of radius on performance

In this section and the next section, we discuss about the two important parameters R and δ.

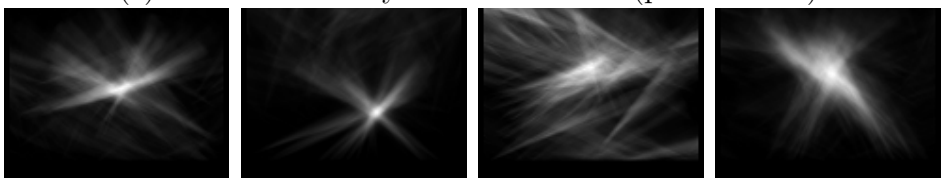
In this section, we focus on the effect of radius R on the estimation performance. The LASV1 method (with $R = 0.65 \times H$ and $\delta = 0.6$) yields a better estimation

(a) Input images and ground truth VPs.



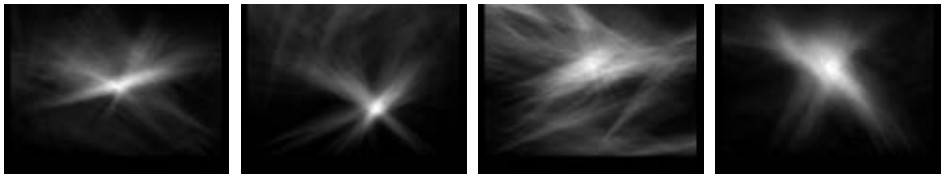
(b) Voting maps by LASV0 method.

(c) Detected VPs by LASV0 method (pink crosses).



(d) Voting maps by LASV1 method.

(e) Detected VPs by LASV1 method (pink crosses).

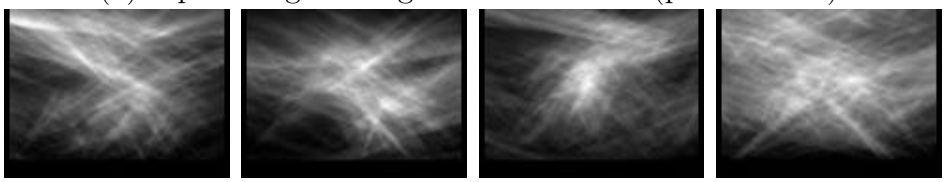


(f) Voting maps by the proposed method.

(g) Detected VPs by the proposed method (pink crosses).

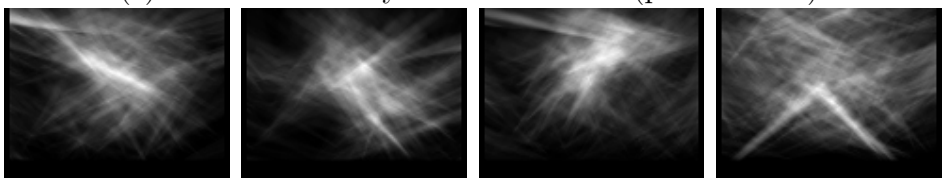
Figure 2.20: Examples of proper VP detection by three methods.

(a) Input images and ground truth VPs (pink crosses).



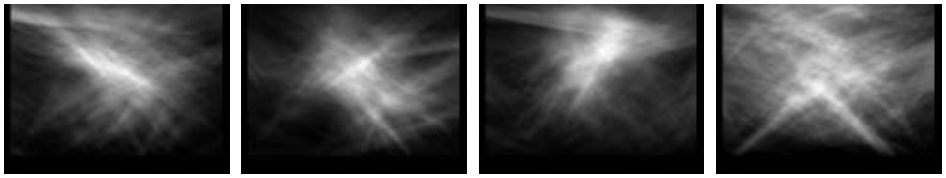
(b) Voting maps by LASV0 method.

(c) Detected VPs by LASV0 method (pink crosses).



(d) Voting maps by LASV1 method.

(e) Detected VPs by the LASV1 method (pink crosses).



(f) Voting maps by the proposed method.

(g) Detected VPs by the proposed method (pink crosses).

Figure 2.21: Examples of improper VP detection by the LASV0 method.

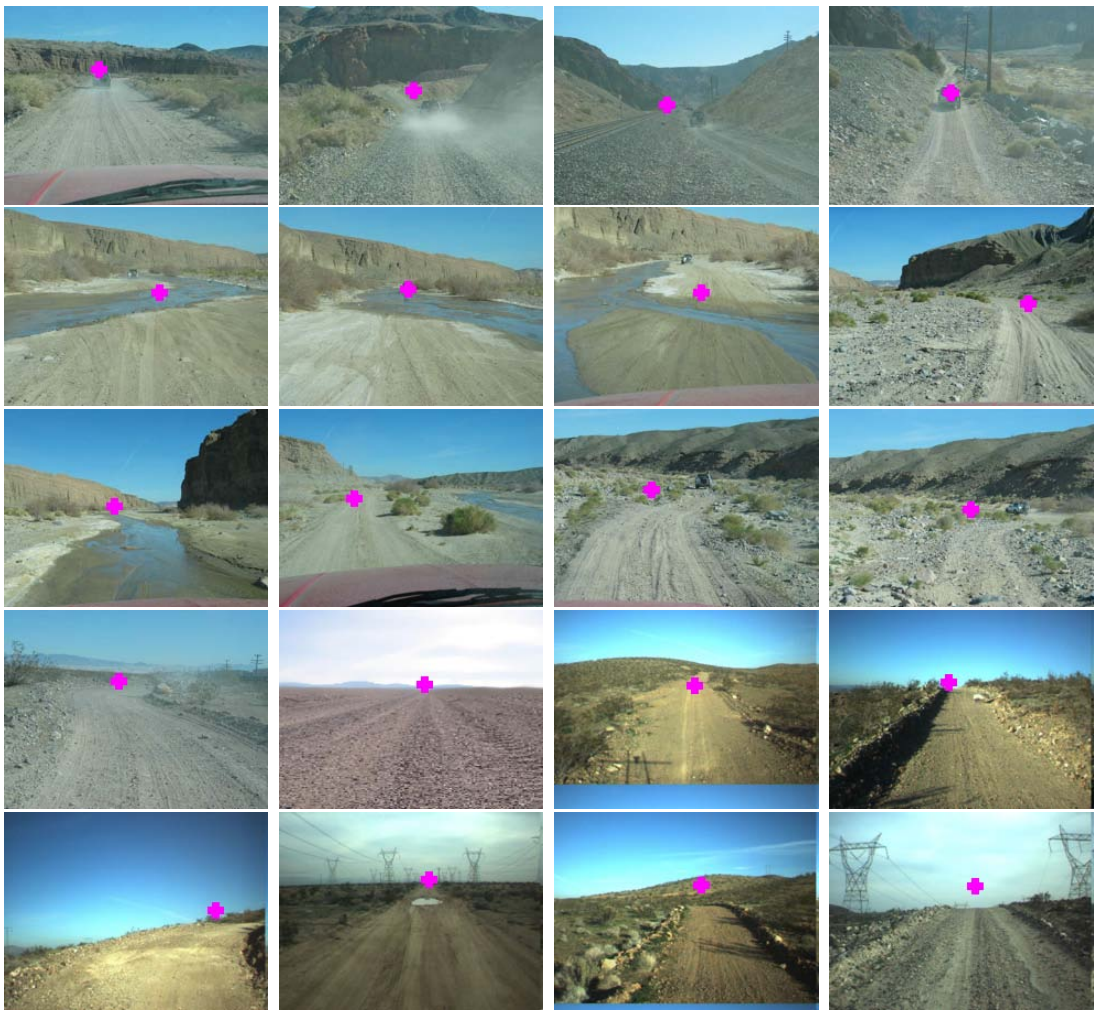


Figure 2.22: Examples of VP detection of the proposed method for desert roads.

Figure 2.23: Examples of VP detection of the proposed method for structured roads.

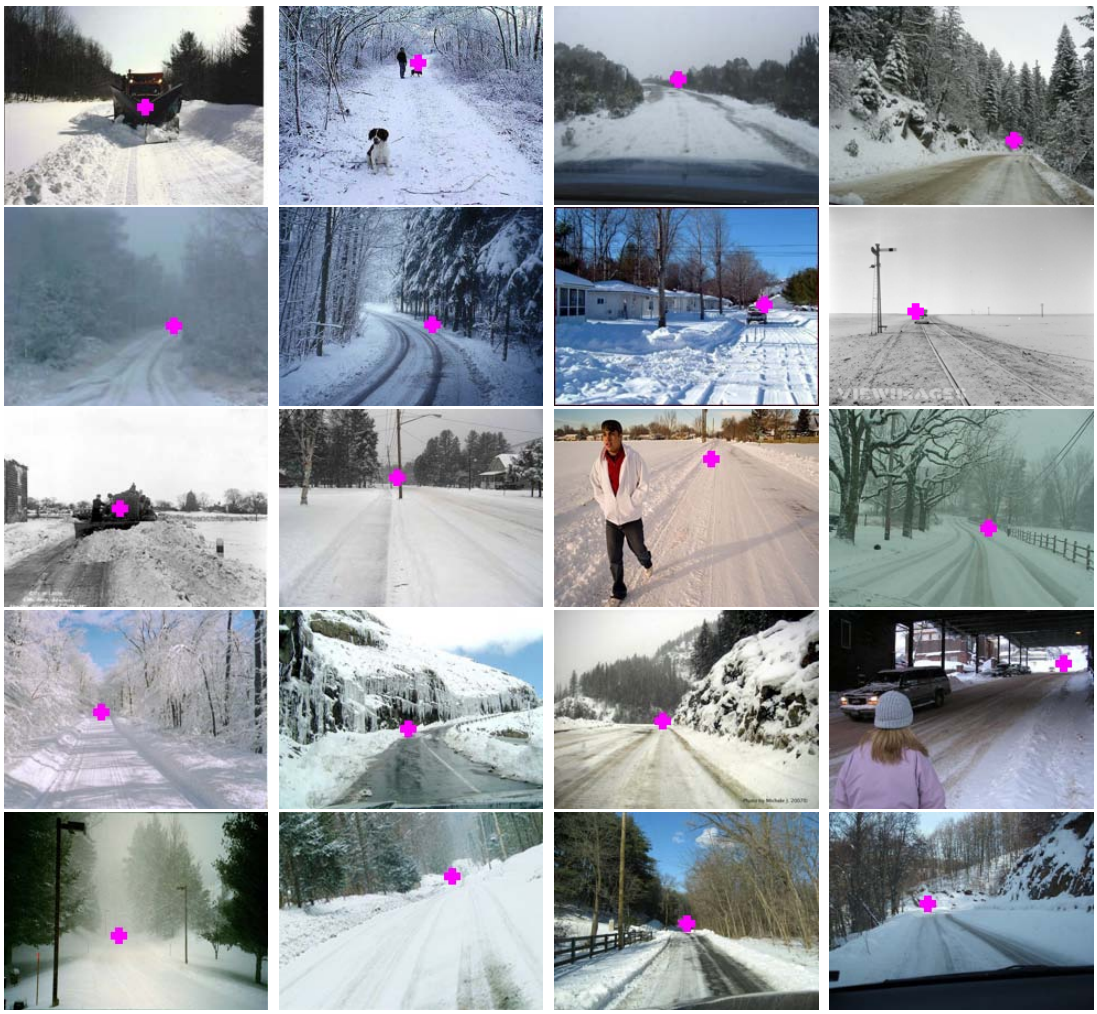


Figure 2.24: Examples of VP detection of the proposed method for snow-covered roads.

Figure 2.25: Examples of VP detection of the proposed method for rural roads.

Figure 2.26: Examples of VP detection of the proposed method for dark roads.

Figure 2.27: Examples of VP detection of the proposed method for highway roads.

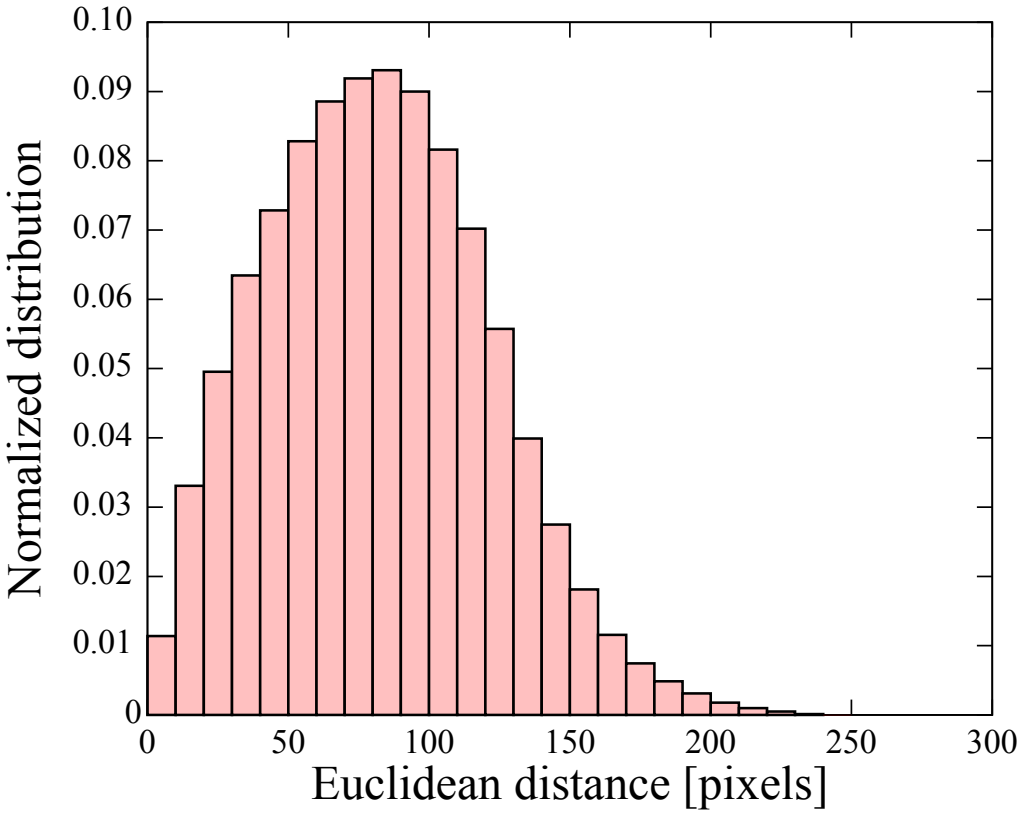


Figure 2.28: Average distribution of distances from the remaining voters to the VP for 1000 road images.

accuracy than the LASV0 method (with $R = 0.35 \times H$ and $\delta = 0.3$).

In order to investigate these results from the viewpoint of radius R, an additional experiment for the LASV method is carried out with $R = 0.35 \times H$ and $\delta = 0.6$ (R is smaller than that of the LASV1 method while δ is the same), and this is denoted by "LASV2". As a result, the average NormDist of the LASV2 method is 0.0999, which is worse than that of the LASV1 method. This result indicates that a small radius of a small voting region yields a worse estimation performance than a larger one.

The above result can be confirmed by the average distribution graph shown in Fig. 2.28. In the figure, the horizontal axis represents the distances from the ground truth VPs in pixels, while the vertical axis represents the normalized distributions of remaining voters. The normalized distribution value for $\mathrm{d} \leq \mathrm{x} < \mathrm{d} + 10$ ($\mathrm{d} = 0,\ 10,\ 20, \ldots,\ 240$; x: Euclidean distance from the ground truth VP) is defined as the

number of the remaining voters satisfying $d \leq x < d + 10$ divided by the total number of the remaining voters.

The graph in Fig. 2.28 shows the average normalized distributions of 1000 images. It can be seen from the graph that the number of remaining voters near the VP is small, which indicates that a small radius of a voting region cannot cover the many voters which possibly vote for the VP. This leads to the fact that the performance of the LASV0 method is worse than that of the LASV1 method.

Also in our proposed method, the graph implies that a short |PQ| (i.e., a small VP candidate region) is not suitable for most images because many remaining voters which possibly vote for the VP cannot vote for it when |PQ| is short. Note that in general the pixels near the VP correspond to the road area far from the camera in the real world, and hence these pixels tend to be blurred. This explains that the number of remaining voters near the VP is small, because most of the blurred pixels cannot be remaining voters since their confidence levels of orientations are generally low.

2.6.4 Effect of δ on performance

In this section, we discuss about the effect of δ on estimation performance.

In order to evaluate how the threshold δ affects the estimation performance and the computational cost, we vary δ of the proposed method from 0.30 to 0.80 with an interval of 0.05, and the result is shown in Fig. 2.29. From the figure, it can be seen that the computational cost almost monotonously decreases with respect to the value of δ. On the other hand, a large value of δ and a small value of δ decrease the estimation performance. When δ is varied from 0.50 to 0.70, the proposed method yields almost the same estimation performance (with less than one pixel difference).

From the viewpoint of estimation performance, we use $\delta = 0.50$ which yields the best result in our method.

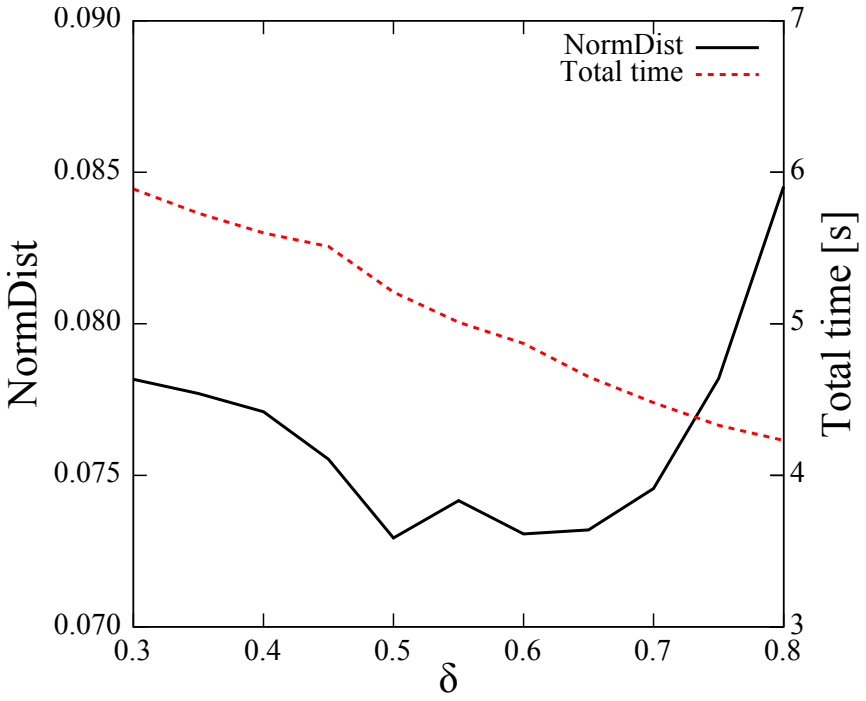


Figure 2.29: The effect of δ on the VP estimation performance and the computational cost for the proposed method.

2.7 Conclusions

Estimating a proper location of VP from a single road image without any prior known camera parameters is a challenging problem due to limited information from the input image. The LASV method [2] for VP detection is very effective for both structured and unstructured roads, and faster than previous texture-based method. However, the computational cost is still high due to a large number of scanning pixels. In addition, an estimation error is obtained in some images, in which the radius of the proposed half-disk voting region is not large enough. Although using the half-disk voting region with a larger value of R may improve the estimation performance of the LASV method, it also increases the computational cost of the algorithm. In this chapter, a new local soft voting method has been proposed to overcome the limitations of the LASV method.

In Section **2.3**, Gabor filters and confidence level function were introduced. In

this section, Gabor filters with 5 scales and 36 orientations are used to calculate the texture orientation at every pixel of the road image, and the confidence level function is used to determine the remaining voters which are useful for the VP voting process by checking the reliability of the obtained texture orientations. A larger threshold value for confidence level function was introduced and a new algorithm based on Hough transform to discard remaining voters in very top of the image was proposed in order to reduce the number of remaining voters.

In Section **2.4**, our novel local soft voting method for VP detection was proposed, in which the number of scanning pixels is much reduced (the remaining pixels are scanned instead of the image pixels) in order to reduce the computational cost, and a new VP candidate region was introduced in order to improve the estimation accuracy of the VP detection algorithm.

The experimental results were demonstrated to show the effectiveness of the proposed method in Section **2.6**. In order to assert the effectiveness of the proposed algorithm, the proposed method and the LASV method have been implemented and tested on 1000 road images which contain large variations in color, texture, lighting condition and surrounding environment. On the other hand, the effect of two important parameters R and δ were also discussed in this section.

Our investigation demonstrated that the number of remaining voters near the VP is small, which indicates that a small radius of a voting region cannot cover the many voters which possibly vote for the VP. This leads to the fact that the performance of the LASV0 method is worse than that of the LASV1 method. On the other hand, the computational cost almost monotonously decreases with respect to the value of δ. In addition, a large value of δ and a small value of δ decrease the estimation performance. From the viewpoint of estimation performance, we use $\delta = 0.50$ which yields the best result in our method.

The experimental results revealed that: i) the proposed method outperforms better than the LASV method which uses a small voting region, especially in some images in

which most remaining voters are far from the VP, ii) the proposed method with using a temporary VP detected by Hough transform improves the estimation performance, and iii) the computational cost of the proposed method is considerably less than that of the LASV method, the computational time for the Gabor convolution and confidence level estimation which accounts for most of the computational time in our proposed method is the same for the LASV method, whereas, the computational time of the proposed method for the voting process is much less than that of the LASV method.

Chapter 3

Road area extraction

3.1 Introduction

Over the past few decades, numerous road area extraction methods have been widely published for urban and highway roads, structured roads, and unstructured roads. In all of these studies, estimating a location of vanishing point (VP) is a key requirement. A set of parallel lines in the world space by perspective projection converges to a common point in image space known as the VP. If the VP can be located correctly, then it is more likely to detect the road area properly

Given an image of the road that is not clear. The computer must determine where the road in that image. Numerous road detection methods have been proposed [2], [3], [20], [42], [43], [44], [45], [46], [47], [48] in which the algorithm is mainly divided into two steps: i) detection of the vanishing point associated with the road, ii) segmentation of the road area.

For instance, in [42], a texture detection method using Gabor Filters is proposed to detect distant stair-cases. When close enough, stair-cases are then detected by looking for groups of concurrent lines, where convex and concave edges are partitioned using intensity variation information. Stair-case pose is estimated by a homography search approach. Using an a priori stair-case model, search criteria and constraints

are established to find its vertical rotation and slope. This algorithm is working in theory, however in terms of speed, it is slow and far from real-time.

In the method proposed in [43], zebra-crossings are detected by looking for groups of concurrent lines, edges are then partitioned using intensity variation information. In order to tackle the ambiguity of the detection algorithm in distinguishing zebra-crossings and stair-cases, pose information is sought. Three methods are developed to estimate the pose: homography search approach using an a priori model, finding normal using the vanishing line computed from equally spaced lines, and with two VPs. As pointed out in [20], this algorithm is also time-consuming.

In [44], the authors develop a two-stage algorithm in order to increase robustness. The first stage detects the borders of the road using a contour-based approach and primarily to estimate the Dominant Vanishing Point (DVP). The DVP and the borders of the road are then used to constrain the region where the points of interest, corresponding to the road lane markers, can be extracted. The second stage uses a robust technique based on projective invariant to match the lines and points between two consecutive images in the sequence. Finally, they compute the homography relating the points and lines lying on the road into the two images.

In general, the key to the approaches [42], [43], [44] is to use a voting procedure like a Hough transform on edge-detected line segments to find points where many intersect. Peaks in the voting function are good candidates for vanishing points. This is sometimes called a cascaded Hough transform because the lines themselves may have first been identified via a Hough transform. Broadly, these methods can not applied to unstructured roads lacking clear painted lines and distinct road borders.

Recently, some VP and texture-based methods have been proposed [2], [3], [20], [48]. In these methods, the estimated VP is used as a constraint to detect two dominant edges for segmenting the road area. These methods can be applied for general road images since they use the texture cue. However, the computational time of those methods is still high.

In this chapter, we propose a new road area extraction method. We detect the most immediate straight road part in the direction of the optical axis of the forwarding looking cameras based on estimating two lines (road boundaries) going from the VP and below the VP in the road image. In our method, in order to achieve a one degree level of precision for road boundaries detection, a histogram of 180 angles corresponding to angles of 180 lines going from the detected VP is used. When generating the histogram, the color information of the road image is combined to improve the estimation performance. The proposed road area extraction method has been implemented and tested on 1000 road images. The experimental results show that our proposed method performs well in challenging conditions.

3.2 Related research

As stated above, previous VP-based road area extraction methods [2], [3], [20], [48] have attempted to detect the VP based on texture orientation calculation.

In [20], a straight road segmentation method is given to detect both road borders simultaneously. It is achieved by optimizing a criterion, which is the difference between the average values of some characteristic (e.g., R,G,B color cues) within the image road region and that characteristic in the region outside the road. It may work when the road and off-road regions have different characteristics. However, it usually fails for both cases where there is little difference in color between road and off-road regions, and where the color is not homogeneous in road region.

In [48], the authors obtain the middle line of the road by using the imaginary road support ray. This technique is well adapted to desert (unpaved) roads where there usually is a clear trace left by previous vehicles and these rays exhibit an even distribution. However, it may not work as well on paved roads whose texture is usually sparser. Therefore, finding the middle line may prove more difficult than road borders.

In [2], the algorithm for road detection is performed as follows: i) scan the VP

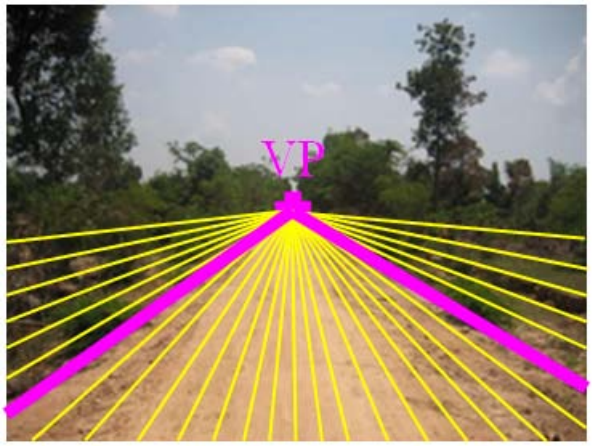


Figure 3.1: 29 constructed lines (yellow lines) going from the detected VP and two detected road borders (pink lines) in [2].

candidates as the uppermost 90% of the pixels in the image ii) create a half-disk voting region for each VP candidate iii) calculate the voting score received by each VP candidate from the remaining voters in its half-disk voting region iv) obtain the VP as the VP candidate having the largest voting score, v) construct a set of 29 lines going from the detected VP and the angle between two neighboring lines is set to 5°, vi) calculate the OCR (Orientation Consistency Ratio) for each line of 29 lines and two road borders can be detected from the maximum of OCR values. An illustration of 29 constructed lines in [2] is shown in Fig. 3.1. The definition of OCR is given in Fig. 3.2. In Fig. 3.2, l is a line consisting of a set of discrete oriented points/pixels (the orientation of these points denoted by a black arrow in the figure). For each point P, if the angle between the point's orientation and the line's direction is smaller than a threshold, this point is viewed to be orientationally consistent with the line. OCR is defined as the ratio between the number of orientationally consistent points and the number of total points on the line.

In [3], the road border extraction algorithm is also based on OCR calculation as same with one in [2]. However, to segment the road area, two most dominant edges are selected as the road borders by combining OCR and color cue. This road detection method integrates texture orientation and color information of the road. In particular,

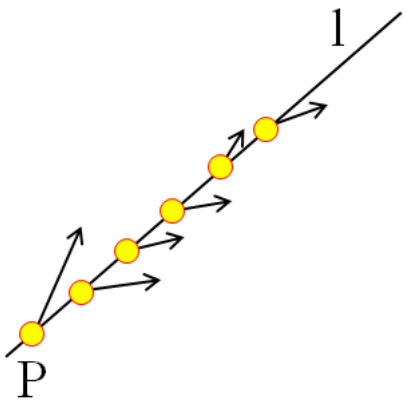


Figure 3.2: An illustration of OCR calculation in [2]. l: one of 29 constructed lines. P: a discrete point on l.

Figure 3.3: An illustration of color calculation in [3].

for each line of 29 constructed lines, the color difference between the two neighboring regions of each ray. Figure 3.3 shows an illustration of neighboring regions for color calculation in [3]. In this figure, A1 and A2 are two neighboring regions of the constructed line l, the area of A1 and A2 is controlled by their wedge angle respectively, which is set to 20°. In this method, the road borders are selected as the rays which maximize the product of OCR value and color difference.

There are three observations: i) the remaining voters which are mostly on the textures and road borders of the image are not used in [2], [3], ii) calculating OCR

for 29 lines may increase the computational time, and iii) using such large regions for color calculation in [3] may also increase the computational cost.

In our method, we use remaining voters for road area extraction since they are very meaningful. In order to achieve a one degree level of precision for road boundaries detection, a histogram of 180 angles corresponding to angles of 180 lines going from the detected VP is used. In order to reduce the computational cost for color calculation, two local parallelograms for color difference calculation are introduced.

3.3 Road area extraction

After having computed texture orientations and estimated the initial location of VP in Chapter 2, it is possible to detect the road region.

In this section, we propose a new VP-constrained method in which histograms based on texture orientations and RGB color information are used to detect the location of the two most dominant borders, and the estimated VP is updated with a joint point of these borders.

3.3.1 Histogram and first border detection

In previous method [2], the road borders are detected based on a calculation of OCR. After detecting the VP, a set of 29 lines going through the VP is constructed such that the angle between two neighboring lines is 5°. For each image pixel, if the angle between the pixel's orientation and the line's direction is smaller than a threshold, this pixel is viewed to be orientationally consistent with the line. In [2], the OCR is defined as the ratio between the number of orientationally consistent pixels and the number of total pixels on the constructed line. Finally, two lines with maximum OCRs being considered as the two most dominant road borders.

In order to improve the performance of road detection, an effective method which integrates texture orientations and color information of the road image was proposed

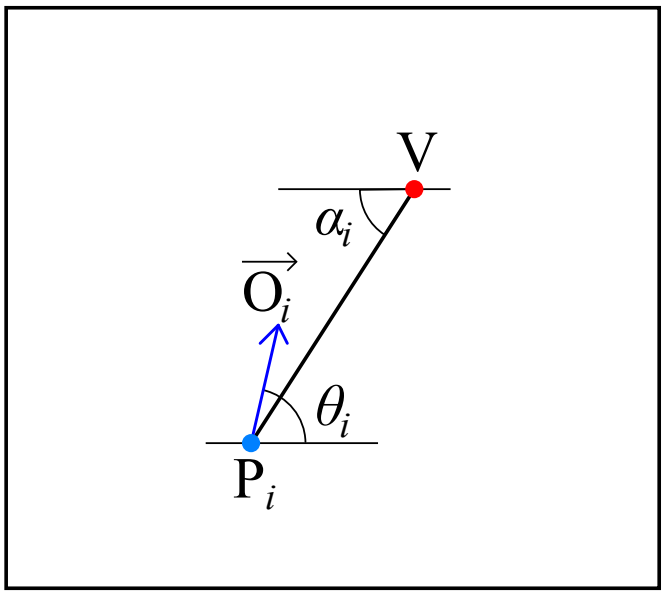


Figure 3.4: Angles for calculating angular difference.

recently by Kong et al. in [3].

The main differences between our method and those methods in [2], [3] lie in the following aspects: i) in the estimation of road region, we do not use all the image pixels as in [2] because a large number of them might be out of the road. In our method, we use only the remaining voters below the detected VP which are high possibility on some edges of the road, ii) instead of constructing a set of 29 lines from the VP with 5° of angular resolution, we consider angles of lines going from the VP passing through the voters, which can help to reduce the angular resolution and improve the accuracy of the algorithm, and iii) when utilizing the color information of the road image for generating histograms, we do not use such large regions as introduced in [3]. Instead, we use the local regions to calculate the color difference as proposed in [49], [50].

To describe our concept, an illustration for calculating the angular difference is given in Fig. 3.4. In the figure, V indicates the estimated VP, P_i indicates the ith voter, $\overrightarrow{O}_i$ indicates the vector orientation of P_i, θ_i indicates the dominant orientation of P_i, and α_i indicates the angle of line going from V passing through P_i. It can be seen that, each voter P_i defines an angle α_i, and α_i will be varied from 0° to 180°. For

each voter P_i, diff(P_i) is used to calculate the angular difference between vector $\overrightarrow{O_i}$ and the direction (VP_i), a smaller angular difference means a higher value of diff(P_i):

$$\text{diff}(P_i) = \exp(-|\alpha_i - \theta_i|). \tag{3.1}$$

On the other hand, in general road images, there is a difference between road and off-road regions which are separated by the road borders. Observing that if the line going from V passing through P_i in Fig. 3.4 is one of the two most dominant road borders, then the colors of two local regions in either side of this line might be remarkable different. From that point, it is worth to use the color information when generating histograms to detect the road borders. It is also needed to point out the distinction between our color calculation method and the method in [3]. The main difference is that after constructing a set of 29 lines from the VP, they calculate the color difference between the two neighboring regions of each line. The wedges of these two neighboring regions are set to 20°, and these are large regions.

In this thesis, we introduce the local regions which are illustrated in Fig. 3.5. In the figure, V indicates the estimated VP, P_i indicates the ith voter, and two parallelograms R_{i1} and R_{i2} indicate two local neighboring regions on either side of the direction (VP_i). In our experiments, d, w, and h are set to 5 pixels, 25 pixels and 30 pixels, respectively. The color difference between R_{i1} and R_{i2} for each channel of the RGB color space is calculated by:

$$\text{diff}(R_{i1}, R_{i2})_c = \frac{|\text{mean}(R_{i1}) - \text{mean}(R_{i2})|}{\sqrt{\text{var}(R_{i1}) + \text{var}(R_{i2})}}. \tag{3.2}$$

where mean() and var() are the mean and variance of pixel values in a region for a single color channel. For a voter P_i, diff(R_{i1}, R_{i2}) indicates the color difference between R_{i1} and R_{i2}, and is defined as the largest of the color difference for each channel:

$$\text{diff}(R_{i1}, R_{i2}) = \max\{\text{diff}(R_{i1}, R_{i2})_c | c = \{R, G, B\}\}. \tag{3.3}$$

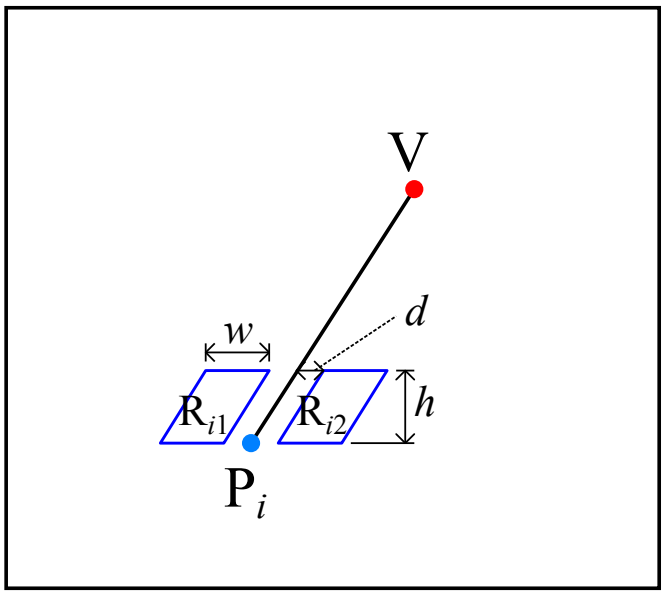


Figure 3.5: Parallelograms for calculating color difference.

To generate the histogram, the algorithm is performed as follows:

Step 1 For each voter P_i, calculate the related angle α_i. After that, two measures are computed: i) the angular difference between α_i and the dominant orientation of voter P_i by (3.1), and ii) the color difference between two neighboring parallelograms on either side of the direction (VP_i) by (3.3). Then, calculate the product of these two measures and add this product value to the location α_i in the defined histogram. Finally, $h(angle)$ which indicates the value of the histogram at the location α_i is determined as follows:

$$h(angle) = \sum_{i=1}^{N} \text{diff}(P_i)\text{diff}(R_{i1}, R_{i2})\delta[angle - \alpha_i], \tag{3.4}$$

where N is the number of voters, δ is the Kronecker delta function, and $angle = 0, 1, \ldots, 180$.

Step 2 In the histogram, exclude those angles which are smaller than 20° or larger than 160° which might cause improper detections.

After generating the histogram, the first dominant road border can be detected by

picking up an angle with maximum value in the histogram. An example of first border detection using histogram is shown in Fig. 3.6.

3.3.2 Second Border Detection and VP Updating

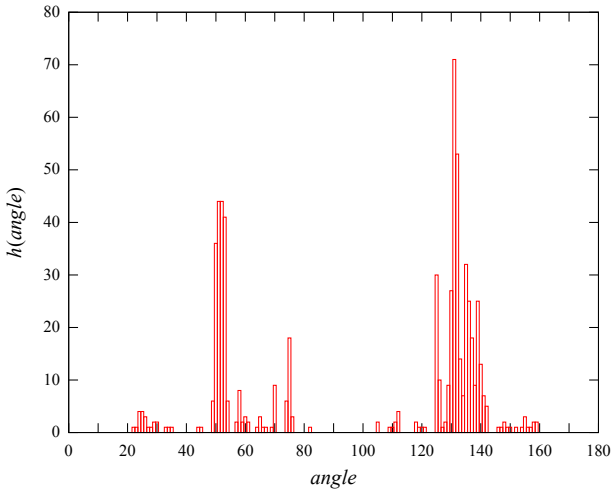
Our experimental results demonstrate that, the estimated VP coincides with a joint point of some dominant borders if this VP is a proper estimation, while it often falls on the extension of one the most dominant road borders if it is an improper estimation (see an illustration in Fig. 3.7). Therefore, we propose a method to find the second road border based on the first detected one, and update the estimated VP with a joint point of the two most dominant borders. The algorithm is performed as follows:

Step 1 Regularly sample m points ($m = 16$) on the first estimated road border and near the estimated VP. In our experiments, d_v which indicates the distance between two neighboring points is set to 2 pixels. (An illustration is shown in Fig. 3.9. In the figure, V is the estimated VP, and blue points indicate sampling points.)

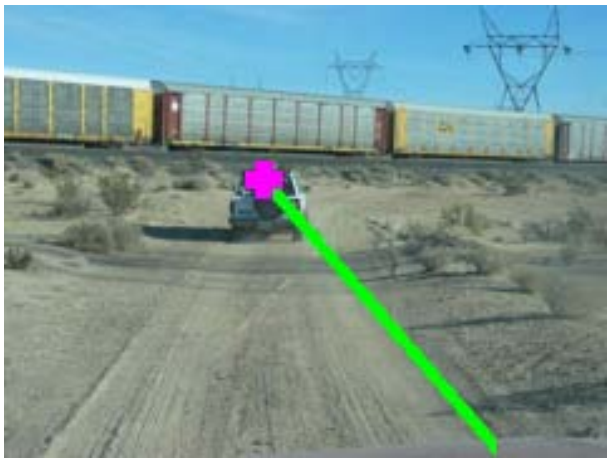
Step 2 For each point, generate a histogram as defined in 3.3.1. Exclude those angles which have an angular difference with the first border smaller than 20°. This is motivated by the assumption that the angle between two borders is generally larger than 20° (see Fig. 3.8). After that, calculate the "sum" of n maximum values ($n = 8$) in the histogram (excluding the first detected border).

Step 3 A point with "maximum sum" will be considered as an updated VP. The second border is then detected based on the generated histogram of this updated VP. In particular, the maximum value in the histogram (excluding the first detected border) of the updated VP is corresponded to the location of second border.

An example of second border detection and updated VP is shown in Fig. 3.10. In the figure, the pink cross indicates the initial estimated VP, and the orange cross indicates the updated VP. An example of voting scores of the updated VP and the first detected VP is shown in Fig. 3.11. In the figure, orange crosses indicate the updated VP, pink crosses indicate the first detected VP, and the voting scores of those points



(a) Example of generated histogram.



(b) First border detection: seclect an *angle* with maximum $h(angle)$ in the histogram.

Figure 3.6: Example of first border detection using histogram.

are written in yellow texts. From Fig. 3.11, it can be seen that, the voting scores of updated VPs are local maximum values of the voting scores of corresponding first detected VPs.

3.4 Experimental results

Our proposed road detection method is implemented in 1000 general road images. All of these images were used in the last chapter for VP detection.

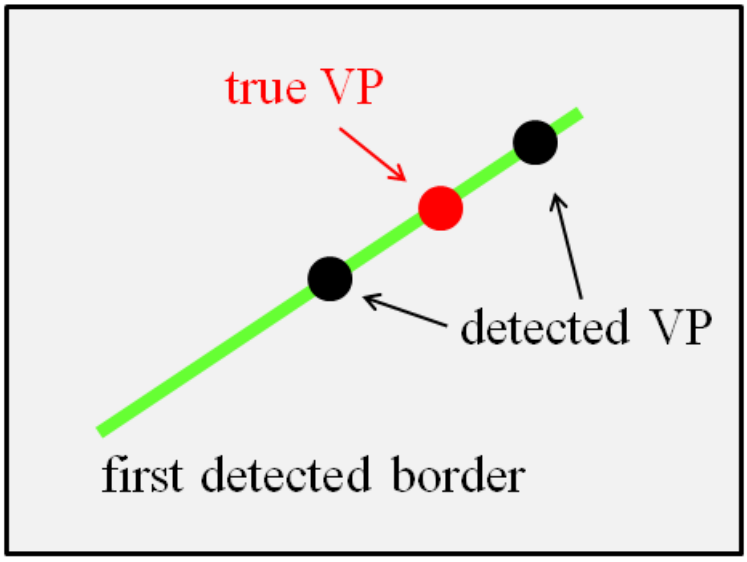


Figure 3.7: Detected VP sometime falls on the extension of one the first detected road borders.

In order to assert the accuracy of the proposed road detection method, we evaluate the performance of the proposed method in four different combinations.

For brevity, we define each combination as follows: the method with updating VPs and using color information is denoted by "update + color", the method with updating VPs and not using color information is denoted by "update + no color", the method without updating VPs but using color information is denoted by "no update + color", and the method not using color information and without updating VPs is denoted by "no color + no update". (Note that "no update" means the second border is detected together with the first border based on the two maximum values in the first generated histogram, and "no color" means the function for generating the histogram does not include the color difference calculation.)

To quantitative show the road detection accuracy, we manually mark the road region for each image of 1000 road images. The "F-measure" function which is based on the "Recall" function and "Precision" function is used to evaluate the accuracy of road detection. The F-measure function is defined as follows:

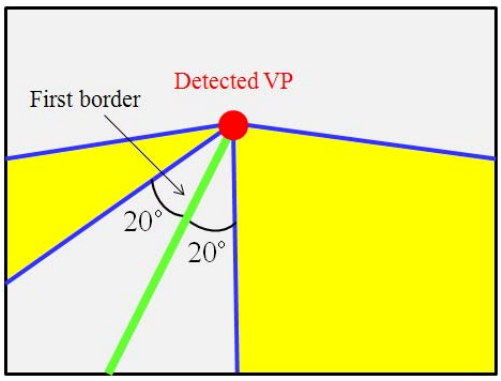


Figure 3.8: Assume that the angle between two borders is generally larger than 20°. Yellow regions: possible areas for second border detection.

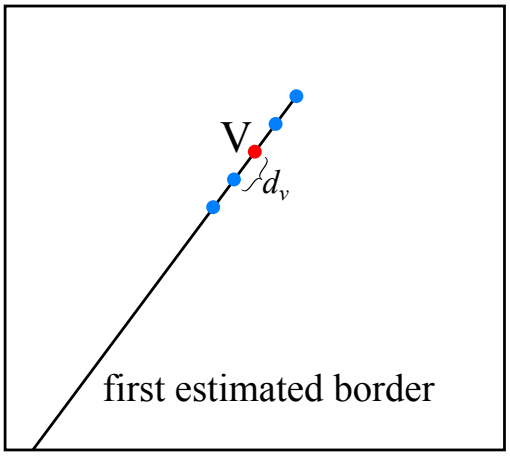


Figure 3.9: V: estimated VP, blue points: sampled points in the first border.

$$\mathrm{F} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}, \tag{3.5}$$

where R denotes the Recall function, and R is calculated as follows:

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{3.6}$$

and P denotes the Precision function, and P is calculated as follows:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{3.7}$$

Figure 3.10: Example of second border detection and updated VP.

with TP denotes the correct (True Positive) detected pixels, FN denotes the incorrect (False Negative) detected pixels of the detected road region compared with the marked ground truth region, FP denotes the False Positive pixels, and TN denotes the True Negative pixels. Figure 3.12 shows an illustration of TP, FN, FP, and TN. In the figure, the images in the first column are the input images, and the images in the second column show the ground truth of road borders (red lines) and the detected results (blue lines). The images in the third columns show the regions corresponding to TP (yellow areas), FN (red areas), FP (green areas), and TN (black areas).

Based on these definitions, we may find that the F-measure reaches its maximum value only when the detected region coincides with the marked ground truth one.

We change the F-measure rate, Recall rate, and Precision rate from 0 to 1 and calculate the statistics of how many road images are correctly detected in four different methods defined above, which are shown in Fig. 3.13, Fig. 3.14, and Fig. 3.15, respectively. In those figures, the horizontal axis represents the F-measure, Recall, and Precision, respectively, and the vertical axis represents the "accuracy" (number of images).

From Fig. 3.13, it can be seen that, the proposed method with using color information and updating VPs improves the accuracy of the algorithm over other conditions.

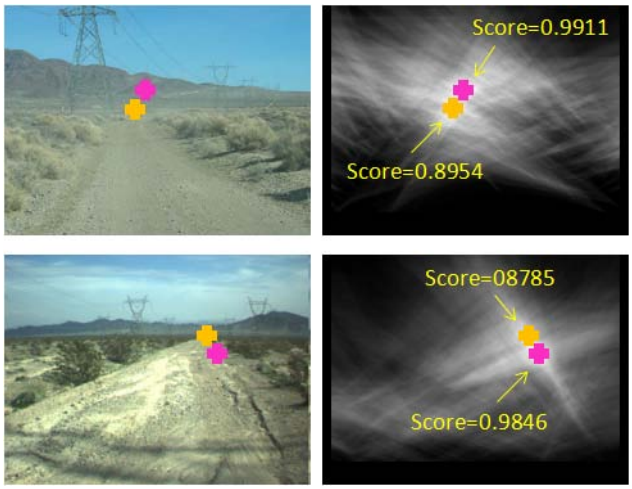


Figure 3.11: Example of normalized voting scores of the updated VP (orange crosses) and the first detected VP (pink crosses).

Table 3.1: Three extra experiments with changing d, h, and w.

	Proposed method	extra1	extra2	extra3
d	5 pixels	0 pixel	5 pixels	5 pixels
h	30 pixels	30 pixels	15 pixels	30 pixels
w	25 pixels	25 pixels	25 pixels	13 pixels

On the other hand, in order to evaluate the effect of d, h, and w which are the parameters of the parallelograms for color calculation on the estimation performance, three extra experiments with changing the values of d, h, and w are carried out as described in Table 3.1. Next, we compare the estimation performances of the proposed method and these three extra conditions.

The F-measure graph, Recall graph, and Precision graph of four methods shown in Table 3.1 are shown in Fig. 3.16, Fig. 3.17, and Fig. 3.18, respectively.

From Fig. 3.16, it can be seen that the method "extra3" is worse than the proposed method. In the other words, the parameter w is the most important parameter among three parameters d, h, and w of the parallelogram.

Figure 3.19 and Fig. 3.20 show some examples of road detection by applying the proposed method with four mentioned combinations in unstructured roads. Figure 3.21 and Fig. 3.22 show some examples of road detection by applying the proposed method

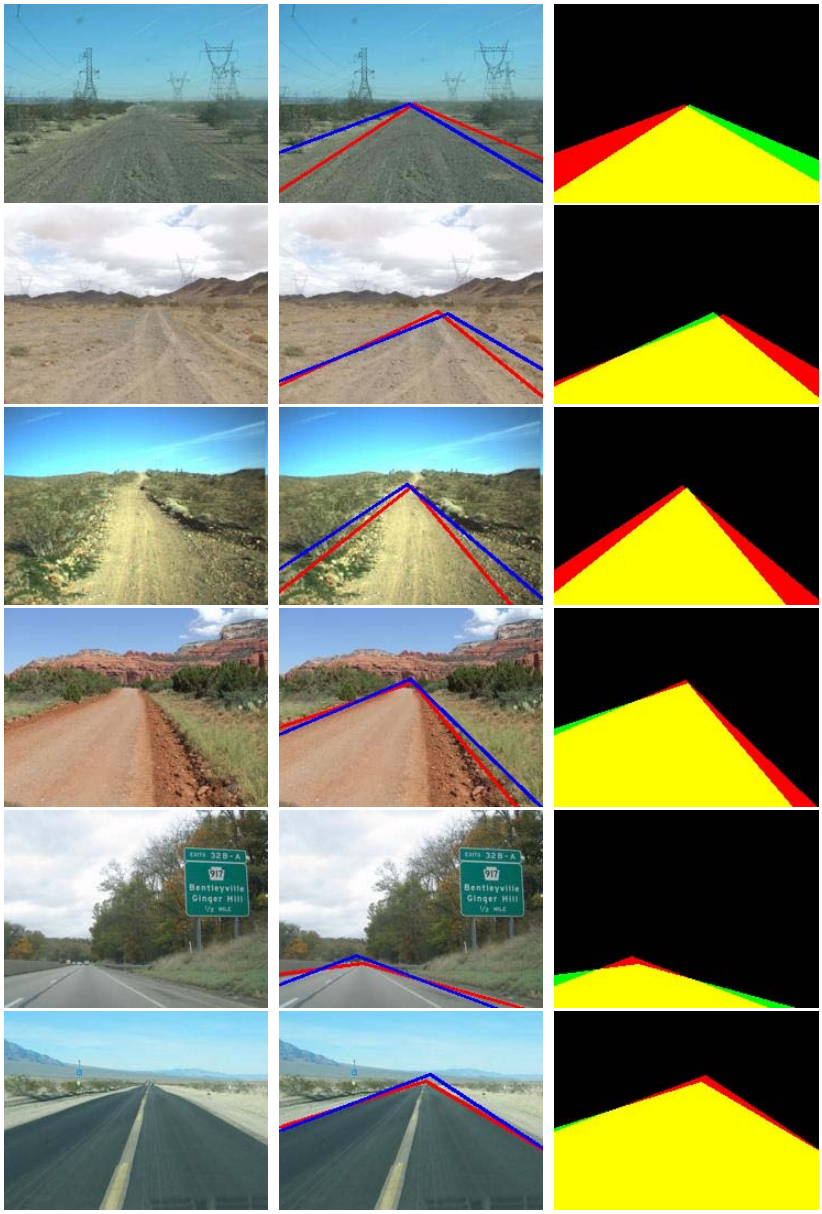


Figure 3.12: Illustration of marked ground truth of road borders and detected results. First column: input image. Second column: red lines are marked ground truth of road borders, and blue lines are detected borders. Last column: yellow regions are TP (True Positive) detected pixels, red regions are FN (False Negative) detected pixels, green regions are FP (False Positive) pixels, and black regions are TN (True Negative) pixels.

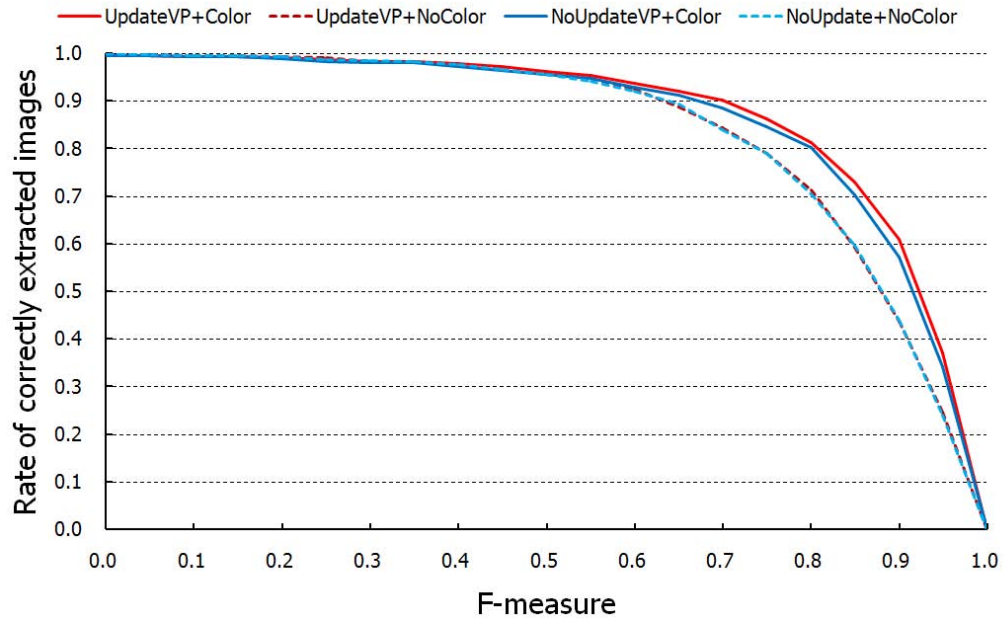


Figure 3.13: Road detection accuracy using the F-measure function: the combination of color and updating VPs improves the accuracy over other conditions.

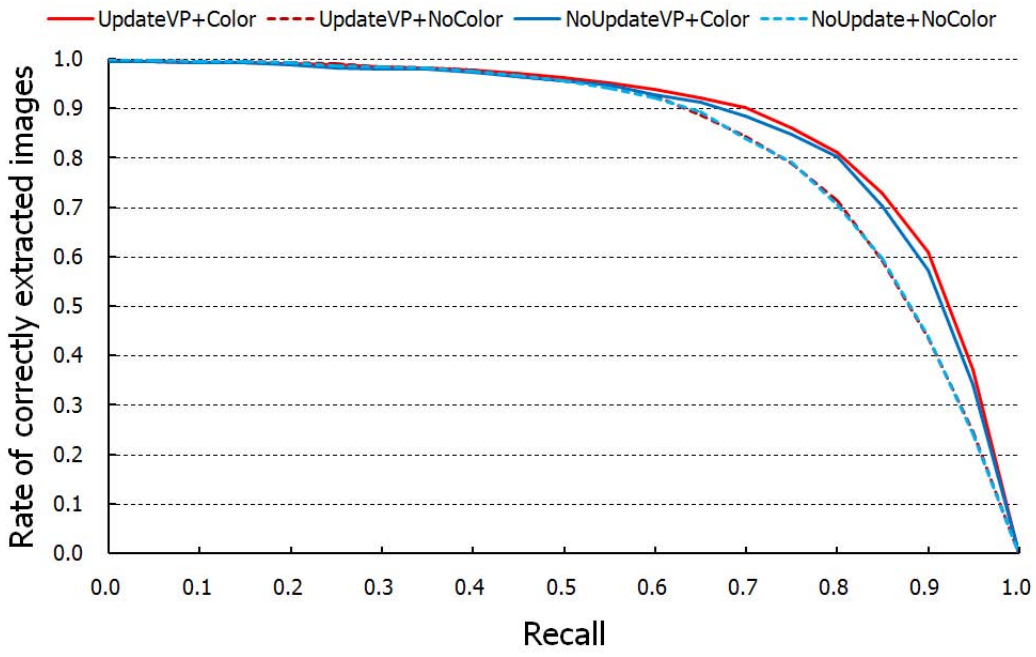


Figure 3.14: Recall graph of the four methods shown in Fig. 3.13.

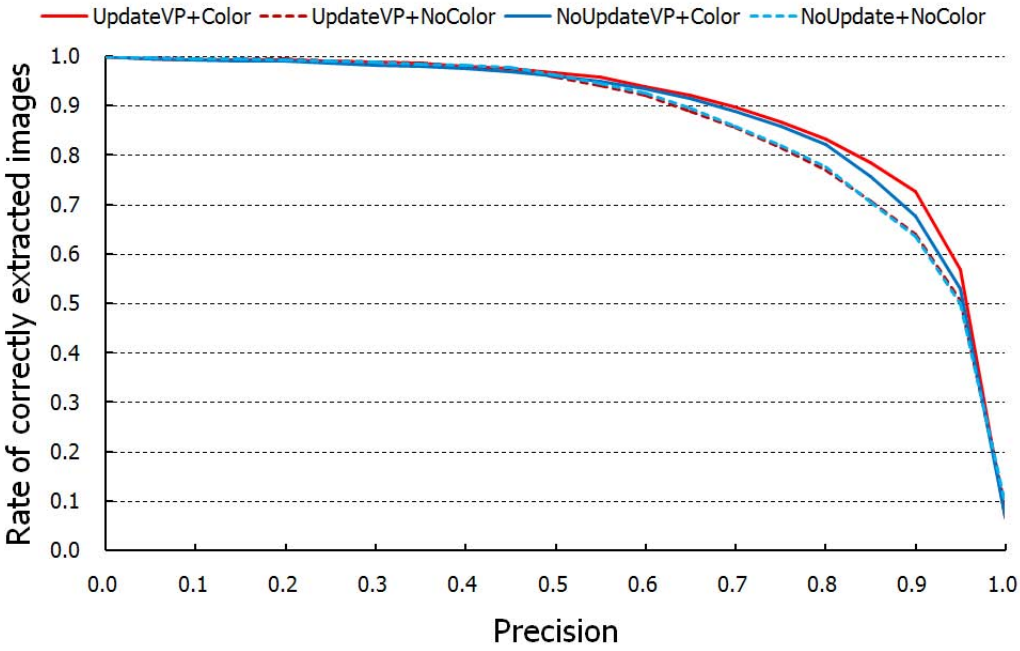


Figure 3.15: Precision graph of the four methods shown in Fig. 3.13.

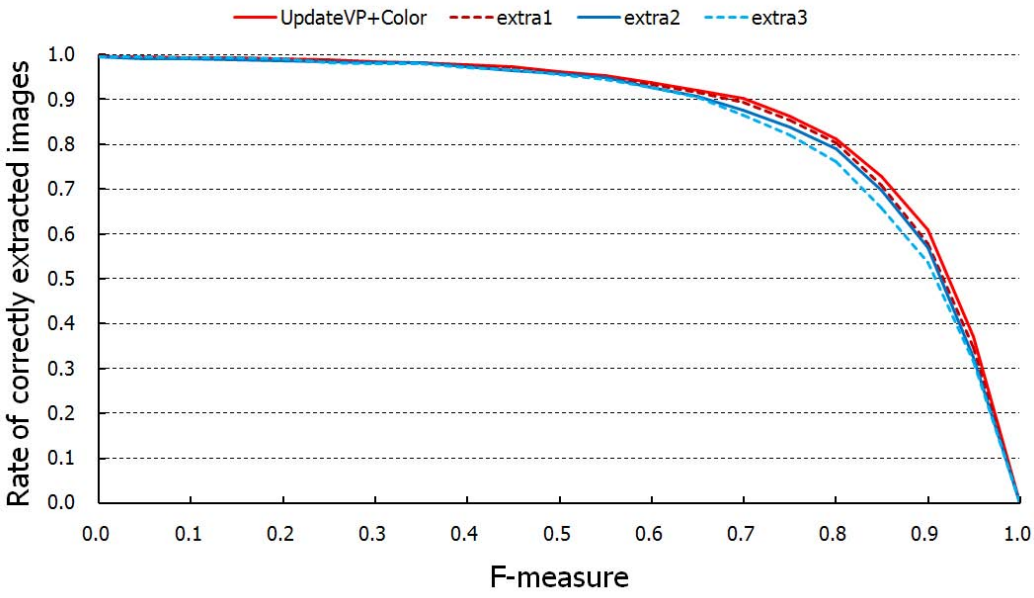


Figure 3.16: F-measure graph of four methods shown in Table 3.1.

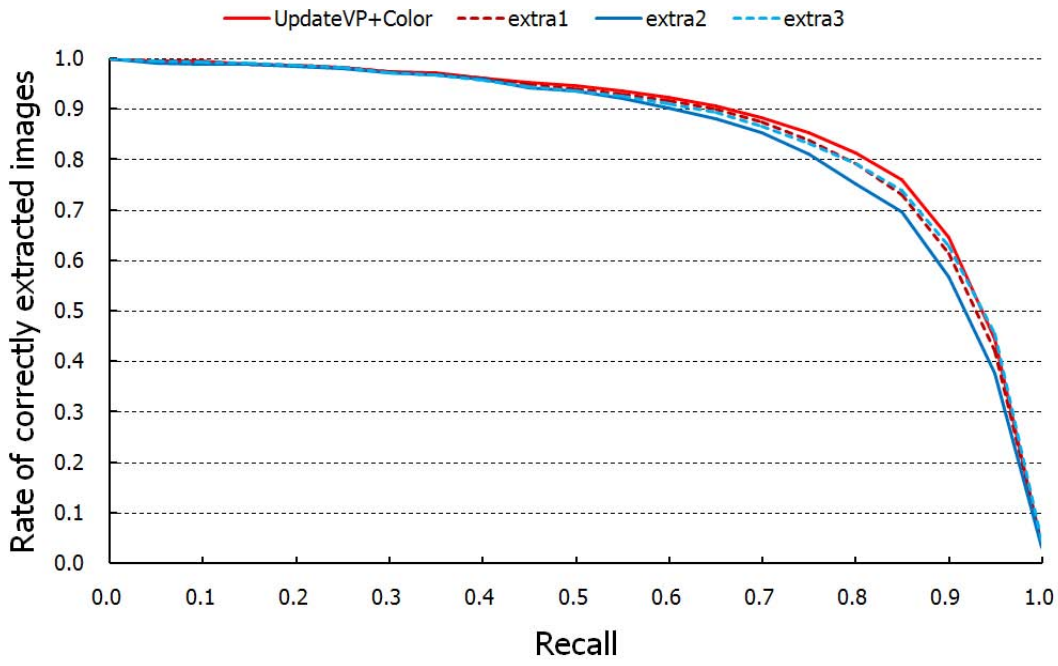


Figure 3.17: Recall graph of four methods shown in Table 3.1.

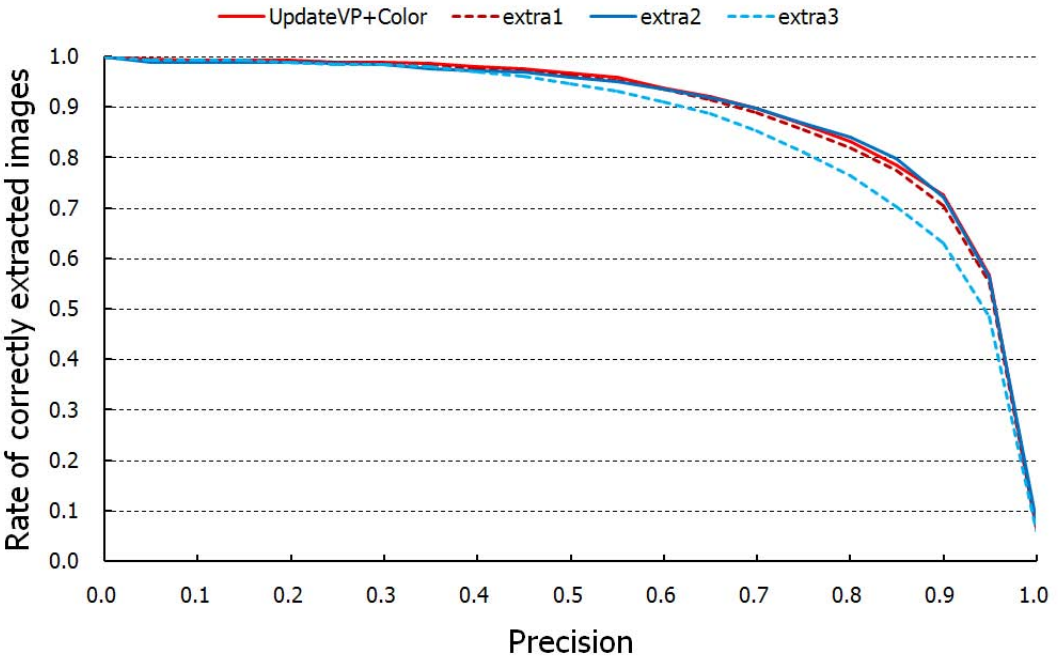


Figure 3.18: Precision graph of four methods shown in Table 3.1.

Table 3.2: Computational times for road detection process of four methods.

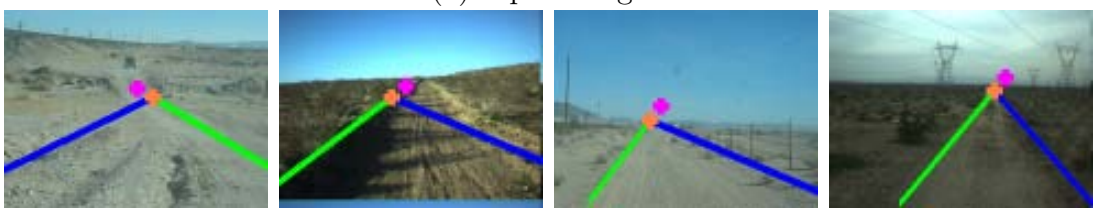
Method	Computational time (s)
update + color	3.082
update + no color	0.043
no update + color	0.201
no color + no update	0.009

with four mentioned combinations in structured roads. From those figures, it can be seen that the method "update + color" yields the best estimation performance when compared to the other three methods.

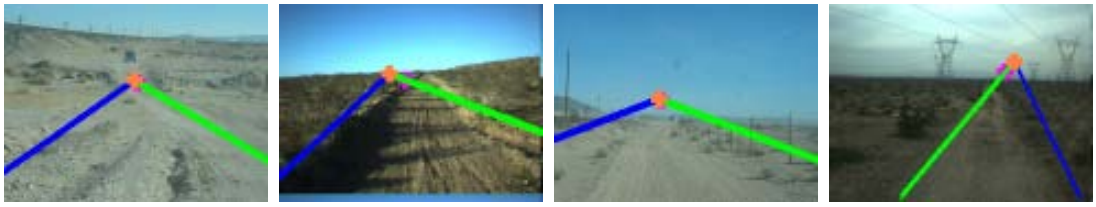
Table 3.2 shows the computational times for the road area extraction process of the four methods: "update + color", "update + no color", "no update + color", and "no color + no update" in second. From the table, it can be seen that the "update + color" method is time consuming (the computational time is 3.082 seconds). From the view point of computational time, the "no update + color" method is selected as the proposed method because its estimation performance is almost the same with the "update + color" method (see Fig. 3.13), while its computational time is much less than that of the "update + color" method (the computational time is 0.201 seconds).

Figure 3.23, Fig. 3.24, Fig. 3.25, Fig. 3.26, Fig. 3.27, and Fig. 3.28 show some examples of road area extraction for desert roads, structured roads, snow-covered roads, rural roads, dark roads, and highway roads, respectively. From those figures, it can be seen that the road area extraction method performs well for general road images.

(a) Input images.

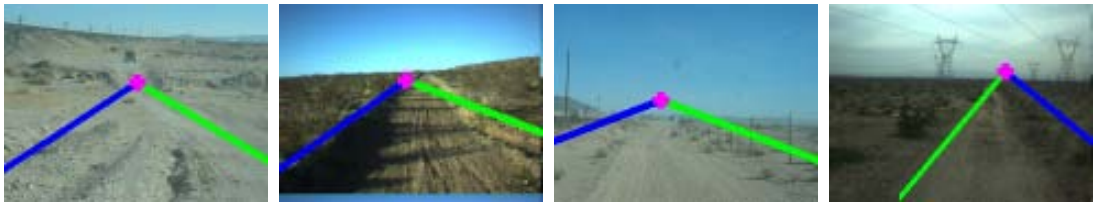


(b) update + color.



(c) update + no color.

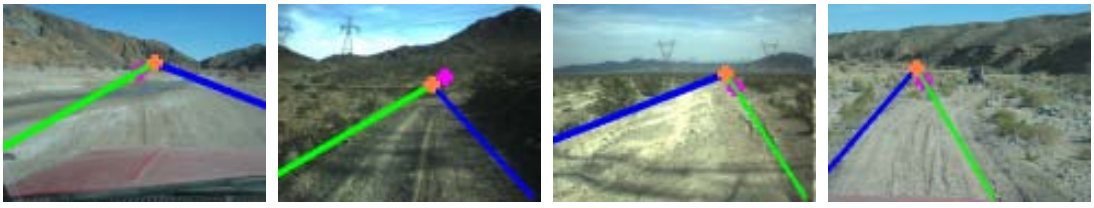
(d) no update + color.



(e) no update + no color.

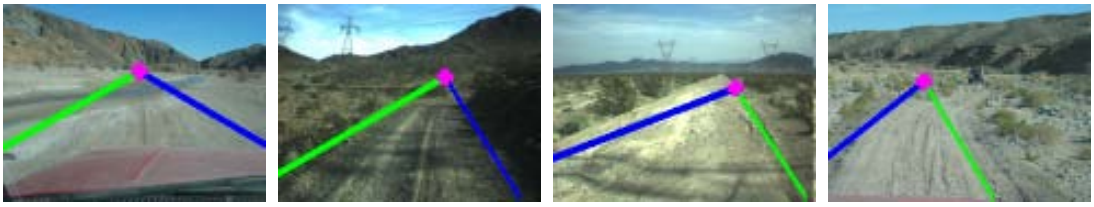
Figure 3.19: Examples of road area extraction in unstructured roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

(a) Input images.



(b) update + color.

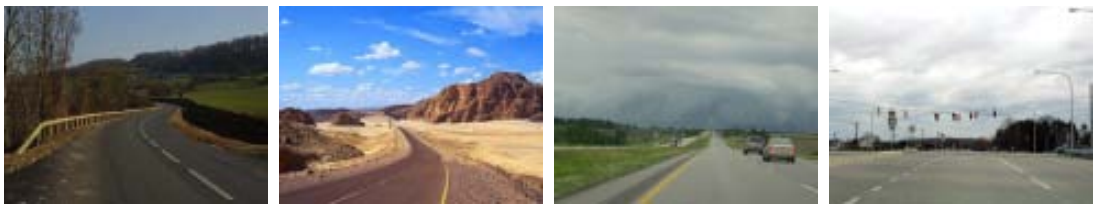
(c) update + no color.



(d) no update + color.

(e) no update + no color.

Figure 3.20: Examples of road area extraction in unstructured roads (continued).



(a) Input images.

(b) update + color.

(c) update + no color.

(d) no update + color.

(e) no update + no color.

Figure 3.21: Examples of road area extraction in structured roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

(a) Input images.

(b) update + color.

(c) update + no color.

(d) no update + color.

(e) no update + no color.

Figure 3.22: Examples of road area extraction in structured roads (continued).

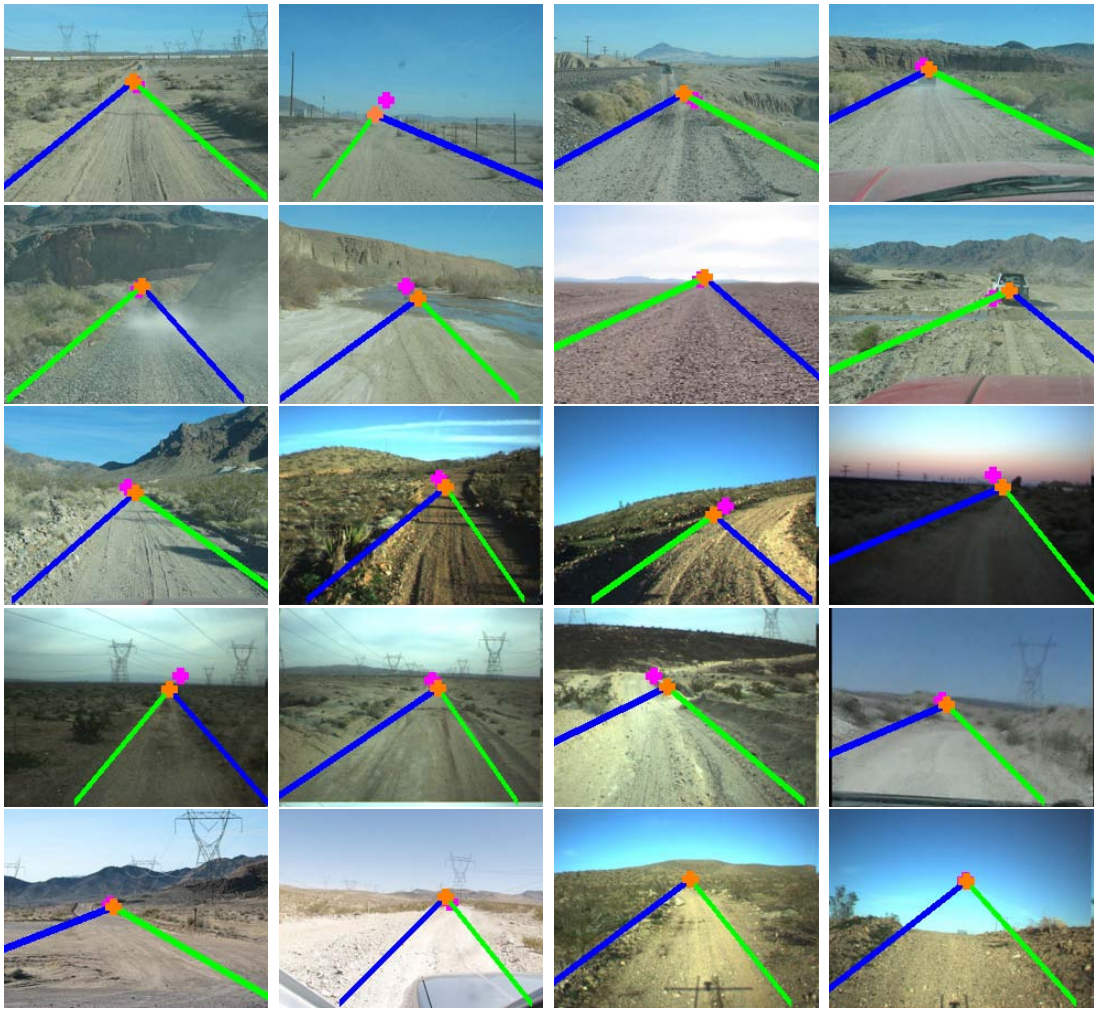


Figure 3.23: Examples of road area extraction for desert roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

Figure 3.24: Examples of road area extraction for structured roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

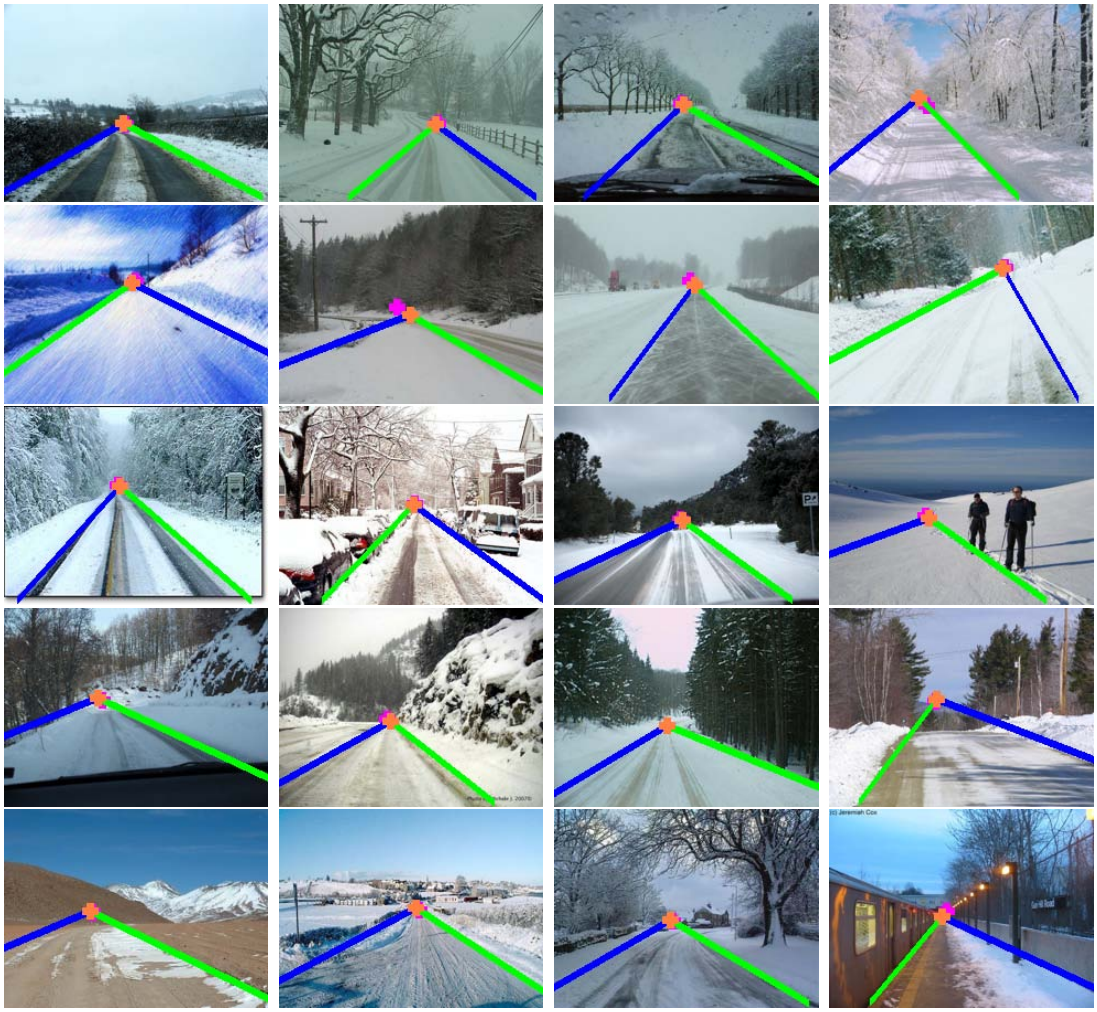


Figure 3.25: Examples of road area extraction for snow-covered roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

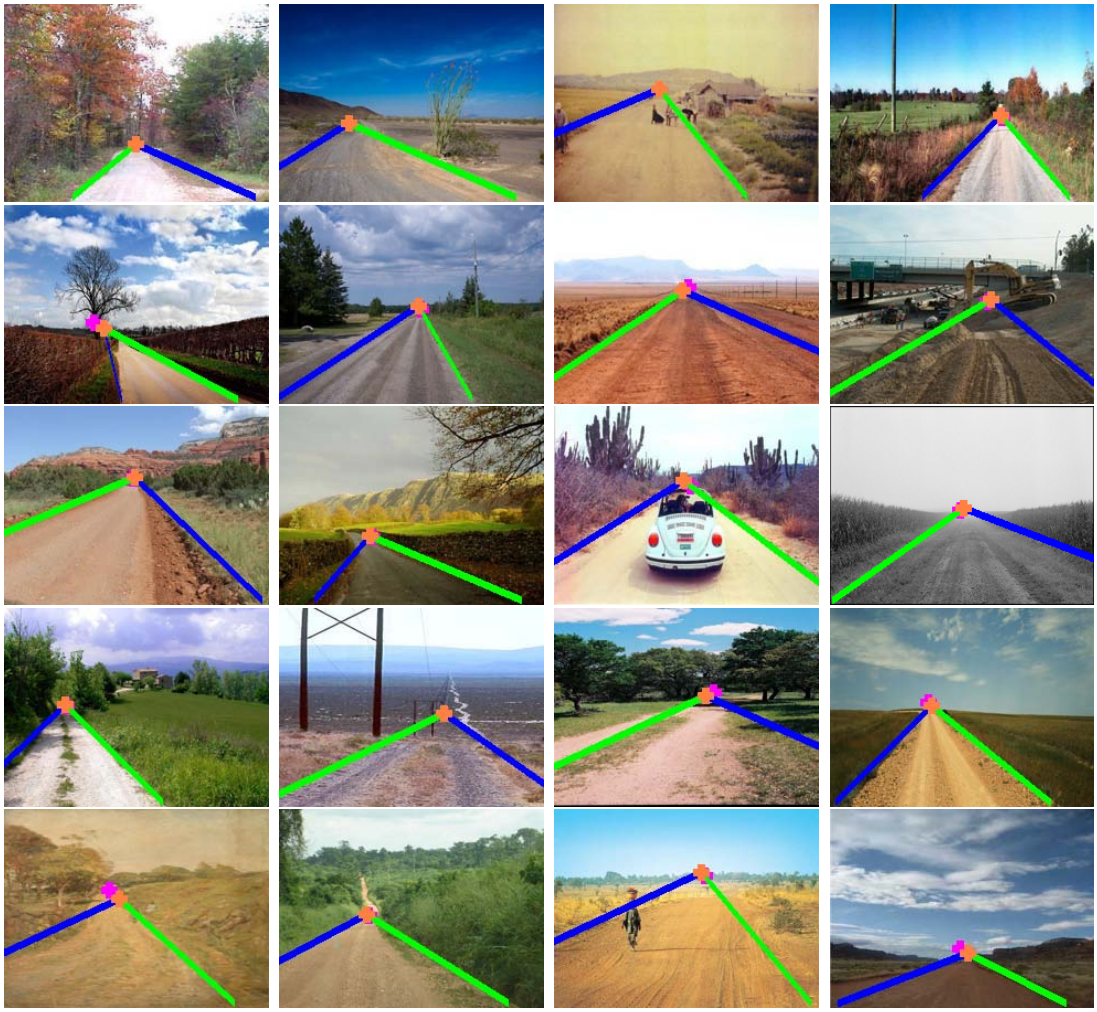


Figure 3.26: Examples of road area extraction for rural roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

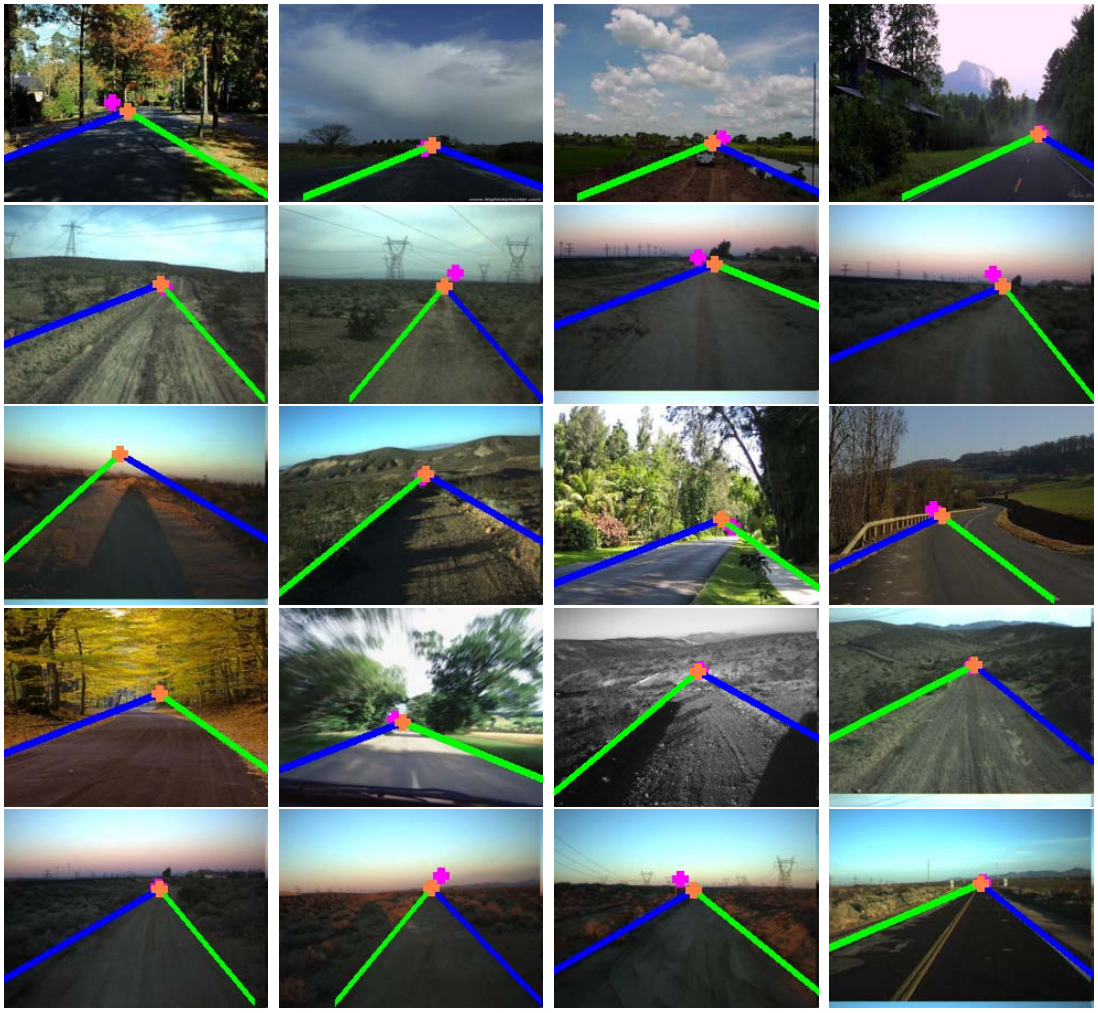


Figure 3.27: Examples of road area extraction for dark roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

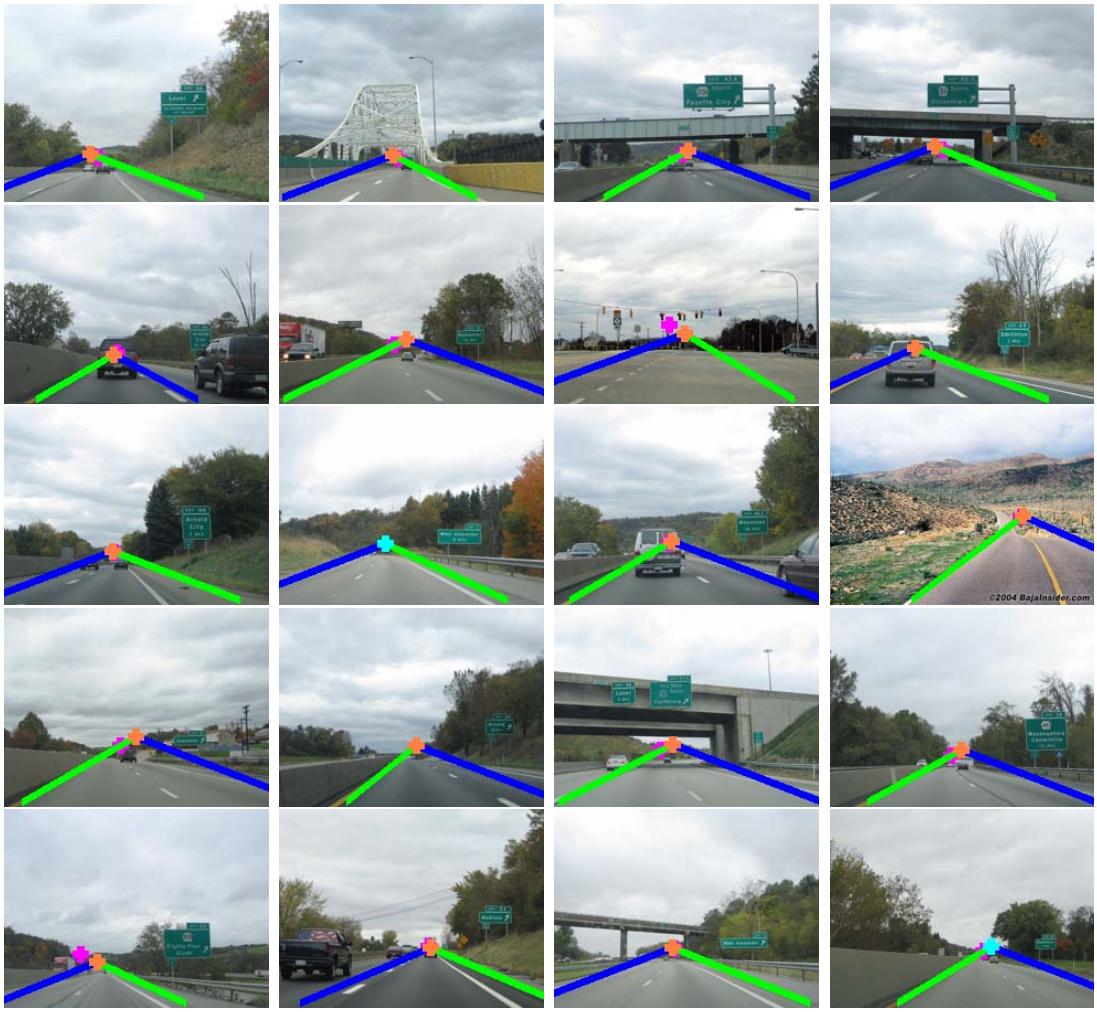


Figure 3.28: Examples of road area extraction for highway roads. Pink crosses: initial VPs, orange crosses: updated VPs, bright green lines: first detected borders, blue lines: second detected borders.

3.5 Conclusions

Detecting road area from a single road image is a challenging problem as the detection algorithm must be able to deal with continuously changing backgrounds, different environment (urban, high ways, off-road), different road types (shape, color), and different imaging conditions (varying illumination, different viewpoints and changing weather conditions).

In this chapter, a new VP-constrained road area extraction method for general road images was proposed. Our method based on a VP estimated in the previous chapter. In the proposed method, the first dominant road border can be detected by using a histogram. In order to generate the histogram, angular difference and color difference are calculated for each remaining voter. Local regions (parallelograms) for calculating color difference were introduced in this chapter. After detecting the first road border, the second most dominant road border can be detected from the first one by optimizing a criterion. The experimental results demonstrate that the proposed method which integrates texture orientations and color information is effective in detecting road regions in challenging conditions. On the other hand, from the view point of computational time, the "no update + color" method without updating VP is selected as the proposed method because its estimation performance is almost the same with the "update + color" method, while its computational time is much less than that of the "update + color" method with updating VP.

Chapter 4

Conclusions

4.1 Conclusions

Road area extraction plays an important role in different areas of computer vision such as autonomous driving, car collision warning and pedestrian crossing detection. Detecting road areas from a single road image is a challenging problem as the detection algorithm must be able to deal with continuously changing backgrounds, different environment (urban, high ways, off-road), different road types (shape, color), and different imaging conditions (varying illumination, different viewpoints and changing weather conditions).

Recently, numerous road area extraction methods have been widely published for urban and highway roads, structured roads, and unstructured roads. In all of these studies, estimating a location of vanishing point (VP) is a key requirement. A set of parallel lines in the world space by perspective projection converges to a common point in image space known as the VP. If the VP can be located correctly, then it is more likely to detect the road area properly.

State-of-the-art VP-based road detection methods can be mainly grouped into three categories: edge-based methods, prior-based methods, and texture-based methods. Most edge-based methods take advantage of computational efficiency, and they can be

applied to real time systems. However, the disadvantage is that they only work well for structured roads with clear painted lines or distinct road boundaries, while they usually fail in unstructured roads lacking sharply defined, smoothly curving edge.

In order to overcome the limitation of these edge-based methods, prior-based methods and texture-based methods for road detection have been widely proposed. Most prior-based methods are robust to varying imaging conditions, road types and surrounding environments. However, they often require a large-scale image or video training database and also manual works for labeling VPs for the training stage. Hence, such prior-based methods are inapplicable to detect the road region from a single road image.

In contrast, texture-based methods are very effective at detecting road areas for both structured roads and unstructured roads. These methods first search for local oriented textures and then make them vote for the location of the road's VP. In order to segment the road area, a VP-constrained group of dominant edges are detected, and two most dominant edges are selected as the road borders. In general, the disadvantages of these texture-based methods are: i) the computational cost of VP detection process is high, and ii) an estimation error of VP detection which may affect the performance of road area extraction is obtained in some images.

In this thesis, our main objectives are: i) to reduce the computational cost and improve the performance of VP detection algorithm, and ii) to propose a new VP-based road area extraction method from a single road image.

In Chapter 2, a new local soft voting method was proposed to overcome the limitations of the LASV method.

In Section **2.3**, Gabor filters and confidence level function were introduced. In this section, Gabor filters with five scales and 36 orientations are used to calculate the texture orientation at every pixel of the road image, and the confidence level function is used to determine the remaining voters which are useful for the VP voting process by checking the reliability of the obtained texture orientations. A larger threshold value

for confidence level function was introduced and a new algorithm based on Hough transform to discard remaining voters in very top of the image was proposed in order to reduce the number of remaining voters.

In Section **2.4**, our novel local soft voting method for VP detection was proposed, in which the number of scanning pixels is much reduced (the remaining pixels are scanned instead of the image pixels) in order to reduce the computational cost, and a new VP candidate region was introduced in order to improve the estimation accuracy of the VP detection algorithm.

The experimental results were demonstrated to show the effectiveness of the proposed method in Section **2.6**. In order to assert the effectiveness of the proposed algorithm, the proposed method and the LASV method have been implemented and tested on 1000 road images which contain large variations in color, texture, lighting condition and surrounding environment.

The experimental results reveal that: i) the proposed method outperforms better than the LASV method which uses a small voting region, especially in some images in which most remaining voters are far from the VP and ii) the computational cost of the proposed method is considerably less than that of the LASV method, the computational time for the Gabor convolution and confidence level estimation which accounts for most of the computational time in our proposed method is the same for the LASV method, whereas, the computational time of the proposed method for the voting process is much less than that of the LASV method.

In Chapter 3, our new VP-constrained road area extraction method was described.

The goal of the road extraction method is to detect the most immediate straight road part in the direction of the optical axis of the forwarding looking cameras based on estimating two lines (road boundaries) going from the VP and below the VP in the road image. In this method, in order to achieve a one degree level of precision for road boundaries detection, a histogram of 180 angles corresponding to angles of 180 lines going from the detected VP was used. When generating the histogram, the color

information of the road image is combined to improve the estimation performance. The proposed road area extraction method was implemented and tested on 1000 road images. The experimental results showed that our proposed method performs well in challenging conditions.

In summary, the main contributions of this thesis are: (1) to reduce the computational cost and improve the performance of VP detection algorithm, and (2) to propose a new VP-based road area extraction method which works well for various road types.

4.2 Future work

Our future work is to reduce the computational time for VP detection algorithm by reducing the computational time for texture orientation calculation and remaining voters determination.

Bibliography

[1] J. M. A. Alvarez. Road detection based on illuminant invariance. *IEEE Trans. on Intelligent Transportation Systems*, 12(1):184–193, 2011.

[2] Hui Kong, Jean-Yves Audibert, and Jean Ponce. Vanishing point detection for road detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 96–103, 2009.

[3] Hui Kong, Jean-Yves Audibert, and Jean Ponce. General road detection from a single image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(8):2211–2220, 2010.

[4] Nrma, motoring and services. http://www.mynrma.com.au/motoring/buy-sell/buying-advice/features/lane-departure-prevention.htm.

[5] Low power design. http://low-powerdesign.com/sleibson/2011/05/01/future-cars-the-word-from-gm-at-idc%E2%80%99s-smart-technology-world-conference/.

[6] Tsung-Ying Sun, Shang-Jeng Tsai, and V. Chan. HSI color model based lane-marking detection. In *IEEE Conference on Intelligent Transportation Systems*, pages 1168–1172, 2006.

[7] Kuo-Yu Chiu and Sheng-Fuu Lin. Lane detection using color-based segmentation. In *IEEE Intelligent Vehicles Symposium*, pages 706–711, 2005.

[8] Yinghua He, Hong Wang, and Bo Zhang. Color-based road detection in urban traffic scenes. *IEEE Trans. on Intelligent Transportation Systems*, 5(4):309–318, 2004.

[9] Ben Southall and Camillo J. Taylor. Stochastic road shape estimation. In *IEEE International Conference on Computer Vision*, pages 205–212, 2001.

[10] Bin Yu and Anil K. Jain. Lane boundary detection using a multiresolution hough transform. In *IEEE International Conference on Image Processing*, pages 748–751, 1997.

[11] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[12] Joel C. McCall and Mohan M Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Trans. on Intelligent Transportation Systems*, 7(1):20–37, 2006.

[13] Claudio Rosito Jung and Christian Roberto Kelber. A robust linear-parabolic model for lane following. In *17th IEEE Brazilian Symposium on Computer Graphics and Image Processing*, pages 72–79, 2004.

[14] Yue Wang, Ean Khwang Teoh, and Dinggang Shen. Lane detection and tracking using B-Snake. *Image and Vision Computing*, 22(4):269–280, 2004.

[15] Jan Sparbert, Klaus Dietmayer, and Daniel Streller. Lane detection and street type classification using laser range images. In *IEEE International Conference on Intelligent Transportation Systems*, pages 454–459, 2001.

[16] Bing Ma, Sridhar Lakshmanan, and Alfred O. Hero. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Trans. on Intelligent Transportation Systems*, 1(3):135–147, 2000.

[17] M. Bertozzi and A. Broggi. GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. on Image Processing*, 7(1):62–81, 1998.

[18] Yaniv Alon, Andras Ferencz, and Amnon Shashua. Off-road path following using region classification and geometric projection constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 689–696, 2006.

[19] Alberto Broggi, Claudio Caraffi, Rean Isabella Fedriga, and Paolo Grisleri. Obstacle detection with stereo vision for off-road vehicle navigation. In *IEEE Society Conference on Computer Vision and Pattern Recognition - CVPR Workshops*, page 65, 2005.

[20] Christopher Rasmussen. Texture-based vanishing point voting for road shape estimation. In *British Machine Vision Conference*, pages 7.1–7.10, 2009.

[21] Christopher Rasmussen. Grouping dominant orientations for ill-structured road following. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 470–477, 2004.

[22] Marcos Nieto and Luis Salgado. Real-time vanishing point estimation in road sequences using adaptive steerable filter banks. In *9th International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 840–848, 2007.

[23] Qi Wu, Wende Zhang, Tsuhan Chen, and B.V.K.V. Kumar. Prior-based vanishing point estimation through global perspective structure matching. In *IEEE International Conference on Acoustics Speech and Signal Processing*, pages 2110–2113, 2010.

[24] Geng Zhang, Nanning Zheng, Chao Cui, Yuzhen Yan, and Zejian Yuan. An efficient road detection method in noisy urban environment. In *IEEE Intelligent Vehicles Symposium*, pages 556–561, 2009.

[25] T. Suttorp and T. Bucher. Robust vanishing point estimation for driver assistance. In *IEEE Conference on Intelligent Transportation Systems*, pages 1550–1555, 2006.

[26] J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *IEEE 12th International Conference on Computer Vision*, pages 1250–1257, 2009.

[27] Yan Wang, Li Bai, and M. Fairhurst. Robust road modeling and tracking using condensation. *IEEE Trans. on Intelligent Transportation Systems*, 9(4):570–579, 2008.

[28] Jise M. Alvarez, Theo Gevers, and Antonio M. Lopez. 3D scene priors for road detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 57–64, 2010.

[29] Nicholas Apostoloff and Alexander Zelinsky. Robust vision based lane tracking using multiple cues and particle filtering. In *IEEE Intelligent Vehicles Symposium*, pages 558–563, 2003.

[30] Camillo J. Taylor, Jitendra Malik, and Joseph Weber. A real-time approach to stereopsis and lane-finding. In *IEEE Intelligent Vehicles Symposium*, pages 207–212, 1996.

[31] Xiqun Lu. New efficient vanishing point detection from a single road image based on intrinsic line orientation and color texture properties. *Trans. on Optical Engineering*, 51(3):037001/1–037001/14, 2012.

[32] Peyman Moghadam, Janusz A. Starzyk, and W. S. Wijesoma. Fast vanishing point detection in unstructured environments. *IEEE Trans. on Image Processing*, 21(1):425–430, 2012.

[33] John Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[34] Antonio Torralba and Pawan Sinha. Statistical context priming for object detection. In *IEEE International Conference on Computer Vision*, volume 1, pages 763–770, 2001.

[35] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[36] Jise M. Alvarez, Theo Gevers, and Antonio M. Lopez. Vision-based road detection using road models. In *16th IEEE International Conference on Image Processing*, pages 2073–2076, 2009.

[37] Tai Subg Lee. Image representation using 2D gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[38] XiaoGuang Feng and Peyman Milanfar. Multiscale principal components analysis for image local orientation estimation. In *36th Asilomar Conference on Signals, Systems and Computers*, pages 478–482, 2002.

[39] Trung Hieu Bui and Eitaku Nobuyama. A local voting method by scanning high-confidence level voters for vanishing point detection. In *International Conference on Instrumentation, Control, Information Technology and System Integration*, pages 790–795, 2011.

[40] Trung Hieu Bui and Eitaku Nobuyama. A local soft voting method for texture-based vanishing point detection from unstructured road images. In *International*

Conference on Instrumentation, Control, Information Technology and System Integration, pages 396–401, 2012.

[41] Robert G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

[42] S. Se and M. Brady. Vision-based detection of staircases. In *IEEE Asian Conference on Computer Vision*, pages 535–540, 2000.

[43] S. Se. Zebra-crossing detection for the partially sighted. In *Proceedings of Computer Vision and Pattern Recognition (CVPR) Conference*, pages 211–217, 2000.

[44] N. Simond and P. Rives. Homography from a vanishing point in urban scenes. In *Proceedings of Intelligent Robots and Systems Conference*, pages 1005–1010, 2000.

[45] A. Polk and R. Jain. A parallel architecture for curvaturebased road scene classication. In *Book Vision-based vehicle guidance*, pages 284–299, 1992.

[46] B. Binu. Road detection using a new method for vanishing point detection based on hough transform. In *Proceedings of 4th International Conference of Global Trends in Computing and Communication Systems*, pages 583–587, 2012.

[47] C. Rasmussen. Roadcompass: following rural roads with vision + ladar using vanishing point tracking. *IEEE Trans. on Autonomous Robots*, 25(3):205–229, 2008.

[48] C. Rasmussen and T. Korah. On-vehicle and aerial texture analysis for vision-based desert road following. In *IEEE International Workshop on Machine Vision for Intelligent Vehicles*, page 66, 2002.

[49] Trung Hieu Bui, Eitaku Nobuyama, and Takeshi Saitoh. Road detection based on an estimated vanishing point and color information. In *International Conference on Computer Vision, Image and Signal Processing*, pages 1748–1753, 2012.

[50] Trung Hieu Bui, Eitaku Nobuyama, and Takeshi Saitoh. Road boundaries extraction for general road images using texture orientation and color information. In *Proceedings of SICE Kyushu Branch Annual Conference*, pages 175–178, 2012.

Publication

- **Journal**

  1. Trung Hieu BUI, Eitaku NOBUYAMA, and Takeshi SAITOH,
     "A Texture-based Local Soft Voting Method for Vanishing Point Detection from a Single Road Image",
     The IEICE Transactions on Information and Systems (Vol.E96-D, No.3, Mar.2013).

- **Proceedings**

  1. Trung Hieu BUI, Eitaku NOBUYAMA,
     "A Local Voting Method by Scanning High-Confidence Level Voters for Vanishing Point Detection",
     Proceedings of the 50^{th} Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE2011),
     pp.790–795, Tokyo, Japan (2011.9) (published).
  2. Trung Hieu BUI, Eitaku NOBUYAMA,
     "A Local Soft Voting Method for Texture-Based Vanishing Point Detection from Unstructured Road Images",
     Proceedings of the 51^{st} Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE2012),
     pp.396–401, Akita, Japan (2012.8) (published).
  3. Trung Hieu BUI, Eitaku NOBUYAMA, and Takeshi SAITOH,
     "Road Detection Based on an Estimated Vanishing Point and Color Information",
     Proceedings of the 36^{th} International Conference on Computer Vision, Image and Signal Processing (ICCVISP2012),
     pp.1748–1753, Paris, France (2012.11) (published).

4. Trung Hieu BUI, Eitaku NOBUYAMA, and Takeshi SAITOH,
"Road Boundaries Extraction for General Road Images Using Texture Orientation and Color Information",
Proceedings of SICE Kyushu Branch Annual Conference 2012,
pp.175–178, Kumamoto, Japan (2012.11) (published).