

國立交通大學

生醫工程研究所

碩士論文

以分界特徵限制為基礎之即時

道路邊緣追蹤演算法

A New Method of Efficient Road Following
Algorithm Based on Temporal Region Ratio and
Edge Constraint

研究生：陳宣輯

指導教授：林進燈 博士

中華民國 九十八 年 六 月

以分界特徵限制為基礎之即時道路邊緣追蹤演算法
A New Method of Efficient Road Boundary Tracking Algorithm
Based on Temporal Region Ratio and Edge Constraint

研 究 生：陳宣輯

Student：Shen-Chi Chen

指導教授：林進燈 博士

Advisor：Dr. Chin-Teng Lin

國立交通大學

資訊工程學院

生醫工程研究所

碩士論文

A Thesis

Submitted to Institute of Biomedical Engineering

College of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Computer Science

June 2009

Hsinchu, Taiwan

中華民國 九十八 年 六 月

以分界特徵限制為基礎之即時

道路邊緣追蹤演算法

學生：陳宣輯

指導教授：林進燈 博士

國立交通大學資訊工程學院生醫工程研究所

中文摘要

近年來隨著車輛數目的增加與性能不斷的成長，道路情況與駕車難度也相對的提升，當今的車輛仍需要駕駛者全神貫注地人為操作，相對地增加駕駛人的壓力與車禍發生的可能性。隨著行車安全愈來愈受到人們重視，為了改善此問題，世界各國的研究機構、知名車廠、學術研究室均投入大量的精力於智慧型運輸系統 ITS(Intelligent Transportation System) 中的安全駕駛輔助系統，並得到許多人的期待與讚許。也因此這類的先進式車輛控制系統的願景，以從當初的駕駛者安全輔助性質轉為主動式的安全操縱系統，甚至最終的目標已經設定為自動駕駛。行駛道路邊緣的偵測為其中一項關鍵的技術，本論文將以 CCD (charge coupled device)攝影機所擷取的影像為基礎利用本論文提出的邊緣特徵穩定地實現道路邊緣偵測與追蹤的演算法。

本研究所提出的道路邊緣特徵結合了邊緣強度與道路顏色兩項重要的特徵，改進了當今許多研究只利用其中一類可能會產生的問題。我們將在論文中說明如何利用所提出的創新演算法修正其他文獻的問題。本研究的演算法利用有效的特徵，增加系統之適應性，可處理多類型的道路邊緣，包括一般市區道路、高速公路、等有明顯車道線的邊緣；另外，由於我們所採用的邊界特徵擁有突出的穩定性與一致性；因此本系統對於一些鄉間小路、產業道路..等，沒有明顯的標線邊緣，與夜間道路仍能提供有效性地偵測結果。本系統在設計時在考慮許多駕駛狀況，因此，系統具有處理一般駕駛狀況與道路環境的改變的能力，如變換車

道，左右偏動，與不平坦道路，造成相機的晃動。本篇論文提出一主車道邊緣偵測與追蹤系統，其具有先進的適應性與即時的處理效率。

A New Method of Efficient Road Boundary Tracking Algorithm Based on Temporal Region Ratio and Edge Constraint

Student: Shen-Chi Chen

Advisor: Dr. Chin-Teng Lin

Institute of Biomedical Engineering
College of Computer Science
National Chiao Tung University

Abstract

In recent years, since the number and properties of the vehicle are continuous increasing, the complexity of road condition and the difficulties of operation increase relatively. As of today, physically driving not only requires drivers' full attention but could potentially amplify drivers' stress and possibility of car accidents. As people become more concerned about the safety in driving, many research institutes around the world, well-known automobile manufactures, and academic research labs invest a tremendous amount of effort to develop a safe driving assistance system of **ITS (Intelligent Transportation System)** to alleviate the problems of physical driving. With many expectation and applaud, the advanced supporting system has transformed into an active safe operating system, and ultimately will develop into an automatic driving system. Since the driving path boundary detection is one of the key techniques of automatic driving system, this paper is based on the use of images captured by the CCD (charge coupled device) camera and we discovered the general boundary features to steadily achieve main driving path boundary detection and tracking

system.

We purpose to integrate two major features, edge intensity and color distribution, of road boundary and mitigate the problems encountered by many other studies which utilize only one major feature for road boundary detection. In this paper we will explain how to make use of the advanced algorithms to amend the problems in other literatures. Our algorithms utilize effective features to enhance the adaptability of the system; the system is able to manage various types of road boundary includes general urban road, highway, or any boundary with the clear lane line. In addition, we use the general characteristics of the boundary, and therefore the system is still able to effectively provide detection result at the road without clear boundary such as the countryside paths, the road with no lane line, and the scene at the night. In the design of the system we take into account many driving situations, such as changing lanes, left and right side move, and non-flat roads lead to camera shake. Therefore, the system possesses good ability to deal with general driving conditions and variations. This paper presents a system which includes main driving path boundary detection and boundary tracking with advanced real-time adaptability and efficiency.

致 謝

本論文的完成，首先要感謝指導教授林進燈博士與蒲鶴章博士、范剛維博士這兩年來的悉心指導，讓我學習到許多寶貴的知識，在學業及研究方法上也受益良多。另外也要感謝口試委員們的建議與指教，使得本論文更為完整。

其次，感謝超視覺驗室的子貴、建霆、琳達、肇廷、明峰與東霖所有的學長與學姐細心地指導，與同學 哲銓、鎮宇與耿維的相互砥礪，一起奮鬥一起歡樂，也感謝學弟們在研究過程中所給我的鼓勵與協助。其中我的大貴人子貴學長，在理論及程式技巧上給予我相當多的幫助與建議，讓我獲益良多。另外也感謝學弟多次辛苦地幫助我拍攝行車影片，建立了許多寶貴的資料庫。這兩年讓我在學問與處事經驗成長很多，感恩之情，永藏於心，希望未來仍有機會為實驗室貢獻自己小小的力量。

感謝我的父母親對我的教育與栽培，並給予我精神及物質上的一切支援，使我能安心地致力於學業。此外也感謝妹妹對我不斷的關心與鼓勵。

謹以本論文獻給我的家人及所有關心我的師長與朋友們，我永遠愛你們。

Contents

Chinese Abstract	ii
English Abstract	iv
Chinese Acknowledgements	vi
Contents	vii
List of Tables.....	ix
List of Figures.....	x
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Motivation.....	4
1.3 Objective	5
1.4 Organization.....	6
Chapter 2 Relate Works	7
2.1 Overview.....	7
2.2 Approaches Introduction.....	7
Chapter 3 Proposed Techniques	21
3.1 Objective	21
3.2 On-lined L*a*b color model.....	21
3.2.1 Objective.....	21
3.2.2 Color space selection.....	22
3.2.3 On-lined Model.....	26
3.3 LMR boundary window	36
3.3.1 Overview	36
3.3.2 LMR boundary window introduction.....	37
3.3.3 Boundary feature of LMR boundary window.....	38
Chapter 4 System Algorithm.....	42
4.1 Overview.....	42
4.2 Driving Path Boundary Detection.....	43
4.2.1 Objective.....	43
4.2.2 Methods.....	44
4.3 Boundary Tracking Algorithm	51
4.3.1 Objective.....	51

4.3.2 The Alignment of LMR Boundary Window	53
Chapter 5 Experimental Results.....	67
5.1 Experimental Results of Driving Path Boundary Detection	67
5.2 Experimental Results of Boundary Tracking.....	69
Chapter 6 Conclusion	73
6.1 Contribution	74
6.2 Future Works.....	76
References.....	77

List of Tables

Table 1 : Traffic accident and violation in Taiwan from 2002 to 2006.....	2
Table 2 : Related research organization of intelligent transportation system	2

List of Figures

Fig. 1-1: The Blind Spot Information System (BLIS) [Volvo]	3
Fig. 1-2: Adaptive Cruise Control System (ACC)	3
Fig. 1-3: Volvo's City Safety System [Volvo]	3
Fig. 1-4: The most popular sensors are used in driver assistant systems	5
Fig. 2-1: The most popular features.	8
Fig. 2-2: The disadvantages of common features.	9
Fig. 2-3: Detection result by region based approach and our desired results.	10
Fig. 2-4: Instability of the region-based methods without adaptive module.	10
Fig. 2-5: The feature and common disadvantages of the region based methods.	13
Fig. 2-6: The method using bird-viewed image by IPM transform	14
Fig. 2-7: The difficult reasons of boundary based approach	14
Fig. 2-8: Points transfer to (θ, γ) plane.	16
Fig. 2-9: Extract the boundaries with maximal accumulator values	16
Fig. 2-10: Edge feature encounter difficult problems	17
Fig. 2-11: The feature and common disadvantages of the boundary based methods ..	17
Fig. 2-12: Spline model to describe the boundary by the control points $P_1 P_2 P_3$	18
Fig. 2-13: The position X_m divides the scene into far field and near field.	19
Fig. 2-14: The lane-curve function (LCF) and its asymptotes	19
Fig. 2-15: The feature and common disadvantages of the model based methods	20
Fig. 3-1: The max-min value of L^*a^*b values	24
Fig. 3-2: The comparison results using L^*a^*b color space and HSV color space	25
Fig. 3-3: On-lined learning procedure.	27
Fig. 3-4: Guide snake keep tracking the curvature of the road	28
Fig. 3-5: LMR boundary window is set as ROI s. Guide Snake make LMR boundary window have its exclusive sampling area	28
Fig. 3-6: Blob noises appear in experimental videos	32
Fig. 3-7: Filtering the blob noises.	32
Fig. 3-8: A color ball i in the L^*a^*b color model	33
Fig. 3-9: Each color ball hold a weight record	34
Fig. 3-10: The pixel matched with first B weight color balls as the standard color.	36
Fig. 3-11: Introduction LMR boundary window	38
Fig. 3-12: The characteristic of the boundary	39
Fig. 3-13: Guide snake set the nearest area as exclusive sampling	41
Fig. 3-14: The boundary resides on the one of the block in LMR boundary cause the maximum boundary energy on that block.	41
Fig. 4-1: The system architecture of the algorithm	42

Fig. 4-2: There are several common challenges in boundary detection.....	43
Fig. 4-3: The flow chart of Driving Path Boundary Detection	44
Fig. 4-4: The procedure of edge-based approach.....	45
Fig. 4-5: Comparison between with de-noise and without it	45
Fig. 4-6: Gaussian filter to de-noise.....	46
Fig. 4-7: The result of each step of the edge-based approach.....	48
Fig. 4-8: The problem of boundary detection by edge feature method.....	49
Fig. 4-9: Modified boundary detection methods.....	50
Fig. 4-10: The result of the modified boundary detection methods.....	51
Fig. 4-11: Primary overview of boundary tracking.....	52
Fig. 4-12: The boundary tracking algorithm in the system architecture.....	52
Fig. 4-13: The flow chart of alignment of LMR boundary window.	53
Fig. 4-14: The boundary features appear or disappear in LMR boundary windows. ..	54
Fig. 4-15: Checking the accumulator window to process the boundary alignment (optimization). Alternatively, it will process boundary detection (searching).....	55
Fig. 4-16: The steps of boundary alignment (optimization)	58
Fig. 4-17: Checking the accumulator window and then Performs the boundary detection procedure.....	59
Fig. 4-18: (a): The boundary is broken line so it detects no boundary appear in the searching range. (b): The LMR boundary will remain at original position and wait for the incoming segment of the broken line.	61
Fig. 4-19: (a): The boundary have deviated the range of LMR boundary window, so the accumulator window detected no boundary trace. (b): The LMR boundary tracks the last boundary position in the searching range and aligns to it.	62
Fig. 4-20: (a): L block with the maximum boundary; (b): LMR boundary move toward left side to align to boundary. The correct alignment regress the region ratio, both RoadPower and NonRoadPower , to previous effective range.	64
Fig. 4-21: (a): The boundary alignment to left is in error because the fake edge resulted from the prominent shadow. (b): After the wrong alignment, the incorrect position of LMR boundary window can be checked by the NonRoadPower of Region Ratio Feature shown in green block.	65
Fig. 4-22: (a): The boundary alignment to right is in error because the fake edge resulted from the object outside path. (b): After the wrong alignment, the incorrect position of LMR boundary window can be checked by the RoadPower of Region Ratio Feature shown in green block.	66
Fig. 5-1: The tested vehicle.....	67
Fig. 5-2: Experimental results of driving boundary detection.	69
Fig. 5-3: Experimental results of boundary tracking	71

Fig. 5-4: Experimental results of change line situation.	72
Fig. 6-1: The main features and disadvantages of region-based methods. Our algorithm can modify and solve the problems.	75
Fig. 6-2: The main features and disadvantages of boundary-based methods. Our algorithm can modify and solve the problems.	75

Chapter 1

Introduction

1.1 Background

In recent year, the number of higher performance vehicle is growing up and causes more problem of traffic safety. As seen in Table 1, the traffic safety is worse and worse in the statistics about the road traffic accident and violation in Taiwan from 2002 to 2009. Therefore, Intelligent Transportation System (ITS) has been preceded for many years. Intelligent conveyance system ITS (intelligent transportation system) integrates with the advance technology of application in many field including computer, electronics, communication science etc. It promotes transportation system of security, efficiency, comfort, and reduces the confusion of road environment.

Duo to the reason of safety requirement of consumers, it attracts a lot of car manufactories, research institution and laboratories to engage in this research area. In this new generation, the intelligent transportation system has been developed rising and flourish as seen in Table 2. ITS has develop to many sub-areas such as Advanced Traffic Management Systems (ATMS), Advanced Traveler Information System (ATIS), Advanced Vehicle Control and Safety Systems (AVCSS), Commercial Vehicle Operations (CVO), Advanced Public Transportation System (APTS), Advanced Rural Transportation System (ARTS). In ITS America of USA, it includes seven sub-systems and among of those items the most important development is the advanced vehicle control and safety system AVCSS. In the same way Europe and Japan all make a lot of effort to study in AVCSS.

Year	2002	2003	2004	2005	2006
Number of accident	86307	120223	137221	155812	160897
Number of Deaths	2861	2718	2634	2894	2734
Number of Injuries	109612	156303	179108	203087	204187

Table 1 : Traffic accident and violation in Taiwan from 2002 to 2006 [20].

Related research organization of intelligent transportation system	
Car Manufactories	GM (Tahoe Boss), Land Rover(Land-E), Ford, Honda, Toyota, Nissan
Research Institution	US Defense Department, CSIST, ITRI, ARTC, MIRDC (Taiwan Automotive Research Consortium, TARC)
Research Laboratories	CMU (Navlab), Stanford (AI lab)

Table 2: Related research organization of intelligent transportation system

The Advanced Driver Assistance System (ADAS) has become an important component in AVCSS research area of Intelligent Transportation System (ITS). From the beginning, the researchers just hope to design a robust system which can help driver to avoid some dangerous occurrence by giving an alarm in time. Along with the progress of technology, the functions of driver assistance system are more powerful and more reliable. The Active Driving Safety System or Autonomous Guided Vehicles (AGV) has become a popular research field in the area of intelligent Transportation System.

Many advanced driver assistance products have published and promoted. The advance driver assistances include Blind Spot Information System (BLIS), Adaptive Cruise Control (ACC), Obstacle Detection, City Safety System, etc shown in Fig. 1-1, Fig. 1-2, and Fig. 1-3 [21]. Among these assistant systems, the fundamental technique is road following system, and main driving boundary detection is especially important

to realize the driving condition around the vehicle.



Fig. 1-1: The Blind Spot Information System (BLIS) [Volvo]

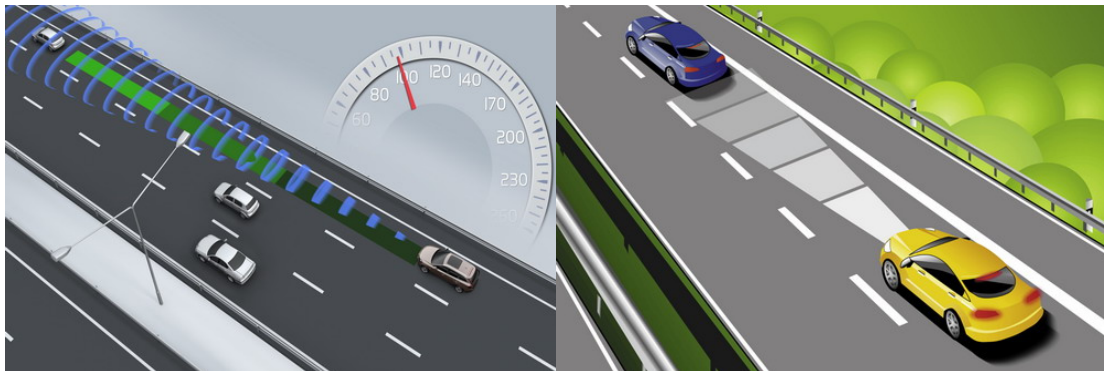


Fig. 1-2: Adaptive Cruise Control System (ACC)



Fig. 1-3: Volvo's City Safety System [Volvo]

1.2 Motivation

There are various sensors which can be exploited to capture useful features to develop different technologies in ITS research area. The tools such like radar, laser scanner, ultrasonic, CCD camera etc have been used on autonomous vehicles to detect the passable road and sense the surrounding environment. These sensors have some characteristic which appropriate to process specific function. The radar and laser scanner are usually used to detect obstacle on the road. Radar is more robust than laser in the rain or in the mist; however the cost of radar is more expensive. The ultrasonic can not work as long as laser scanner but it still helps to detect near obstacle on the lateral of vehicle. The radar, laser and ultrasonic are classified to the active sensor because they deliver the signal by themselves and estimation the distance with the obstacle using the reflective signal. The main disadvantage of active sensor is the limit of scan speed and without accurate distance estimated with obstacle. The CCD camera is another type of sensor which is called passive sensor. Due to the progressive performance of computer system, the camera becomes a common sensor in driver-assistance system.

In the AGV system, the road detection and road following is the principal element because it verifies vehicle position on the roadway and it is to support automatic navigation or curvature estimation. As a result, road following and road detection techniques have been extensively developed due to their widely applications in the Intelligent Transport System (ITS) and other similar researching fields in the past decades. The CCD camera has some prominent advantages such as the capability of acquiring abundant road information in a non-invasive way and with cheaper way to realize the road detection and road following. The most important reason is that the

active sensor can not handle the road detection and road following requirement. As a result, the vision-based road following system is a significant research area and critical component of AGV system.



Fig. 1-4: The most popular sensors are used in driver assistant systems

1.3 Objective

In the thesis, it is proposed a vision-based efficient road following algorithm. Since 1980s, road following become a widely studied field in computer vision. Although many methods have existed to handle the road detection or road following problem, these method are demonstrably limited by various assumption of the ideal condition in order to simplify the task of road following. Most of researchers focus their works on structured road, on which clear marks can be recognized or without sharply curve. Therefore the approach proposed attempts to deal most condition of road. It is presented a novel approach based on the general feature of the boundaries regardless of encountering unstructured roads, unclear road edges, non-homogeneous road appearance, arbitrary road shape and be adaptive automatically to resist interferences such as paint covered on the road or variations of lightness especially from shadows.

1.4 Organization

This thesis is organized as follows. In the next chapter, we briefly review the related topic papers then describe methods and discuss the most features which they adopted. In chapter 3, the proposed core techniques in this system are described in detail. They modify the inadequacy and disadvantage of the features. In chapter4, the main algorithm is presented. In chapter 5, we will show the experimental results and have a discussion on efficiency and reliability of proposed algorithm. Finally, the conclusions of our algorithm and future work will be presented in chapter 6.

Chapter 2

Relate Works

2.1 Overview

In recent years, many researchers have developed many techniques for road and driving situation detection in the frontal view of vehicle. The related topics to our research include “road detection”, “road recognition”, “road following”, and “lane detection” etc. In this chapter, these related works are classified by the reason that they used different approaches or different features. Furthermore, we will describe the problems which might be happened in the real condition. In section 2.2, it divides the existed methods into several bases by the features which are adopted to extract the positions of boundaries or roads and then we briefly introduce these methods.

2.2 Approaches Introduction

To distinguish the roads and boundaries for driving situation detection in the frontal view of vehicle, many researchers have proposed diverse approaches which depend on different bases and using different features. We describe the most important bases which include “region-based”, “boundary-based”, and “model-based” and introduce briefly the most popular features.

The most common features can be divided into four typical types as seen in Fig. 2-1. The methods usually utilize one or integrate different features to realize the frontal situation of vehicle during driving.

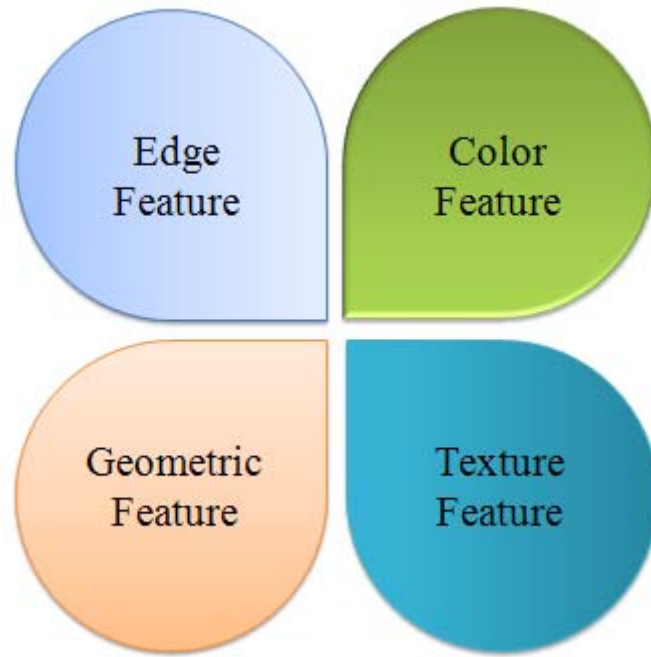


Fig. 2-1: The most popular features.

The first feature selection to detect road area is applying color information. There is some unstableness of color feature. The common causes of these unstableness including non-homogeneous surface or light variation. It is important that the method using color feature can adapt the color setting simultaneously. However, the color feature method cannot capture the precise road area and compute current road information adaptively. Furthermore, the speed of processing is time-consuming because color images require three times data quantity than the grey-scale images.

The second feature selection is to locate road boundary using edge characteristics of the boundary. If there is no distinct boundary features, or non-boundary objects may produce stronger edge intensity, all of these reasons may mistakenly lead to false detection. The third feature selection is to utilize the geometry of the road. Geometric feature largely depends on vanish line information, but camera shaking encountered during driving will destabilize the position of the vanish line in the image. On the other hand, many approaches adopt inverse perspective mapping (IPM) to transform the original image to bird-viewed image. Bird-viewed images are capable to

capture the parallel characteristic of the road boundary; but many adjustments of parameters such as the placement of the camera and internal camera information are required. Finally, the last feature, texture selection, encountered the same drawback as the color feature. Besides, since most road surface is smooth, it is hard to discriminate the drive way from the whole surrounding. We have summarized the overall disadvantages of common features in the Fig. 2-2.

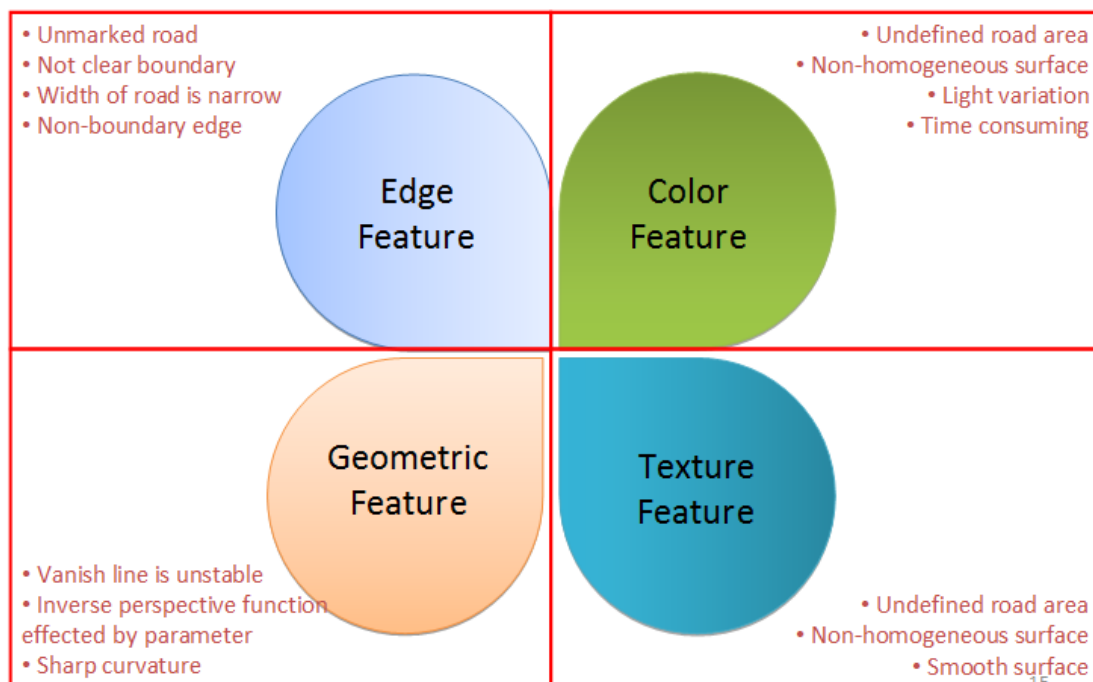


Fig. 2-2: The disadvantages of common features.

The universal approaches which can be divided to three kinds of bases are “region-based”, “boundary-based”, and “model-based”. We introduce the main idea of these methods as follows:

1. *Region-Based* :

The methods with topics about road detection or road recognition usually belong to region-based. The idea of region-based is like the solution of segmentation problems. Region-based methods adopt road features to extract the road region. The features include the color features and texture features and many methods use these

features to identify if the pixel or the region is the road or non-road. However, the road detection methods usually identify the approximate road region as seen in Fig. 2-3. Actually, the ultimate goal is to detect the boundary of the main drive path. Some methods combine the edge and geometric features about the boundary and shape of the road to estimate the candidate road regions firstly and narrow the search area from the whole image to the candidate regions in order to save the detection time.

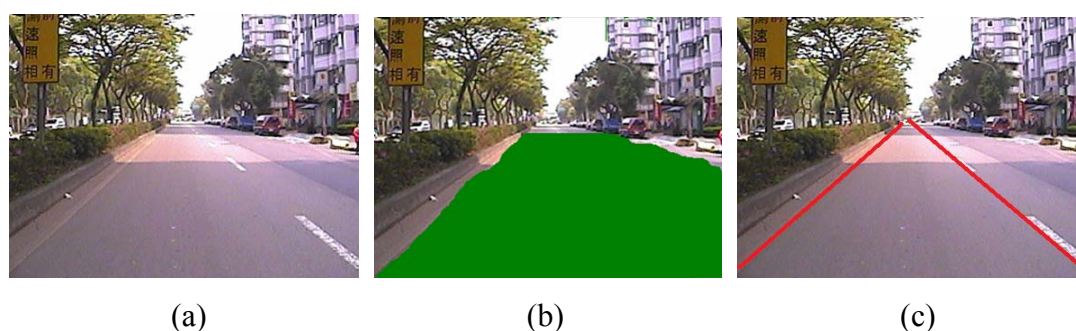


Fig. 2-3: (a) captured image, (b) road detection result, (c) the boundary of the main drive path.

The region based methods usually catch the key point on color or texture information between road region and non-road region, and apply the clustering algorithm by these features. They design a threshold or hyper-plane such as decision boundary to identify which category region or a single pixel is. Such region based methods are usually insensitive to road shapes, but sensitive to illumination variation and inconsistent surface such as shadows and non-homogeneous appearance as seen in Fig. 2-4.



Fig. 2-4: The non-homogeneous appearance of surface causes instability of the region-based without adaptive module.

The supervised classification applied to road following (SCARF) [1] and unsupervised clustering applied to road following system (UNSCARF) [12] are integrated with navigation system at NAVLAB of Carnegie Mellon University (CMU). The system can deal with unstructured in urban without obvious lane marking but with considerable difference in color between road area and surroundings. They are considered as a method of color segmentation based on clustering. The SCARF uses the RGB as color features to build multiple Gaussian color models in road/non road area and using standard nearest mean clustering (K-means) to cluster the pixels into four classes for each road and non-road with similar colors. The UNSCARF adopt iterative self-organizing data-analysis technique algorithm (ISODATA) to cluster a pixel x with five-dimensional as feature extracted by RGB color space and the coordinate of the pixel x in the image such as $x = [R, G, B, X, Y]$. These methods using the clustering technique have to process iteratively to fit the best clustering result but the steps are too time-consuming to meet the real-time requirement. In addition, they are difficult to distinguish the road from its surroundings only by color for the similar color or road area and the area that is beyond road shoulder. Furthermore, they assumed the width of road is known and the road is linear so the road recognition result is less accurate when the curvature of road is sharp and the width of road is non-uniform. As a result, they display its inability to detect arbitrary road shapes explicitly.

J.Huang 2007 [3] uses HSV color space to detect the unstructured road without priori knowledge. It just assumes a small area in front of the vehicle as a sample of road to compute means of Hue, color purity defined as Saturation multiplied by Value of HSV color space and judge a whether a pixel belongs to the road or not by the simple Manhattan distance measure with the means of color purity. The color of road

is grayish, and RGB is similar value. Therefore, the Hue component is always sensitive and unstable, but color purity composed of Saturation and Value components is not enough discriminative from road and non-road. In addition, the method belongs to the pixels based so the processing time is slow. Because they do not use any edge information of boundary, so the effectiveness of the approach is not stable. Mobileye [4] designed an autonomous driving vehicle for Darpa Grad Challenge Race in the Mojave desert. They adopted Adaboost learning principle based on the texture information of the road which is extracted by oriented filter, Walsh-Hadamard kernels and Moment to detect the off-road path. The approach is only appropriate to the surface with uniform texture on the off-road path. However, this method often fails when current road is dissimilar to that of the training set. Rasmussen [5] utilized the texture cue to navigate the vehicle on the unstructured road which is without any lane markings and a homogeneous surface. Previous approaches use color feature to build a Gaussian mixture models [6][7] but the color on the road is actual not homogenous. Road colors are different produced by the reasons including: different light, different material of surface etc, so the color model is not adaptive enough to perform precisely. Some methods proposed the learning model [8], but they don't have methods to define where the road regions are. Therefore, the samples to build the model are not appropriate and the reason why the effective of learning become even worst.

We have summarized the overall features selected and common disadvantages of the region-based approaches in the Fig. 2-5.

<i>Feature</i>	<i>Disadvantage</i>
1. Color Feature 2. Texture Feature	1. Non-homogeneous 2. Light variation 3. Can not find the driving path 4. Feature is not stable 5. Not precise 6. Consuming Time

Fig. 2-5: The feature and common disadvantages of the region based methods.

2. Boundary Based :

Some methods use the characteristic includes edge and shape of the roadside to extract the boundary position. Bertozzi et al. [10] proposed the GOLD (Generic Obstacle and Lane Detection) system by using a stereo vision-based hardware and software architecture developed to increment road safety of moving vehicles. The GOLD system addresses both lane detection and obstacle detection at the same time. Lane detection is a boundary based skill which relies on the presence of road marking, while the localization of obstacles in front of the vehicle is performed by the processing of pairs of stereo images. The IPM (Inverse Perspective Mapping) [11] is the most important component. The IPM algorithm can be used to project image plane to ground plane which is assumed to be $z=0$ in the real world as shown a bird-viewed scene shown in Fig. 2-6 (a) to (c). Meanwhile, road boundaries become parallel two lines in the bird-viewed image and extract the boundaries by the hint. However, the IPM algorithm needs appropriate parameters of the camera setting information so it is not compatible enough to use immediately after setting on the vehicle. Besides, some objects may produce edge as interference as seen in Fig. 2-6 (c), it cause difficulties of the detection.

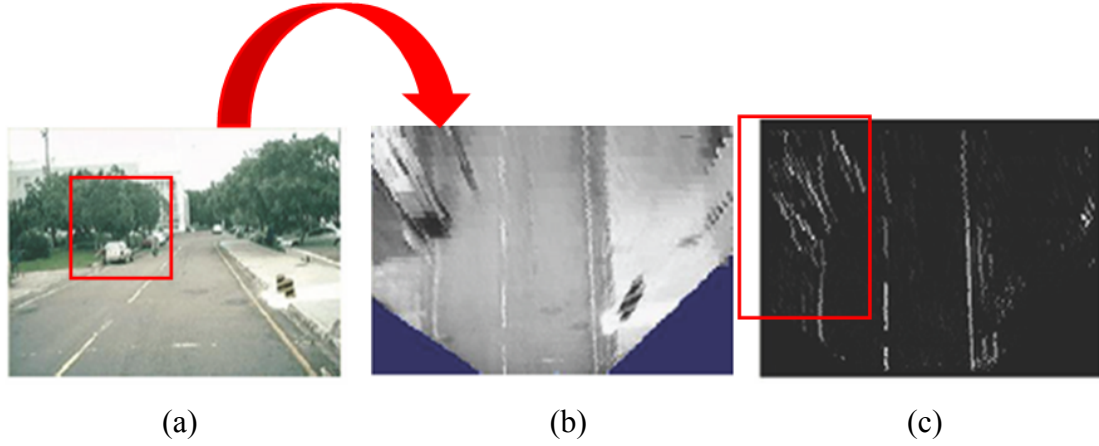


Fig. 2-6: The method transform to bird-viewed image using inverse perspective mapping (a) to (b) and extract the boundaries from bird-viewed image (b) to (c)

Most of the boundary based approaches [10][12][13][14] rely on the distinctness of the boundary, such methods assume a clear and solid lane mark on the surface of the road. The ambiguous country roadside or the broken lane mark in the middle of the road could lead to the failure in following the detection of main drive path shown in Fig. 2-7 (a) and (b). The shadow or objects at the side of the road could sometimes render the strong edge information as seen Fig. 2-7 (b) and (c); such interferences need to be avoided in order to precisely detect the road boundary.

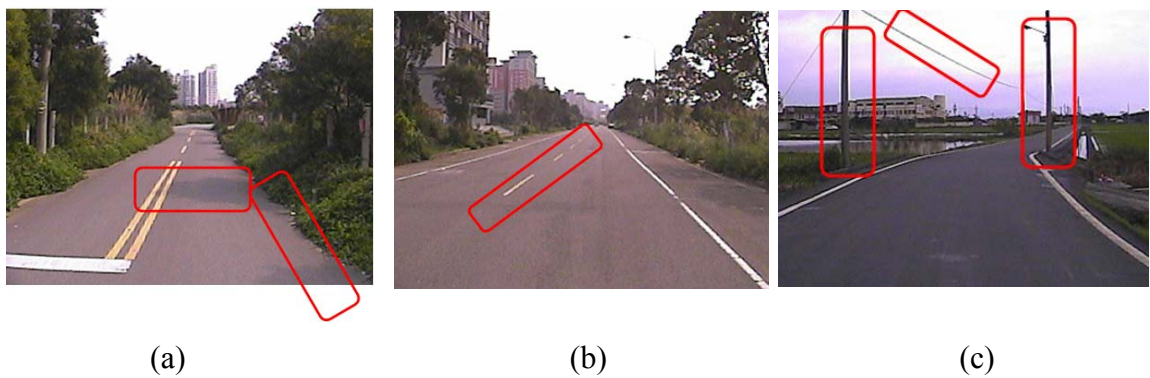


Fig. 2-7: The difficult reasons of boundary based approach.

In order to increase the accuracy, some methods usually combine the edge feature with geometric properties of the road [10][12][13]. Due to the perspective

effect of camera, each pair of parallel lines in the real world would connect to the vanish point on the horizon in the captured image. Generally, the key of the approach is to use voting procedure such as Hough transform for the edge points of the image to the (θ, γ) plane by equation (2-1) and form a sinusoidal curve in (θ, γ) plane as seen Fig. 2-8 (a) and (b) .

$$x \sin \theta + y \cos \theta = \gamma \quad (2-1)$$

They divide the image to many sections and compute the accumulator values after all edge points have been transformed to the (θ, γ) plane. It detect the pair of boundaries by extracting the pair of line with maximal accumulator and intersecting on the vanish line as seen in Fig. 2-9. These algorithms are fast and appropriate for the task of highway driving; however, it make assumptions about the structure of the road, and do not work on the unmarked roads without clear and distinct lanes on the boundaries. They have great problem by the effect on the edges of the surrounding objects, so it always is not acceptable when the road is too narrow to extract enough edge of boundaries. In addition, the camera which is set on the vehicle is shaken due to flatness of road when the vehicle is moving. As a result , the vanish line position is not stable which cause the algorithm to extract the incorrect boundaries. In addition, these approaches could not detect precisely when the road curvature is very sharp, because the boundary in the far field sections is unclear and the strong edges produced by the objects around surroundings make the curve boundary detection failed as shown Fig. 2-10. Therefore, the height of far field section must be decreased to handle the sharp curve boundary.

We have summarized the overall features selected and common disadvantages of the boundary-based approaches in the Fig. 2-10.

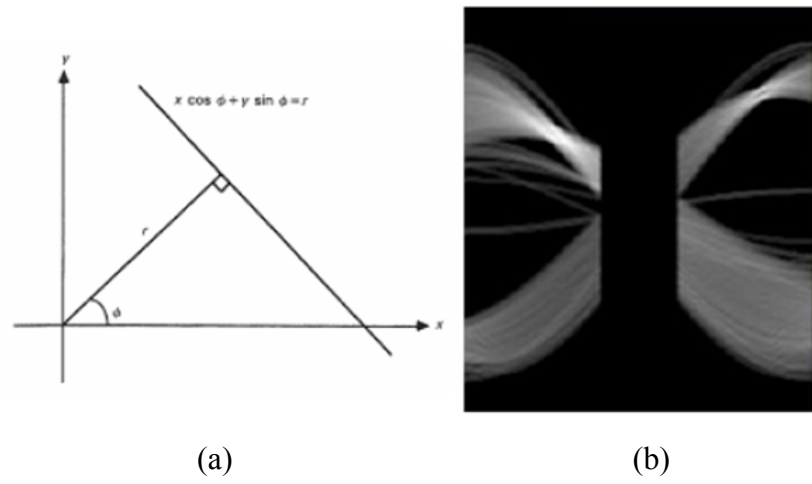


Fig. 2-8: Points transfer to (θ, γ) plane.

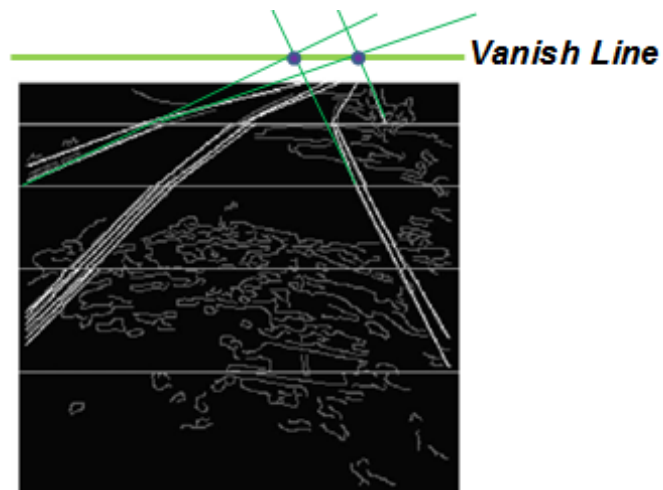


Fig. 2-9: Extract the boundaries in each section by selecting the lines with maximal accumulator values [12].

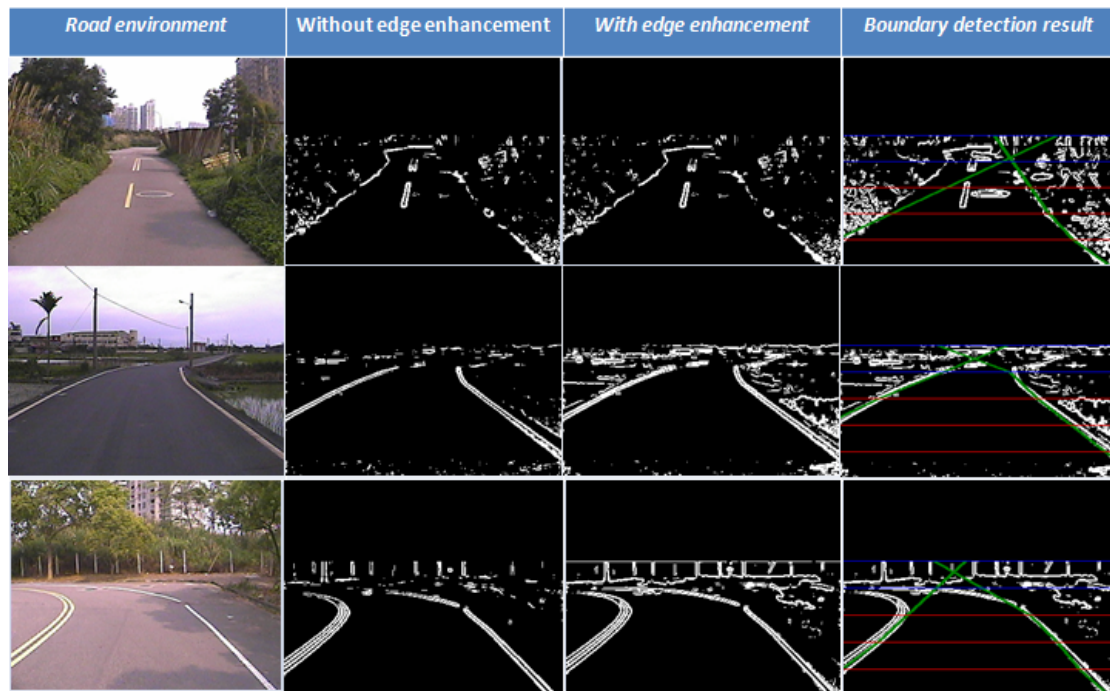


Fig. 2-10: It is suffered from edges of non road boundary and difficult to detect the curve boundary precisely

Feature	Disadvantage
1. Edge Feature	1. Unstructured road such as unmarked paved road
2. Geometric Feature	2. Non-continuous markings
	3. Horizon is not stable when encounter camera shaking
	4. Edge produced by non-boundary
	5. Sharp curvature make incorrect
	6. IPM needs different parameters

Fig. 2-11: The feature and common disadvantages of the boundary based methods

3. Model Based

The last method is the model based. They apply the mathematic function to model the road boundary shape. They use the some feature such as edge and geometric feature to converge to model to the boundary position and minimize its divergence errors. [12][13] Wang use a local interpolating spline such like

Catmull-Rom Spline and B-Spline to describe the boundary shape by setting the control points on the boundary shown in Fig. 2-12. Jung 2005 [14] utilized the edge feature and then combined edge distribution function and modified Hough transform to locate the boundary first and then used a linear-parabolic model to perform lane following and lane departure detection. The linear-parabolic model use linear equation to track the near field boundary, and then use quadratic equation to track the far field. The near field and far field is shown in Fig. 2-13 and the linear-parabolic mode is as equation (2-2). The same idea is adopted in [15] and they use vanish point and the control points on the boundary to model their lane-curve function (LCF) shown in Fig. 2-14.

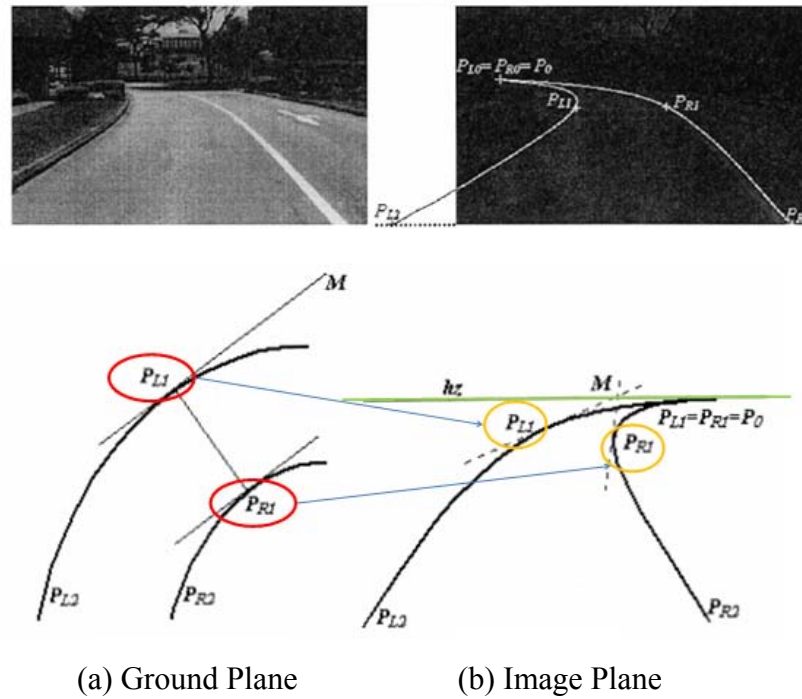


Fig. 2-12: Spline model to describe the boundary by the control points P_1 P_2 P_3

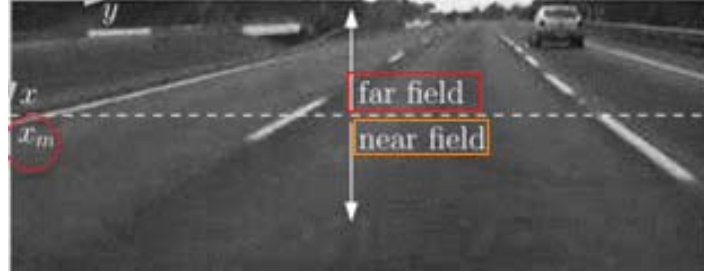


Fig. 2-13: The position X_m divides the scene into far field and near field.

$$f(x) = \begin{cases} a + bx, & \text{if } x > x_m \\ \frac{2a + x_m(b-d)}{2} + dx + \frac{(b-d)}{2x_m}x^2, & \text{if } x < x_m \end{cases} \quad (2-2)$$

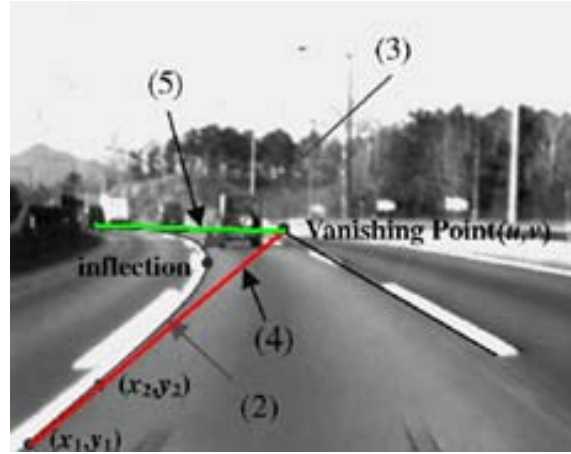


Fig. 2-14: The lane-curve function (LCF) and its asymptotes. (2) is the near field LCF, (3) is the far field LCF, (4) is the near field asymptote, and (5) is the far field asymptote.

Actually, the model based approach relies on the accuracy of the feature. If the feature is unstable, the model cannot converge to the precise position of the boundary. Moreover, if the shape of the boundary is too twisted, the first and second degree equation cannot match the curve. Multiple curves of the road could also complicate the parameters of the formula. We summarize all the common features of model based method and its disadvantage in the below Fig. 2-15.

<i>Feature</i>	Disadvantage
1. Edge Feature 2. Geometric Feature	1. Horizon is not stable when encounter shake 2. Suffer form strong edge of non-boundary object 3. Cannot work on non-continuous lane 4. Sharp curvature make incorrect 5. Solve many parameters

Fig. 2-15: The feature and common disadvantages of the model based methods

Chapter 3

Proposed Techniques

3.1 Objective

The last chapter two has introduced current approaches using different bases including region based, boundary based, and model based. Besides we have realized frequent inadequacy of the features they adopt. Therefore, we proposed the two core techniques in this chapter in order to promote the precision and plasticity of the road following system. The two main core techniques are: 1. On-lined L^*a^*b color model and 2. LMR boundary window. The first technique attempts to train the adaptive color model and modify the non-homogeneous drawback of the road color. The second technique uses a new tracking tool called LMR boundary window to extract boundary position by edge intensity difference. We introduce the details of the core techniques and explain how they can solve the inadequacy of the features.

3.2 On-lined L^*a^*b color model

3.2.1 Objective

Most region based approaches frequently use the color feature to detect the road. Color information extends into three times dimensions of original grey scale image so it is with better discrimination. Therefore, there will be results in a more completed and precise detection. Most current papers propose off line method to determine the ranges of each color vector. This method only enables the detection of road area

within the color range. However, off-line method is very ineffective because of the difficulty to cope with the heterogeneity of surface material and the variation of illumination. Therefore, we purpose an on-line learning model that allows continuously update during driving. Through the training method, we can enhance plasticity of the system.

Besides, due to the uneven distribution of the road color, we select the samples which most closely resemble the detection zone to build the exclusive color model. In addition, we don't detect the whole image but set the each LMR boundary windows which prepare to track the boundary position as the region of interesting (ROI). Therefore, when we track the boundary we just detect each LMR boundary window (ROI) in order to increase the detection speed and the performance of updating models. Besides, each LMR boundary window has exclusive color model whose on-lined sampling area is very close to its corresponding LMR boundary window. The exclusive model and its adequate sampling area can promote detection accuracy in each LMR boundary window when the road surface with uniform color appearance.

3.2.2 Color space selection

In this section, we illustrate the advantaged properties of L^*a^*b color space by comparing the other color space and explain the reason we adopt the L^*a^*b color feature for modeling.

We attempt to describe the color appearance in the driving environment by selecting the color features and using these color features to build a color model of the road, therefore we have to choose a color space which has uniform, little correlation, concentrated properties in order to increase the accuracy of the model. In computer color vision, all visible colors are represented by vectors in a three-dimensional color

space. There are many common color spaces that have been used to facilitate the analysis of color image.

Among all the different color spaces, RGB color space is the most common color feature selected because it is the initial format of the captured image without any distortion. Using the RGB color space to build model had been seen in many approaches [16]. However, the RGB color feature is high correlative, and the similar colors spread extensively in the color space. As a result, it is difficult to evaluate the similarity of two color from their 1-norm or Euclidean distance in the color space.

It takes some experimental results to explain the L*a*b color space is a better choice than others. RGB color space is susceptible to illumination variation on the road such as shadow. In Fig. 3-1, we track the values of (Imax-Imin) of each component on a selected road area over time which distribute at the horizontal axis using both RGB and L*a*b. We can see the RGB is greatly interfered by the shadow which leads to the values (Imax-Imin) of each component in the road area fluctuate drastically. On the contrary, the (Imax-Imin) of L*a*b is quite stable around the value of 0 to 30 intensity at the vertical axis except during the time interval from 80th to 100th frames. Fig. 3-1, the peak during the time 80th to 100th frames is resulted from a plait on the road.

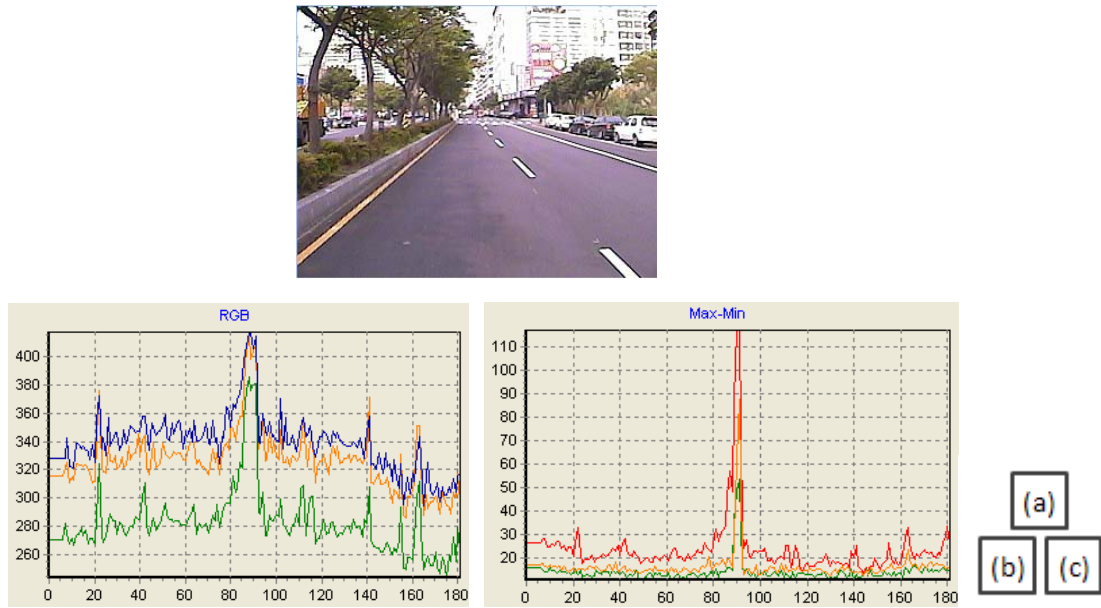


Fig. 3-1: The captured video (a). The max-min value of each three components of RGB at the vertical axis distribute around the frame sequence at the horizontal axis (b) and same as L*a*b (c). It is obvious that the L*a*b is more stable.

The other standard color space HSV is supposed to be closer to the way of human color perception. Both HSV and L*a*b resist to the interference of illumination variation such as the shadow when modeling the road area. However, the performance of HSV model is not as good as L*a*b model because the road color cause the HSV model not uniform lead to the HSV color model not as uniform as the L*a*b color model. There are many reasons attribute this result. Firstly, HSV is very sensitive and unstable when lightness is low. Furthermore, the Hue is computed by dividing $(I_{\max} - I_{\min})$ in which $I_{\max} = \max(R, G, B)$, $I_{\min} = \min(R, G, B)$, therefore when a pixel has a similar value of Red, Green and Blue components, the Hue of the pixel may be undetermined. Unfortunately, most of the road surface is in similar gray colors with very close R, G, and B values. If using HSV color space to build road color model, the sensitive variation and fluctuation of Hue will generate inconsistent road colors and decrease the accuracy and effectiveness.

L*a*b* color space is based on data-driven human perception research that assumes the human visual system owing to its uniform, little correlation, concentrate characteristics are ideally developed for processing natural scenes and is popular for color-processed rendering [17]. L*a*b* color space also possesses these characteristics to satisfy our requirement. It maps similar colors to the reference color with about the same differences by Euclidean distances measure and demonstrates more concentrated color distribution than others. As a result, we will compare with BGR and HSV through experimental result.

We compare the detection results between L*a*b* and HSV color space in the Fig. 3-2. First of all, L*a*b* is more sensitive toward the grayish color than the HSV color space, so it can yield a better result in road detection. Secondly, due to the concentration property of L*a*b* color space, it can use fewer and less spread Gaussian models to express overall road scenery. As a result, the method using L*a*b* color space takes less time for road detection.

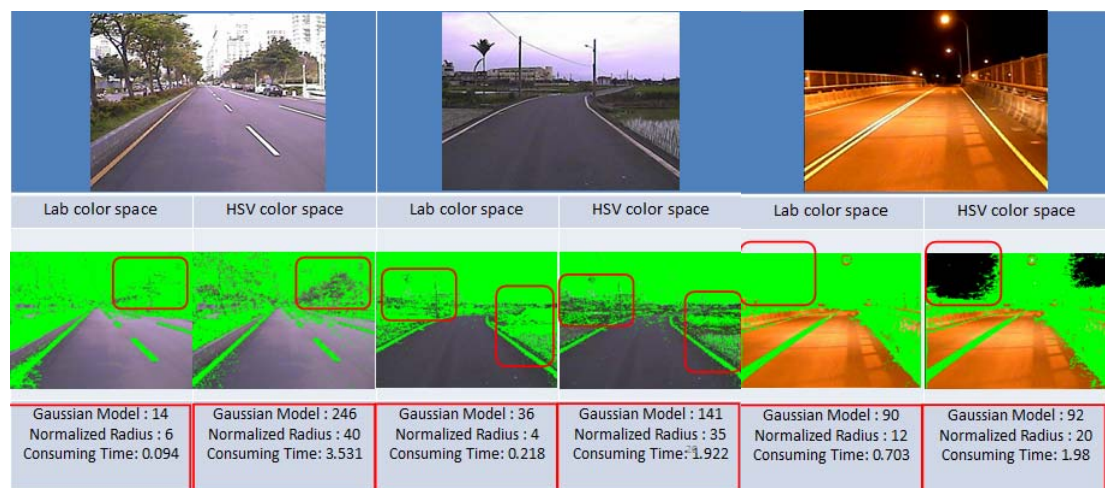


Fig. 3-2: The comparison results using L*a*b* color space and HSV color space.

The RGB-L*a*b conversion is described as follow equations:

1. RGB-XYZ conversion:

$$X = 0.431 \cdot R + 0.342 \cdot G + 0.178 \cdot B \quad (1)$$

$$Y = 0.222 \cdot R + 0.707 \cdot G + 0.071 \cdot B \quad (2)$$

$$Z = 0.020 \cdot R + 0.130 \cdot G + 0.939 \cdot B \quad (3)$$

2. Cube-root transformation:

$$L^* = 166 \cdot [Y / Y_n]^{1/3} - 16 \quad \text{if } Y / Y_n > 0.008856$$

$$L^* = 903.3 \cdot [Y / Y_n]^{1/3} \quad \text{if } Y / Y_n \leq 0.008856 \quad (4)$$

$$a^* = 500 \cdot [f(X / X_n) - f(Y / Y_n)] \quad (5)$$

$$b^* = 200 \cdot [f(Y / Y_n) - f(Z / Z_n)] \quad (6)$$

Where X_n , Y_n , Z_n , are XYZ tristimulus values of reference white point, and $X_n=95.05$, $Y_n=100$, $Z_n=108.88$, and

$$f(x) = \begin{cases} t^{1/3} & Y / Y_n > 0.008856 \\ 7.787t + 16 / 116 & Y / Y_n \leq 0.008856 \end{cases} \quad (7)$$

3.2.3 On-lined Model

This section illustrates the each step of the modeling procedure in detail. The rest of this section is organized according to the order of modeling step: I. guide snake sampling, II. blob noise checking, and III. modeling and updating of the L*a*b color model, as seen Fig. 3-3. These three steps describe building and updating the model and they perform every n input frames to increase processing speed but still maintain high accurate performance. The IV part illustrates using on-lined learning model to achieve road detection.

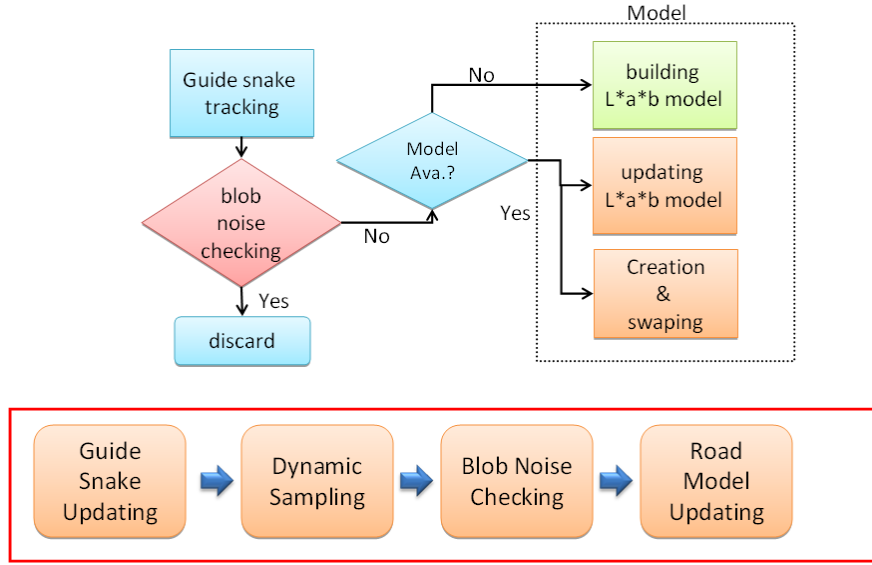


Fig. 3-3: The upper part shows the flow chart of on-lined modeling. On-lined learning procedure is summarized in the red block.

I. Guide Snake Tracking

The proposed method is based on on-line training to promote the adaptability of the color model. Therefore, properly label the road sample for learning is crucial. There are some papers [8][18] assume the frontal fixed area pixels are the road samples. There are two disadvantages we like to discuss. Firstly, using this assumption to define road is not quite reliable. If the road curvature is large, this static way of labeling might encounter errors in defining road area. Secondly, due to variability of road color, fixed samples cannot closely resemble the detection area. As a result, we proposed a more reliable sampling method called “guide snake tracking”. The head of guide snake will track the end of the road scene by pattern matching [8], optical flow, and edge concentration property as seen in Fig. 3-4, and connecting from the tail of the snake (origin of the arrow) to the snake’s head (head of arrow) forms a body of guide snake. The guide snake follows along the road curvature by wiggling its body with its head tracking the end of road. Therefore, it can sample the accurate road area for learning when the road is sharp curve. Besides, when we track the

boundary in Section 4.3.2, we set the LMR boundary window as a detection area but don't detect the whole image shown in Fig. 3-5 (a). Therefore the guide snake set the exclusive sampling area shown in Fig. 3-5 (b) and lay the position of the exclusive sampling area is as closer to the corresponding LMR boundary window as possible. Therefore, it makes the on-lined learning model more adaptive to the non-uniform color of the road surface.



Fig. 3-4: Guide snake keep tracking the curvature oft the road

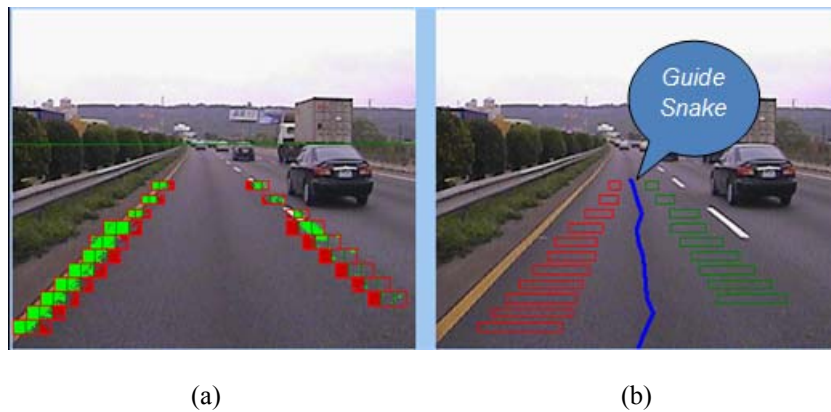


Fig. 3-5: (a). LMR boundary window is set as ROI when performing tracking step in section 4. (b). Guide Snake make LMR boundary window have its exclusive sampling area for on lined model learning.

II. Blob Noises Checking

There are many non-road objects which exist on the surface of roads including paints, lane markings, and other signs. These non-road objects existed on road area are defined as “blob noises”. Whenever sampling area of each LMR tracking window pass over these blobs noises on the road, blobs noises will generate large numbers of non-road pixels which mistakenly assimilate into sample. If the samples are used to update the model, it could detriment the fidelity of model. Many methods with updating module resist these blob noises by slowing down the speed of updating to increase the noise toleration of the model. Nevertheless, it would be resulted in a big error when driving onto different kinds of surface because the newly arrived road surface can not update the model rapidly and delay the learning progress. Only rely on adjusting the updating speed to solve the blob noises will decreases the model’s adaptive ability, so it is a contradiction problem. Therefore, we propose the “blob noise checking” algorithm to detect blob noises appeared on the road area. If the blob noise checking detects the blob noises in the on-lined sampling area, it will discard these non-road pixels to prevent from sampling these pixels to update the model and therefore enhance the robustness of the model.

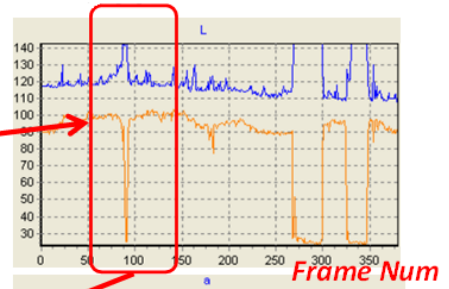
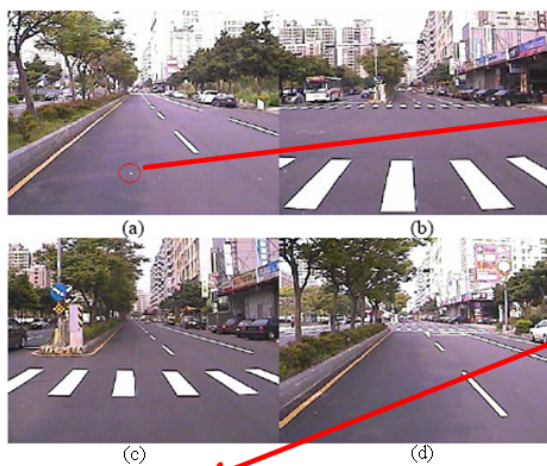
The algorithm to automatically detect the blob noises in the samples area relies on special property of L^*a^*b color feature. From the experimental results, we find L^*a^*b color feature is sensitive to the apparent non-road objects. It provides the significant cue to identify which the pixels belonged to blob noises.

When the sampling area covers the road area with blob noises, the maximum and minimum each component of L^*a^*b will rise and drop distinctly. It is because of the color of blob noises is quite different from the color of road areas and the edge of blob boundary is sharp. The maximum value was rose for the first reason and the minimum

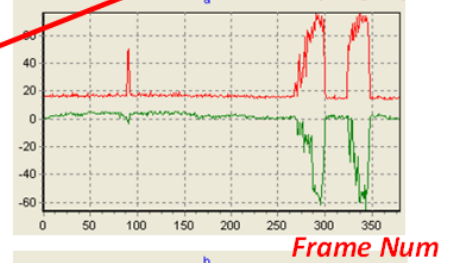
value was fallen for the second reason. As shown in Fig. 3-6 (a) to (c) the blob noises cause a distinct pattern of peaks and troughs of maximum and minimum value at the times around 80th, 250th, and 325th frames shown as a red arrow from image to L intensity chart and with similar pattern happened in a and b intensity chart to corresponding frame number, in which the vertical axis represents intensity value and the horizontal axis represents frame number. Nevertheless, since $L*a*b$ is robust against the lightness variation, there is no such patterns when the sampling area encounters shadows on the road shown in Fig. 3-6 (d) corresponding to the stable $L*a*b$ max and min values.

As a result, we could characterize the maximum L, a, b value in the sampling area subtracting from minimum one and filter out non-road samples. As seen the max-min (L, a, b) chart in Fig. 3-6, if the peak exceeds $TH_{blob}=100$, the blob noise checking activates to prevent these samples with blob noises at the moment from updating the model. Therefore, we can filter the samples with blob noises by threshold process show as the red blocks in Fig. 3-7.

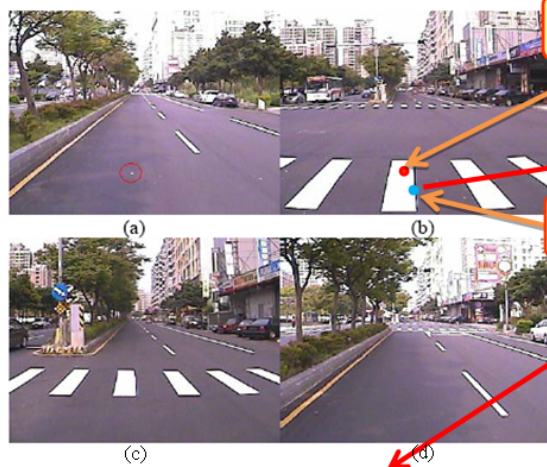
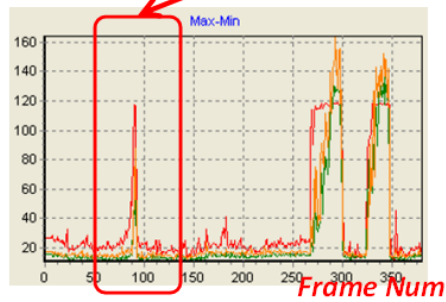
If the vehicle is driven onto a different kind of road surface such as from asphalt surface to cement surface, the blob noise checking may impede updating temporarily, but it will restart to update the model in a moment owing to the fact of the smooth surface homogeneity existed on every different types of roads.



a



b



Max

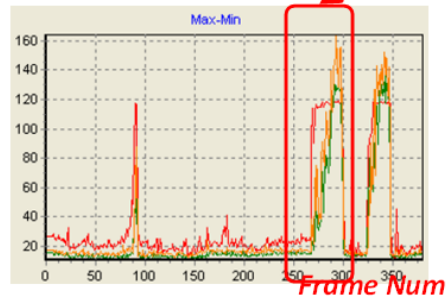
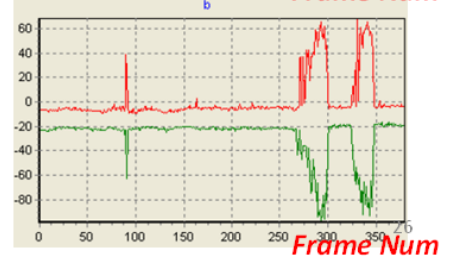
L



a



b



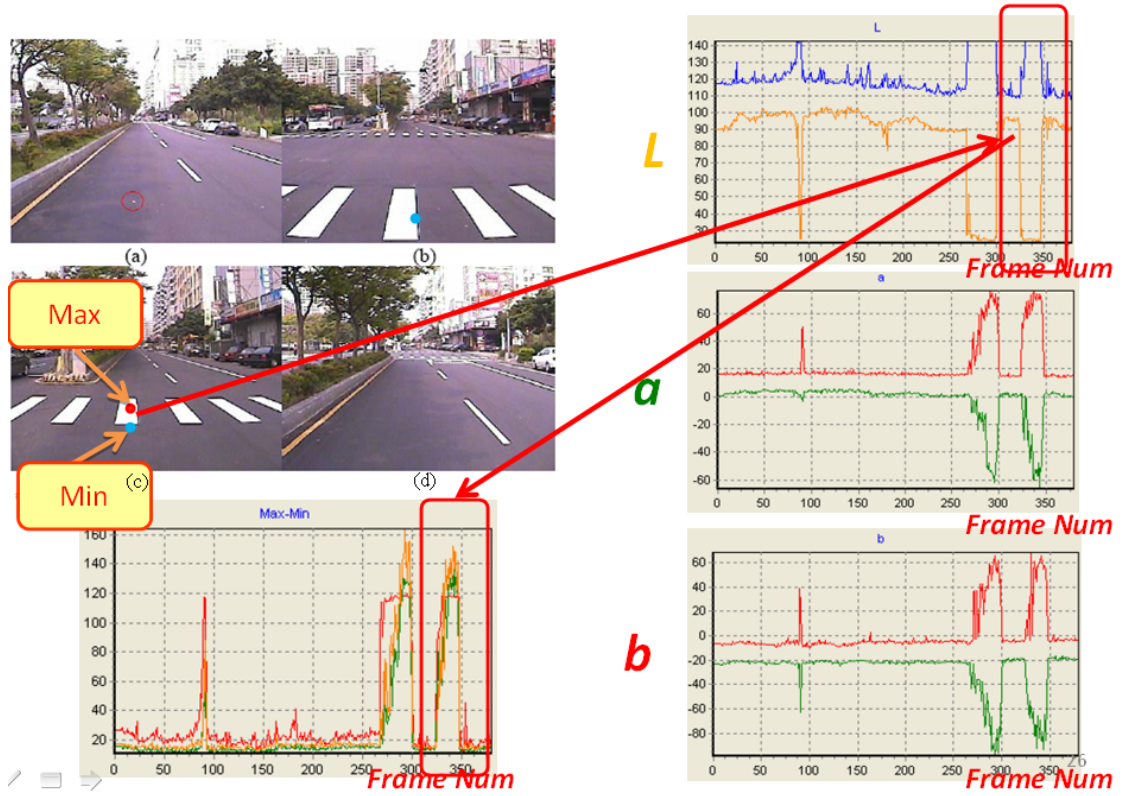


Fig. 3-6: Experimental images. (a)~(c) are blob noises, and (d) has no blob noise but with shadow. Using the pattern of the peak value of max-min in each L^*a^*b component to detect the blob noises. The peak around 90th frames corresponds to the blob in (a), and the other two peaks around 280th frames and 340th frames correspond to the blobs (b) and (c). The other stable max-min values correspond to the scene with some shadows as (d).

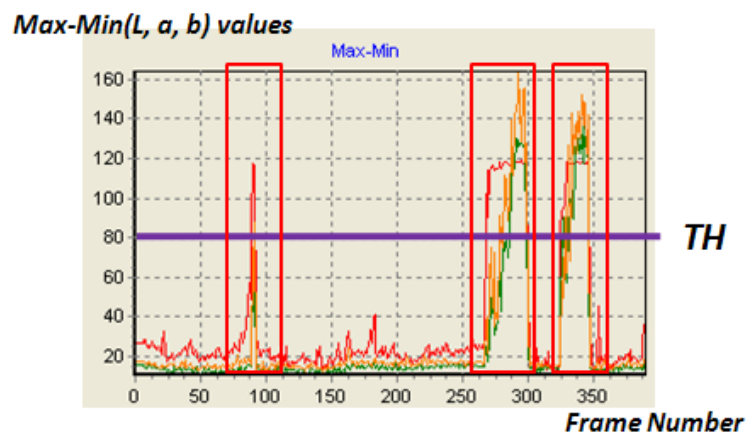


Fig. 3-7: Filtering the blob noises.

III. Building and updating the color model

The objective of the proposed method is to train the model. This on-lined model can learn the road color immediately. The L^*a^*b model is constituted of K color balls, and each color ball m_i is formed by a center on $(L_{m_i}, *a_{m_i}, *b_{m_i})$ and a fixed radius $\lambda_{\max} = 5$ as seen in Fig. 3-8.

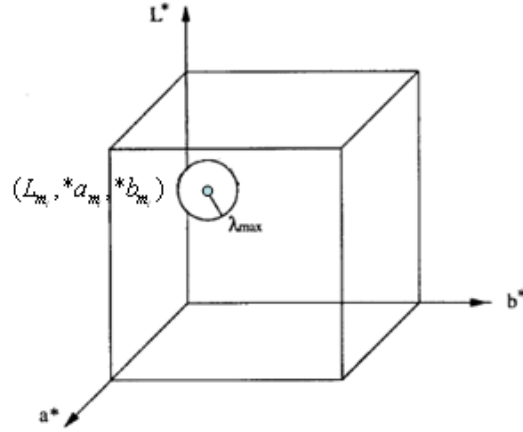


Fig. 3-8: A color ball i in the L^*a^*b color model whose center is at $(L_{m_i}, *a_{m_i}, *b_{m_i})$ and with radius λ_{\max} .

The previous record of on-lined sample of each LMR tracking window tracked by guide snake is modeled by a group of K weighted color balls. We denote the weight and the counter of the m_i *th* color ball at a time instant t by $W_{m_i,t}$ and $Counter_{m_i,t}$. We applied some ideas proposed by Stauffer and Grimson in [19], the weight of each color ball represents the stability of the color. The color ball which more on-line samples belonged to over time accumulates a bigger weight value shown in Fig. 3-9. Adopting the weight module increases robustness of the model. The counter of each color ball records the number of pixels added from the on-line samples at each sampling iteration, therefore the counter must be set to zero for each new incoming samples.

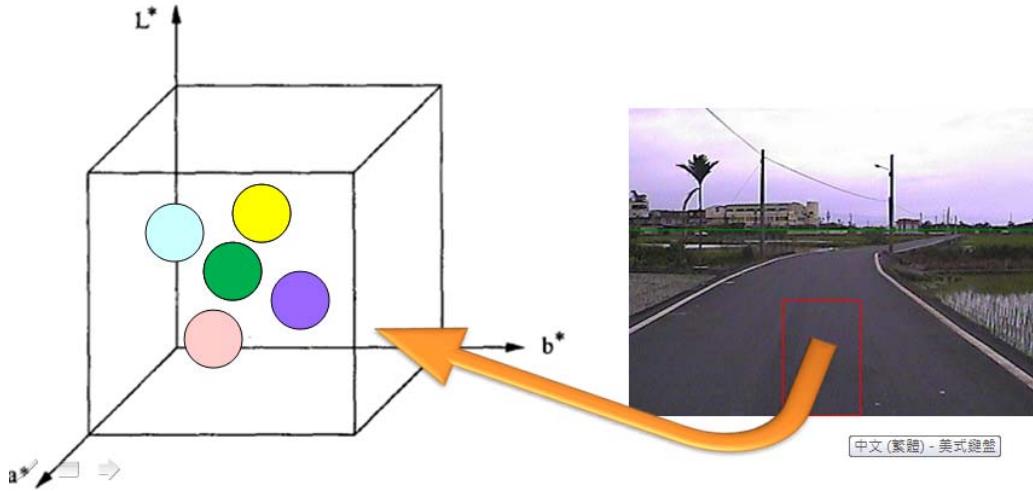


Fig. 3-9: Every color ball with a weight which represents the similarity to current road color. We use the saturation of the color to indicate the intensity of the weight.

The weight of each color ball is updated by its counter when the new sample is coming which is called one iteration. Therefore the counter would be initialized to zero at the beginning of iteration. The counter of each color ball records the number of pixels added from the on-line samples in the iteration. The first thing to do is that which color ball is chosen to be added. We measure the similarity between new pixel x_t and the existing K color balls using a Euclidean distance measure (3-1). The maximum value of K is 50 which represents each on-lined model contains 50 color balls at most.

$$Similarity(x, m_i) = \sqrt{(L_{m_i} - L_x)^2 + (a_{m_i} - a_x)^2 + (b_{m_i} - b_x)^2} \leq \lambda_{\max} \quad (3-1)$$

If none of the color ball covers the new pixels x_t from incoming samples, there are two procedures to handle this situation. First, if K is lower than 50, a new color ball will be created whose center tagged as $(L, *a, *b)$ of x_t and assigned 1 as its counter, 0 to its weight. On the other hand, if K is equal to 50, a new color ball whose center tagged as x_t and assigned 1 as its counter will be substituted for the lowest weight color ball.

If a new pixel x_t was covered by any of the color ball in the model, one will be added to the counter of best matching color at this iteration as the equation (3-2). After entire new sample pixels at this iteration undertake the matching procedures mentioned above, the weights of every color ball are updated according to their current counter and their weight at last iteration. The updating method is follows (3-3) :

$$find\ m(x_i) = \arg_{m_i} \min(Similarity(x_i, m_i) \leq \lambda_{\max}) \quad (3-2)$$

$$W_{m_i, t+1} = \alpha_w \cdot W_t + (1 - \alpha_w) \cdot counter_{m_i} / |N_{snake}|$$

$$\alpha_w \in [0, 1], \quad |N_{snake}| \neq 0 \quad (3-3)$$

, where α_w is the user-defined learning rate.

If a color space accumulate incoming pixels constantly, its weight will increase to indicate that the color is similar to the road color prototype. Therefore, the weight of the color ball is directly proportional to the stability and resemblance to the current road. On the contrary, if there is no sample pixel added to the color ball over several iterations, the color ball becomes less important in the detection due to its decreasing weight value. Eventually, it is replaced by the new color ball.

Next, we need to decide which color ball of the model most adapt and resemble current road. We use the persistence of the color ball as an evidence for this, because the persistence of the m_i th color ball is directly related to its weight $W_{m_i, t}$. The color balls are sorted in a decreasing order according to their weights. As a result, the most probable road color features are at the top of the list. As a last phase of the updating procedure, the first B color balls (3-4) are selected to be enabled as standard color for road detection in the next IV detection part. For this reason, the color ball with a higher weight has more importance in detection step.

$$W_{m_1,t} + W_{m_2,t} + W_{m_3,t} + \dots + W_{m_B,t} \geq TH_w$$

where $TH_w \in [0,1]$

(3-4)

IV. Road detection using on-lined color model

Road detection is achieved via comparison of the new pixel x_t with the existing B standard color balls selected at the previous instant of time shown in Fig. 3-10. If no match is found, the pixel x_t is considered as non-road. On the contrary, the pixel x_t is detected as road.

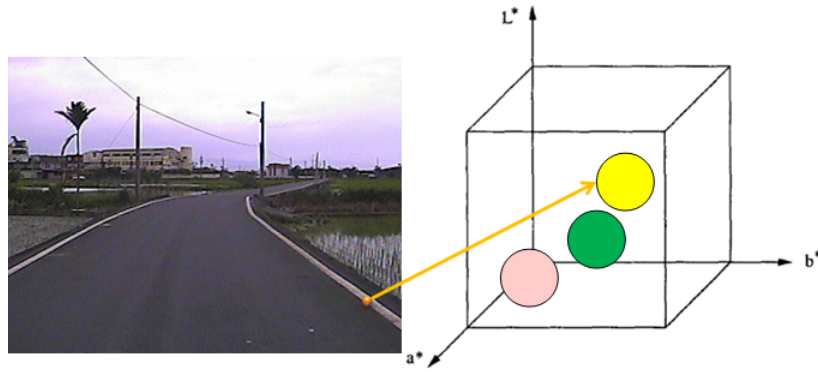


Fig. 3-10: The pixel matched with first B weight color balls which are the most represent standard color.

3.3 LMR boundary window

3.3.1 Overview

After describing the method of training on-lined color model which applies the learning method to automatically extract the color information of road; we will introduce another core technique of the system: LMR boundary window. In the Chapter 4, we utilize LMR boundary window with simple computation cost to locate the most suitable boundary position and continuously track along the boundary

position. Therefore, we focus on introducing the structure of LMR boundary window and its power (attraction force) is attributed from the two factors, Region Ratio and Boundary Energy, while the window tracking the boundary. We apply related concepts from the Chapter 2 which adopt similar idea of region based, boundary based and model based approaches.

3.3.2 LMR boundary window introduction

LMR boundary window is constituted of three separate blocks of L, M, and R. The size of each block is according to the distance from the block to the vanish line. The greater the distance away from the vanish line, the wider the range of the boundary will appear on the image. Therefore, the boundary window also has to extend relatively in size to completely cover the boundary region, so it can capture sufficient boundary characteristics. These characteristics include powers (attraction force) attributed from the road area distribution and boundary intensity difference. The LMR boundary intends to change its original position because of the powers (attraction force). Then it aligns the most current position of boundary to achieve tracking effect as shown in figure 3-11. Next, we will introduce powers (attraction force) which induced by characteristics of boundary in section 3.3.3.

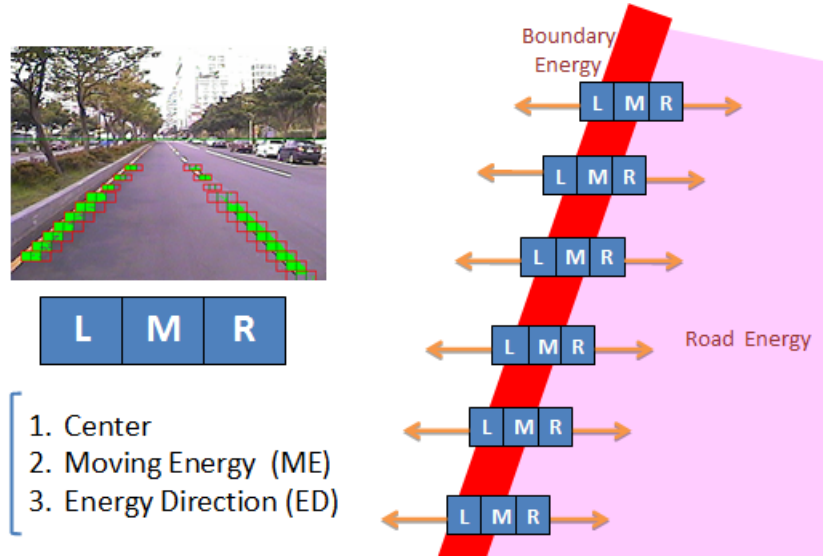


Fig. 3-11: The structure of LMR boundary window and the main moving energy concerning the boundary energy and road energy.

3.3.3 Boundary feature of LMR boundary window

We utilize the concept of region based and boundary based approaches to discover experimentally that there are two characteristics belonged to region based and boundary based at the road boundary. Firstly, the road boundary stands between the transition of road and non-road region. As shown in the red box in Fig. 3-12, the green area belongs to non-road region; other area belongs to the road. As a result, there must contain both road pixels and non-road pixel within the LMR boundary window. We design a method to extract this characteristic by analyzing the road and non-road ratio. We call this feature as **Region Ratio**; **Region Ratio Features** can be subdivided into **NonRoadPower** and **RoadPower** as following equation (3-5). When computing Region Ratio, we utilize on-lined color model learned from the exclusive sampling area of LMR boundary window to achieve most accurate performance of detection, as shown in Fig. 3-13. Secondly, based on some concepts of LMR boundary window; we aim to characterize the LMR boundary window at the most extreme edge intensity difference. We purpose second characteristic which is

Boundary Energy Feature to assist calculating the biggest boundary energy in one of the L, M, and R block in the LMR boundary window; it can therefore represent the most appropriate boundary position. The boundary energy of each of the block is expressed in the equation (3-6) as BEnergy. Within the equation, the GS(i) accumulates edge gradient intensity through M_n frames in the block “i”. δ is the average gradient intensity of every pixel on a smooth road. If maximum boundary energy is no longer resided in the middle M block of LMR boundary, it signals a translocation of the boundary, and necessitates changing the position of LMR boundary to dynamically follow along the movement of road boundary. We illustrates the translocation or deviation of LMR tracking window as the top and the last row of Fig. 3-14 (a) and (b) and the alignment of middle M block to the road boundary as the middle row of Fig. 3-14 (a) and (b). In Fig. 3-14 the solid red line represents road boundary and the pink highlighted region represents road area.



Fig. 3-12: The area around the boundary shown the red box must have specific road region ratio/non-road region ratio of the whole area. Besides, the area has the obvious difference as seen the yellow line.

If LMR boundary window is on the left boundary of driving path

$$\begin{cases} \text{NonRoadPower} = \text{NonRoad}(L+M)/\text{Total Pixels} \\ \text{RoadPower} = \text{Road}(R)/\text{Total Pixels} \end{cases}$$

else

$$\begin{cases} \text{NonRoadPower} = \text{NonRoad}(R+M)/\text{Total Pixels} \\ \text{RoadPower} = \text{Road}(L)/\text{Total Pixels} \end{cases}$$

NonRoad:the number of non-road pixels in the LMR boundary window detected
by the color model.

Road : the number of road pixels in the LMR boundary window detected
by the color model.

If LMR boundary window is on the left boundary of driving path

$$\begin{cases} \text{BEnergy}(L) = \left[\sum_{i=0}^{i=M_n} \text{GS}(L) - \sum_{i=0}^{i=M_n} \text{GS}(M) \right] / N(\text{block}) \\ \text{BEnergy}(M) = \left[\sum_{i=0}^{i=M_n} \text{GS}(M) - \sum_{i=0}^{i=M_n} \text{GS}(R) \right] / N(\text{block}) \\ \text{BEnergy}(R) = \left[\sum_{i=0}^{i=M_n} \text{GS}(R) - M_n * \delta * N(\text{block}) \right] / N(\text{block}) \end{cases}$$

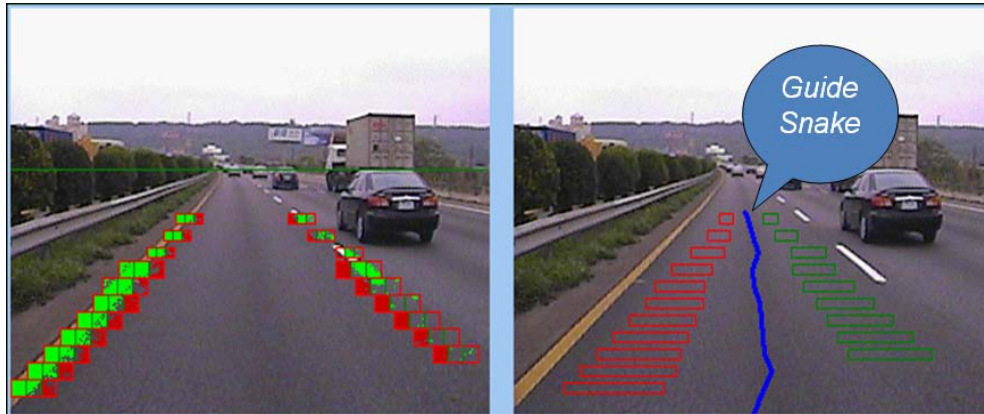
else

$$\begin{cases} \text{BEnergy}(L) = \left[\sum_{i=0}^{i=M_n} \text{GS}(L) - M_n * \delta * N(\text{block}) \right] / N(\text{block}) \\ \text{BEnergy}(M) = \left[\sum_{i=0}^{i=M_n} \text{GS}(M) - \sum_{i=0}^{i=M_n} \text{GS}(L) \right] / N(\text{block}) \\ \text{BEnergy}(R) = \left[\sum_{i=0}^{i=M_n} \text{GS}(R) - \sum_{i=0}^{i=M_n} \text{GS}(M) \right] / N(\text{block}) \end{cases}$$

GS(i):total gradient sum of block "i"

δ :mean gradient of noise per pixel

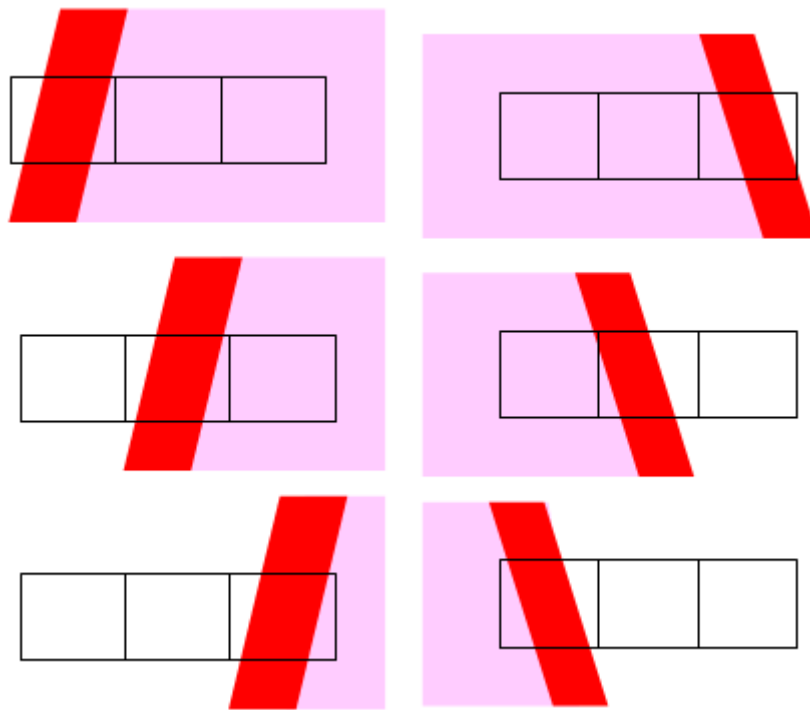
N(block):total pixels of the block



(a)

(b)

Fig. 3-13: Guide snake set the nearest sampling in (b) corresponding to each LMR boundary window in (a) and build their exclusive color model.



(a)

(b)

Fig. 3-14: The boundary resides on the one of the block in LMR boundary cause the maximum boundary energy on that block. (a) is situation that the left boundary of driving path lies different block . On the contrary, (b) is show the right LMR boundary window.

Chapter 4

System Algorithm

4.1 Overview

After completely introducing two main techniques of the system, we will dissect main algorithm of the system in this chapter. Our system can be categorized into two main parts: 1. Main Driving Path Boundary Detecting and 2. Boundary Tracking. The comprehensive algorithm flow chart is shown in the Fig. 4-1. System starts with preprocessing by maintaining the edge intensity of notable boundary and eliminating noises, then it enter the yellow box of the flow chart. The yellow block is about driving path boundary detection which is described in the Section 4.2. After completion in boundary locating, the process subsequently apply boundary tracking procedure as shown in red block of the flow chart and we discuss it in Section 4.3 .

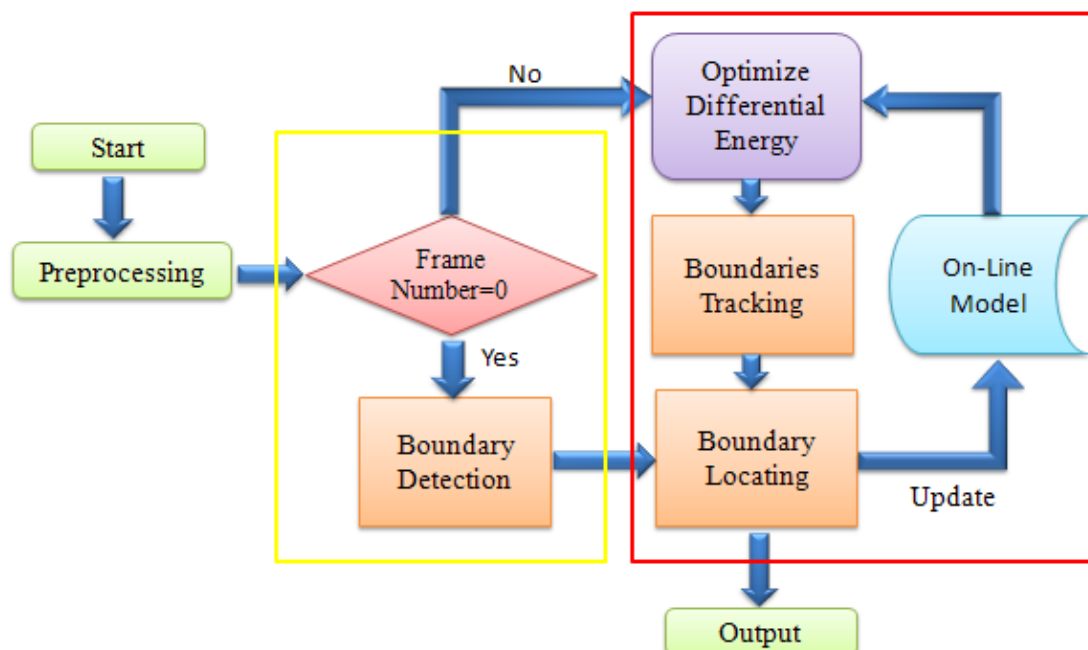


Fig. 4-1: The system architecture of the algorithm.

4.2 Driving Path Boundary Detection

4.2.1 Objective

In this chapter, we cover the initial step of the purposed algorithm. The major goal of the initial step is to detect the main driving path boundary and precisely locate the position of boundary, then to further promote the tracking ability of LMR boundary window. On a road of multiple lanes, we define the boundary of our driving lane as main driving path boundary. In order to enhance the adaptability of the system, we need to overcome the major challenges by acquiring the capability to handle various types of boundaries. Those types of boundaries which weaken the edge feature include several conditions, for example, ambiguity of lane mark, absence of lane mark, broken lane mark, and sharp curvature of the driving path. By making use of general characteristic of road boundary, the method we purposed is still able to accomplish stability in detection performance.

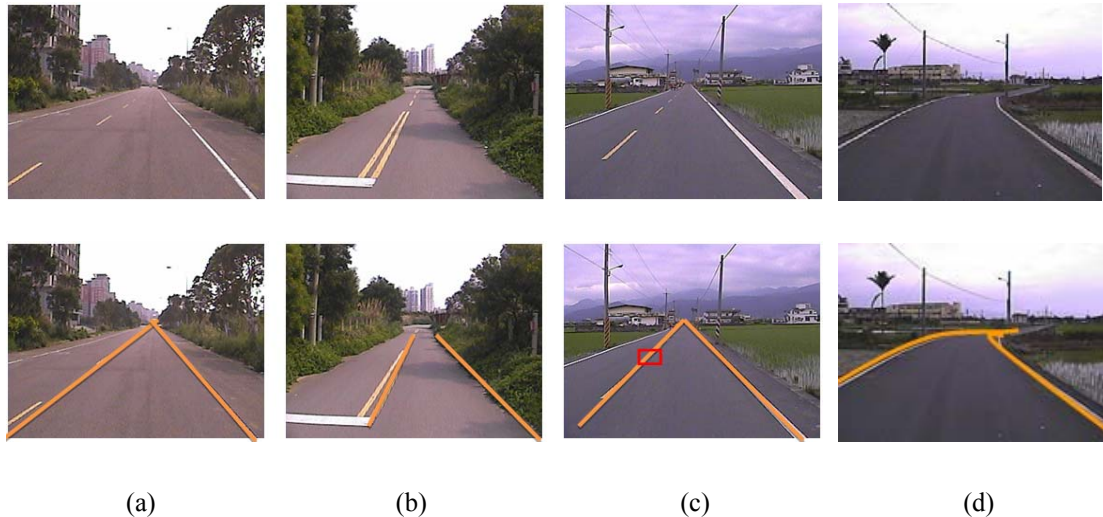


Fig. 4-2: There are several common challenges in boundary detection. (a) Ambiguity of lane mark. (b) Absence of lane mark. (c) Broken lane mark. (d) Sharp curvature of the driving path. We detect the main driving path boundary as the yellow line and locate the LMR boundary window at the boundary as the red box.

4.2.2 Methods

In this initial step, it is right at the beginning of our driving. We utilize initial 15 to 30 extracted frames to swiftly locate the accurate boundary position. The comprehensive algorithm flow chart is shown in Fig. 4-3. We mainly apply edge feature but still encounter the insatiable detection result, therefore we alter the method by incorporating consideration of color feature in the region of road detection. As a result, the method combines both edge based feature and color based feature to detect the main driving path boundary. Next, we will introduce the processing procedures of edge based method and colors based method, and discuss the integration of color-based feature to solve the inadequacy of merely using the edge feature.

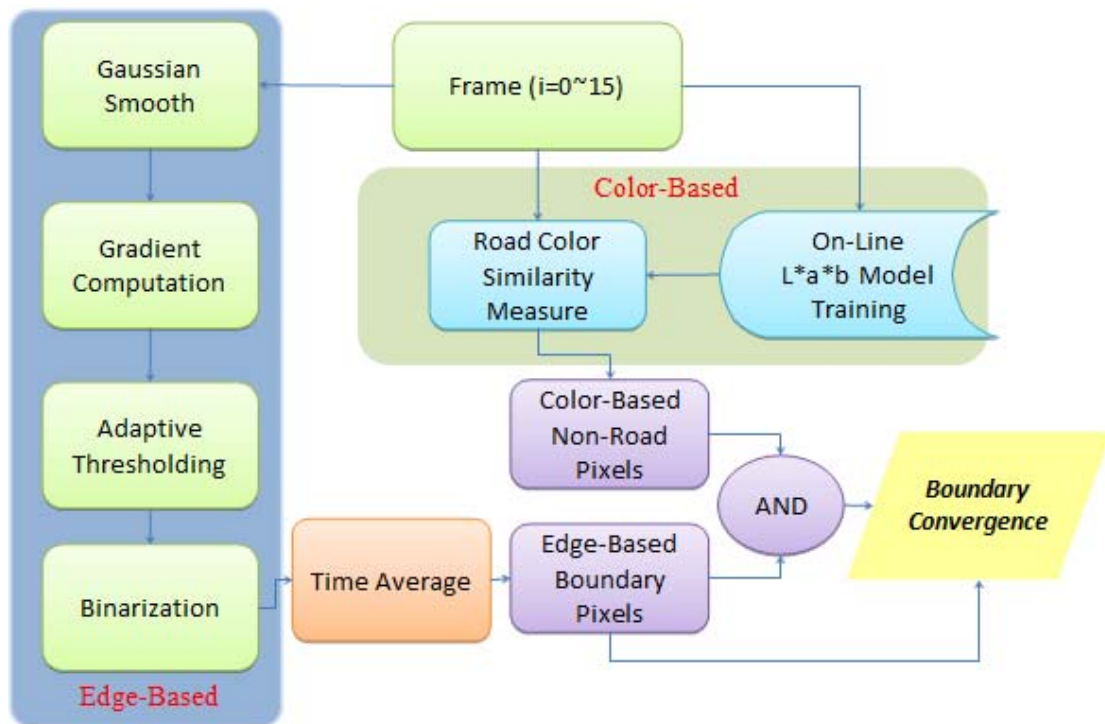


Fig. 4-3: The flow chart of Driving Path Boundary Detection

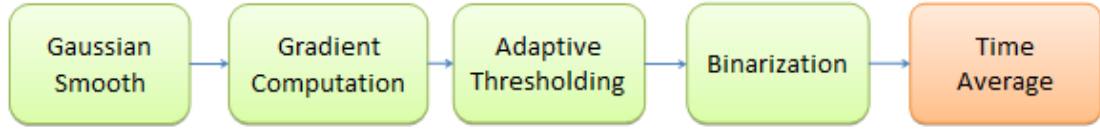


Fig. 4-4: The procedure of edge-based approach

Our method mainly adopts the edge feature to detect the boundary and we describe the whole procedure about it. A set of frames at the initial detection all undergoes primary procedure shown in Fig. 4-4; green blocks indicate multiple processing steps. Once the processing steps are completed, the temporal binary result submits to the final step of the procedure called Time Average as shown in the orange block.

The first step is Gaussian Smooth, because CCD camera might capture images contained serious noise from the environment and illumination, as shown in Fig. 4-5 (c). Since the major characteristic of boundary is edge feature, a step to eliminate the noise is necessary, or it might produce similar extreme gradient intensity which result in false edge detection and reduce the accuracy of detection. We manipulate 5 x 5 Gaussian masks, shown in Fig. 4-6, to filter out the noise. Compare with mean filter, our method not only maintain the relative edge gradient intensity of the boundary but also is competent for de-noise ability, as show in Fig. 4-5 (b).

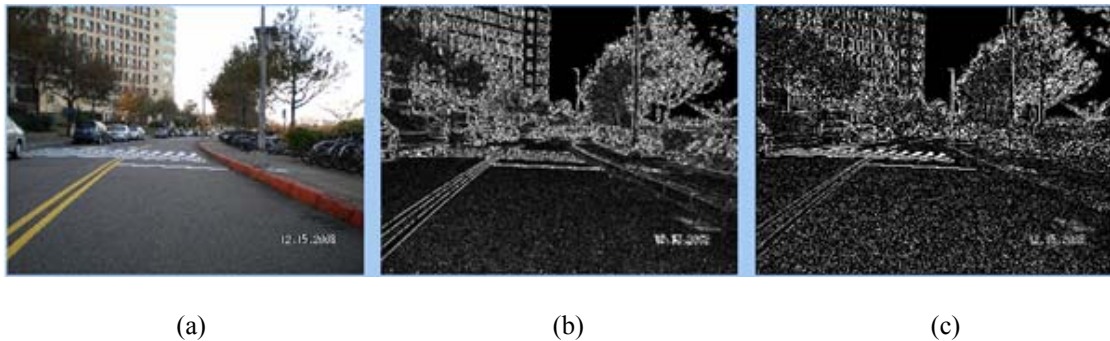
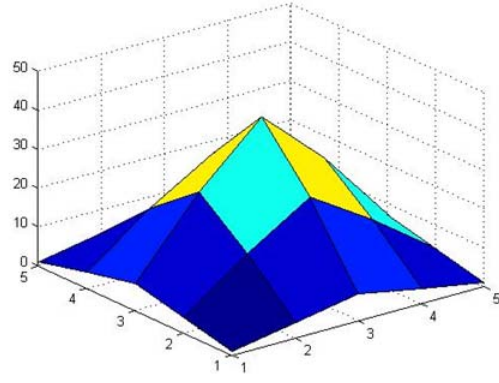


Fig. 4-5: (a): The captured image. (b): Gradient image after de-noise. (c): Gradient image without de-noise.

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

(a)



(b)

Fig. 4-6: (a): Suitable 5×5 mask of Gaussian filter with $\sigma=1$ and (b): 2-D Gaussian distribution representation

Once Gaussian Smooth is done, the subsequent step is Gradient Computation. Using x direction and y direction of sobel filter to acquire the $G_x(x,y)$ and $G_y(x,y)$ value of each pixel; $G_x(x,y)$ and $G_y(x,y)$ each represent the gradient intensity of x direction and gradient intensity of y direction. Then we apply equation (4-1) to calculate the gradient intensity $I(x,y)$ of each pixel.

$$I(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

$$\begin{cases} I(x, y): \text{Gradient intensity of the pixel (x,y)} \\ G_x(x,y): \text{The x-direction gradient intensity of the pixel (x,y)} \\ G_y(x,y): \text{The y-direction gradient intensity of the pixel (x,y)} \end{cases} \quad (4-1)$$

After processed by gradient computation, the image needs to be decided the threshold and binaries using the adaptive threshold to extract strong boundary. Due to the contrast between the boundary and the neighborhood road surface, the gradient intensity of the boundary caused by the edge operator as we mention before is usually larger than other locations. Therefore, the adaptive threshold uses the values of mean and standard deviation computed by each row in the image to be selected as the threshold for different region.

Take the normal distribution for example, the range which contained the distance

for one standard deviation from the mean will account for about 68% of the whole set. Besides, the range will account for about 95% if it contains the distance for two standard deviations from the mean. For each row in the image, the threshold value is still selected by referencing above scattered property since the gradient magnitude of boundary is relatively higher than that of the normal road surface.

However, we take into consideration that as car approaches the vanish line, boundary will become less obvious and non-road region will occupy bigger area. It is possible that some other objects might also produce strong gradient intensity. If we only consider the mean and standard deviation of each row to calculate the threshold of the entire pixels within the row, it could hinder the extraction of the terminal road boundary in the subsequent binarization. Therefore, the values of thresholds situated in different location are selected by tuning the mean value of each-row pixels and the arrangement of magnitude for them are from the bottom to the top sub-region, as described in the following (4-2). We manipulate the value of α to adjust the end threshold and enable binary extraction of furthestmost road boundary.

$$Threshold(y) = \left(\text{Mean}_{x \in (0, \text{width of Image})} [I(x, y)] - \alpha \right) + k \cdot \text{Standard deviation}_{x \in (0, \text{width of Image})} [I(x, y)]$$

where $\alpha = 0.2 \cdot \sum_{i=\text{Image Height}}^{y-1} (\text{Standard deviation}_{i \in (0, \text{width of Image})} |I(i, j)|) \cdot k / (y - 1),$ (4-2)

After applying binary image to extract the most obvious boundary shown in Fig. 4-7 (c), we rely upon the white perimeter to locate the main driving path boundary. Since one or both sides of main driving path boundary could consist of broken lines, shown in the right hand side of Fig. 4-7, we proposed a method called “Time Average” to promote accuracy of boundary detection. Through the process of binarization, we sum up the times of each pixels belonging to boundary in the several

frames of the beginning, and divide this sum by the number of passing frames to acquire the $TimeAverage(x,y)$ of each pixel. If the $TimeAverage(x,y)$ is bigger than $TH=0.5$, as shown in (4-3), then this coordinate could become the potential driving path boundary. Utilizing this method, we preserve the main driving path boundary by its consistent patterns of solid line and broken line and disregard the edges produced by other objects shown in Fig. 4-7 (d).

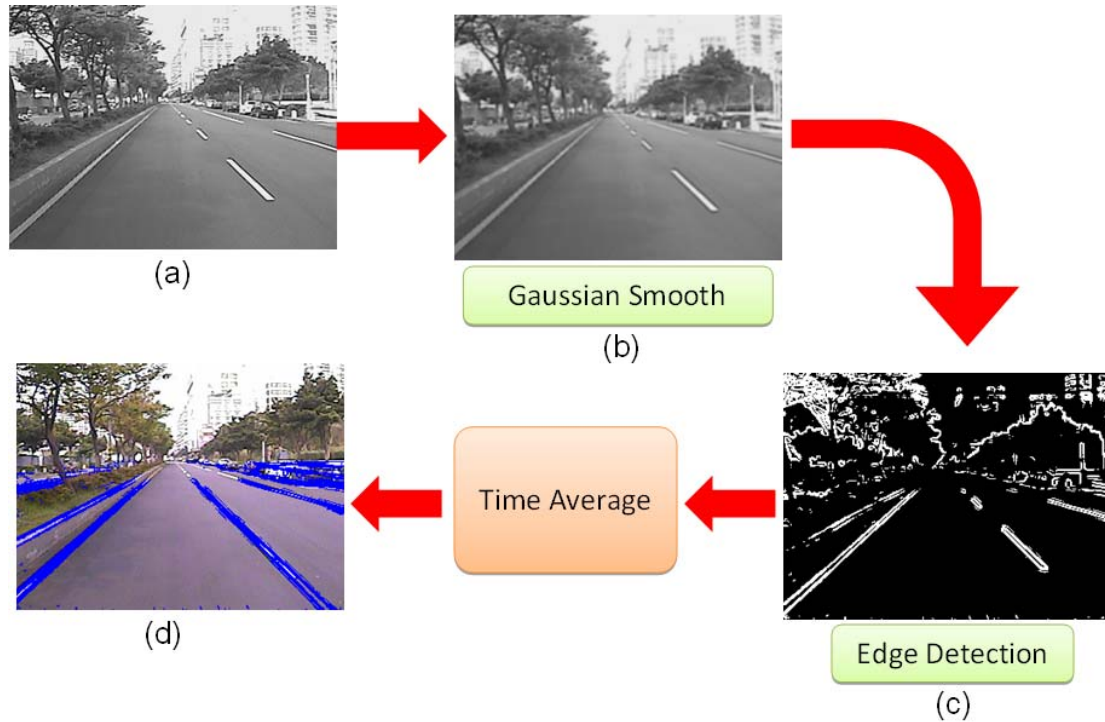


Fig. 4-7: The result of each step of the edge-based approach

$$\begin{cases} IsBoundary(x, y) = True & ,if \quad TimeAverage(x, y) > TH \\ IsBoundary(x, y) = False & ,else \end{cases} \quad (4-3)$$

If we only rely on the previous edge feature method to locate the driving path boundary, the detection result will not be precise enough when the road boundary is unclear or curl. Especially, if right hand side boundary turn rightward and left hand side boundary turn leftward, the false detection result would occur. After a curl driving path boundary is processed by the procedure of $TimeAverage$ mentioned above, a build-up boundary location at the turn is too close the road side in error as

shown in Fig. 4-8. Therefore, the modified method integrate the color feature to robust the accuracy and solve the problem which illustrate as the following.

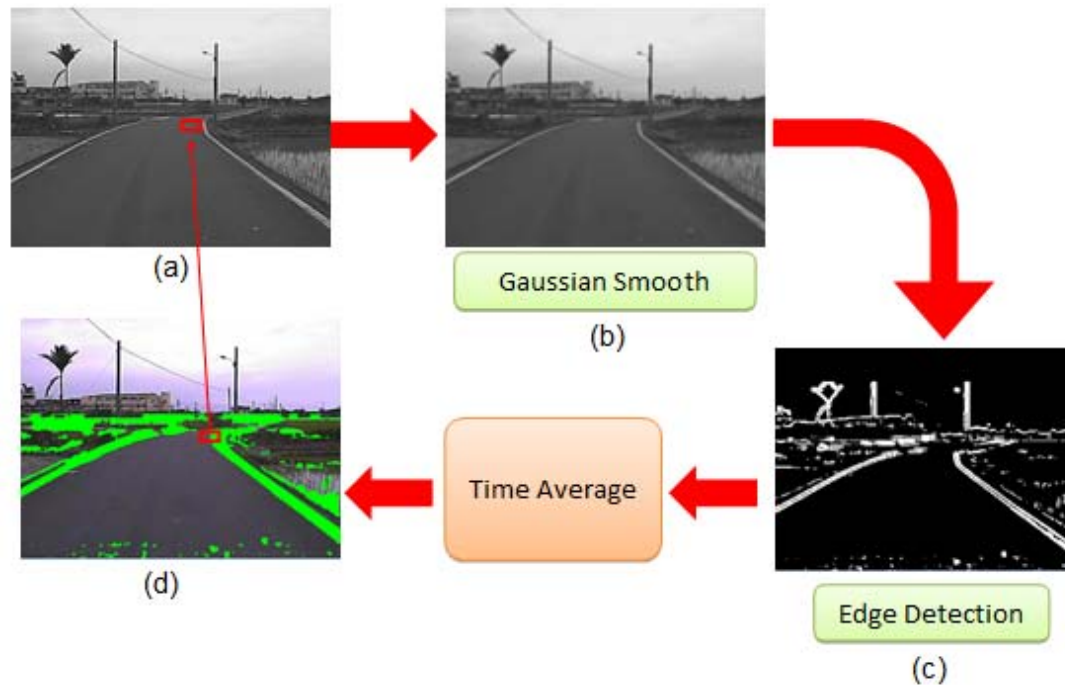


Fig. 4-8: If the method only apply the edge based approach, it detects the wrong position of boundary shown as the red box in (d) corresponding to (a) when encountering curve road.

To solve the problem, we more adopt the color feature in the initial detection boundary step. We use the on-lined color model technique mentioned in Section (3-2) to learn the color distribution on the road. In this moment, the training sample is extracted from the pre-defined region which just in front of vehicle. After training by a set of frames at the initial, the system will have enough capability to detect non-road pixels at the last frame and the result is called the color-based non-road pixels shown in Fig. 4-9. Then we process the AND operator using the pixels of color-based non-road pixels result and the edge-based boundary result which come from TimeAverage step. The AND operator can produce the optimal boundary location shown in Fig. 4-9 (a). Then, the edge-based boundary result shown in Fig. 4-9 (b) converges toward this optimal boundary location to find the accurate boundary

position. Using this modified method which integrate the edge-based feature and color-based feature to increase the plasticity of boundary detection. Therefore, it can handle various type of boundary including the particular challenge, broken line boundary and road with sharp curve.

By way of driving path boundary detection, the LMR boundary windows set their initial tracking position following the boundary location. Then we will enter the algorithm of boundary tracking in Section 4.3.

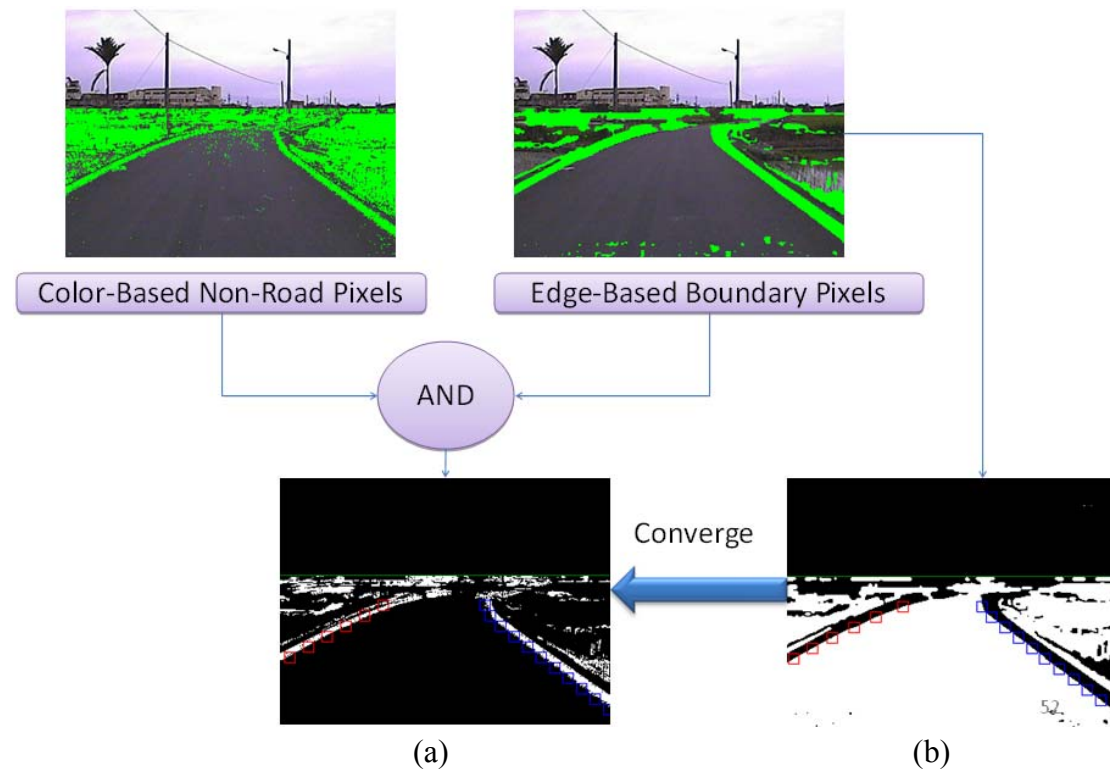


Fig. 4-9: (a): Optimal convergence boundary. (b): The boundary detected by only edge-based method. We modify the edge-based boundary detection (b) to converge to (a) which is formed using AND operator by the color-based non-road pixels position and edge-based boundary pixels.



Fig. 4-10: Using the modified boundary detection solves the problem when encountering curve road boundary. The modified result is converging from (a) to (b).

4.3 Boundary Tracking Algorithm

4.3.1 Objective

After finishing the driving path boundary detection in Section 4.2, we have set the LMR boundary window to the boundary location shown in Fig. 4-11 (a) to (c) and the guide snake dynamically arrange the specific sampling area to each exclusive $L \times a \times b$ color model of each LMR boundary window Fig. 4-11 (d). Next, we focus on the boundary tracking algorithm shown in the red block of Fig. 4-12.

We attempt to utilize both boundary features, *Region Ratio Feature* and *Boundary Energy Feature*, of LMR boundary window which is mentioned in Section 3.3.3 to adjust its position according the boundary moving variation. We use this adjusting procedure to track the boundary and this adjusting procedure is using the idea of optimization differential energy shown in Fig. 4-12. The optimization can be acquired by moving LMR boundary window to the boundary position for a most stable differential energy. When LMR boundary window moving to appropriate position, it update the boundary location. It repeats the same procedure to achieve

boundary tracking.

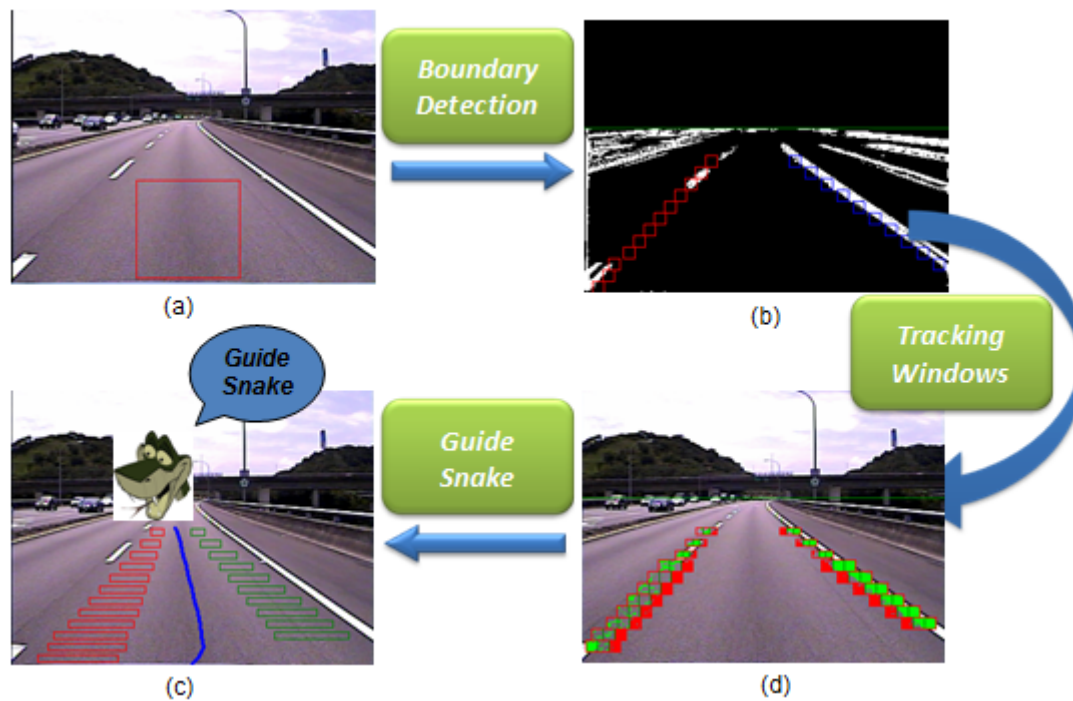


Fig. 4-11: (a) Main driving path boundary detection, the red box area is used to learn the color model at the driving path boundary detection step. (b) The result of driving path boundary detection. (c) Set the LMR boundary window at the boundary. (c) Each color model of LMR boundary window is arranged a specific sample area by the guide snake.

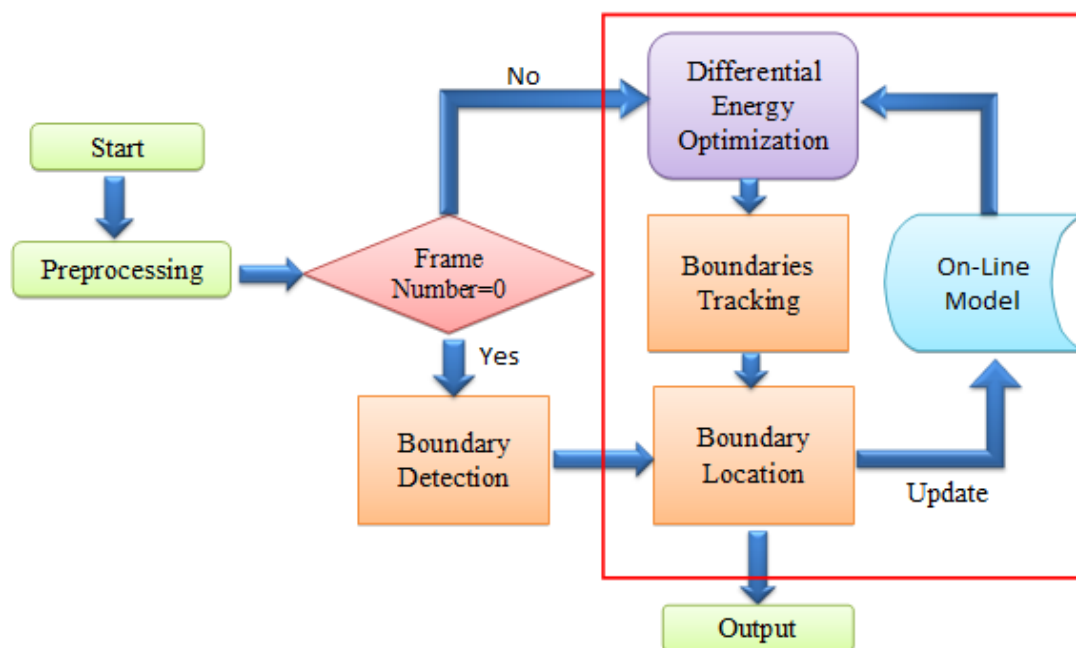


Fig. 4-12: The red box is the part of boundary tracking algorithm in the system architecture.

4.3.2 The Alignment of LMR Boundary Window

In this section, we describe how to accomplish the differential energy optimization and align the LMR boundary window to the appropriate location when it deviates from boundary. The system accumulates boundary feature in M_n frames before initiating alignment. Therefore, LMR boundary window doesn't need to align to boundary every frame in order to increase the consuming time and can accumulate more comprehensive boundary feature. When the system capture the M_n frames, it accumulates boundary energy of L, M, R blocks in the LMR boundary window and updates the color model of each LMR boundary window shown in the Fig. 4-13.

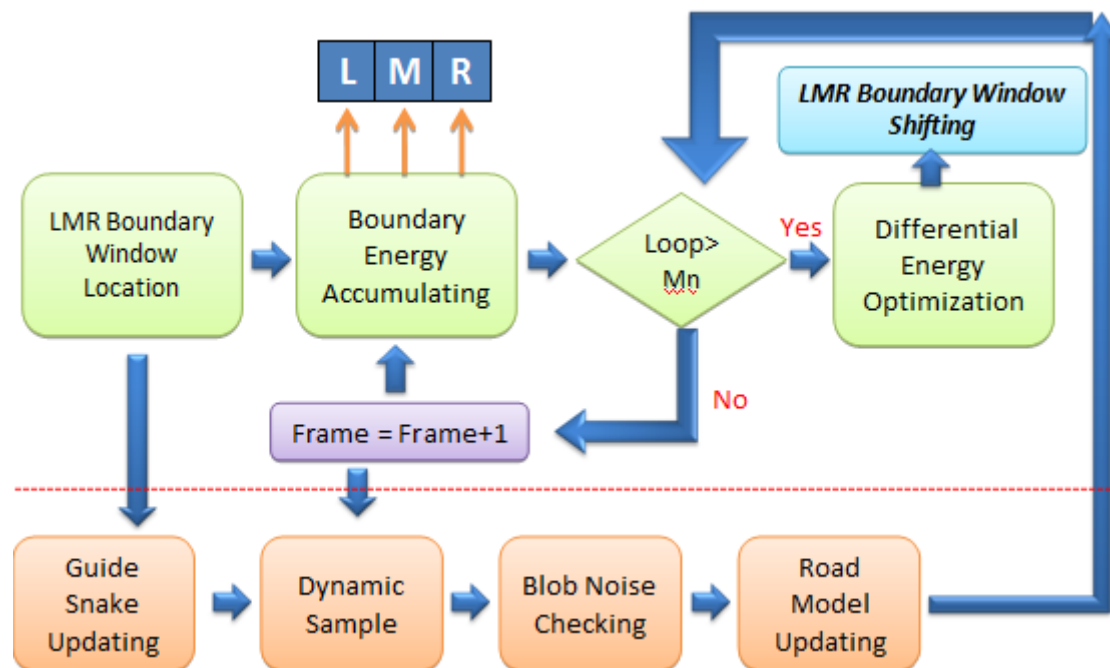


Fig. 4-13: The flow chart of alignment of LMR boundary window.

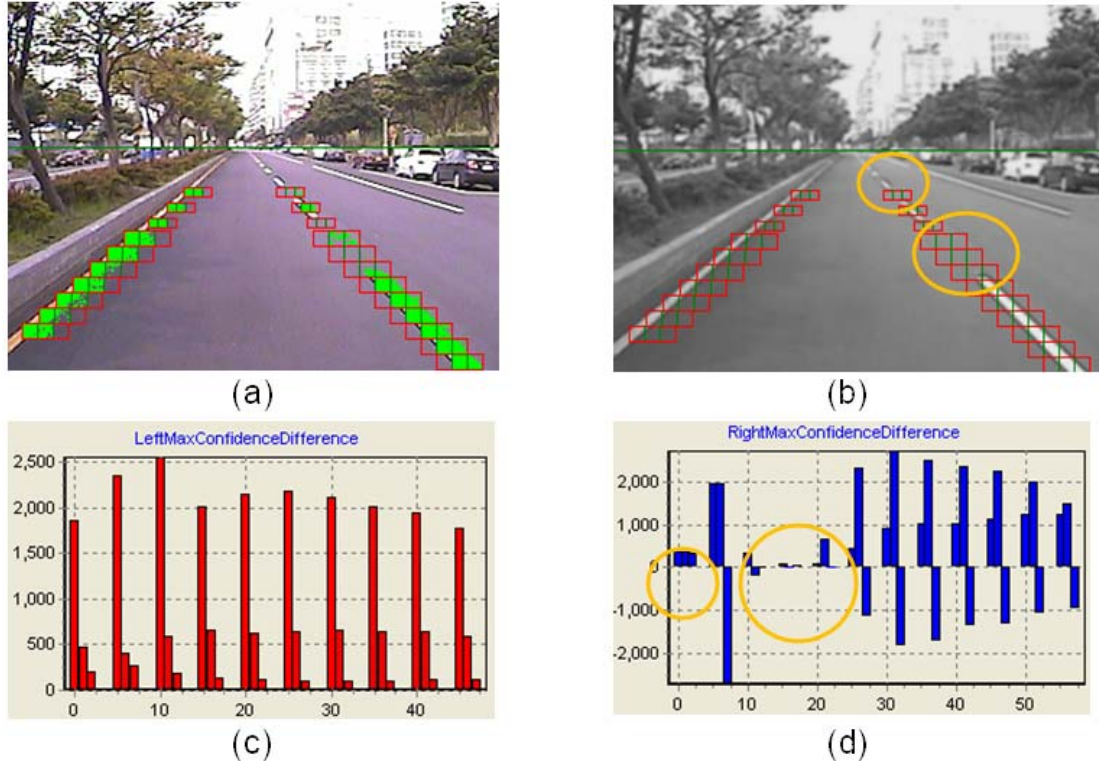


Fig. 4-14: Left boundary is continuous line with comprehensive boundary energy (c) and region ratio (a); The LMR boundary windows in yellow circle on right boundary (b) don't accumulate boundary features, such as boundary energy (d) and region ratio (a).

After accumulating M_n frames, it starts to align LMR boundary window from last position to current boundary position. Firstly, it needs to check whether the boundary has passed through the LMR boundary window. If there is no boundary passing through, the window will not accumulate boundary feature. For example, when boundary is constituted of broken lines, some LMR boundary windows are absence of boundary feature in M_n frames shown in Fig. 4-13. Therefore, we proposed an exclusive accumulator window for each LMR boundary window. The accumulator window collectively records pixels belonged to the boundary in a series of subsequent M_n frames. If any of the accumulator windows has detected the boundary as white pixels in each window shown in Fig. 4-15, it will process the boundary alignment (optimization). Alternatively, in an absence of boundary, it will

process boundary detection (searching). Next, we illustrate how to achieve boundary alignment (optimization) and boundary detection (searching) as follows I and II. Finally, if the LMR boundary window align to a new position, it will be verify by alignment checking as follows III.

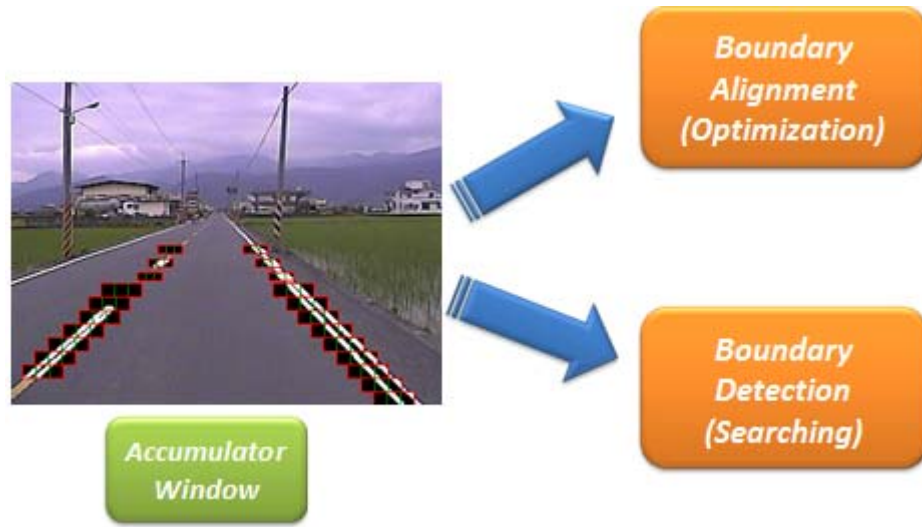


Fig. 4-15: Checking the accumulator window which records the boundary passing through the LMR boundary window. If any of the accumulator windows has detected the boundary, it will process the boundary alignment (optimization). Alternatively, it will process boundary detection (searching).

I. Boundary Alignment (Optimization)

Any of the accumulator windows has detected the boundary as white pixels in each window shown in Fig. 4-15, it will process the boundary alignment. This step is called optimization because we align the M block of LMR boundary window to position of boundary if the boundary deviates from each M block. When the boundary resides in M block, it is most stable. Therefore, the boundary alignment process is like the idea of optimization.

Firstly, we can utilize the boundary energy to check whether the boundary has deviated from M block. If maximum boundary energy is not resided in M block, it represent the deviation of boundary from the center of LMR boundary window. The

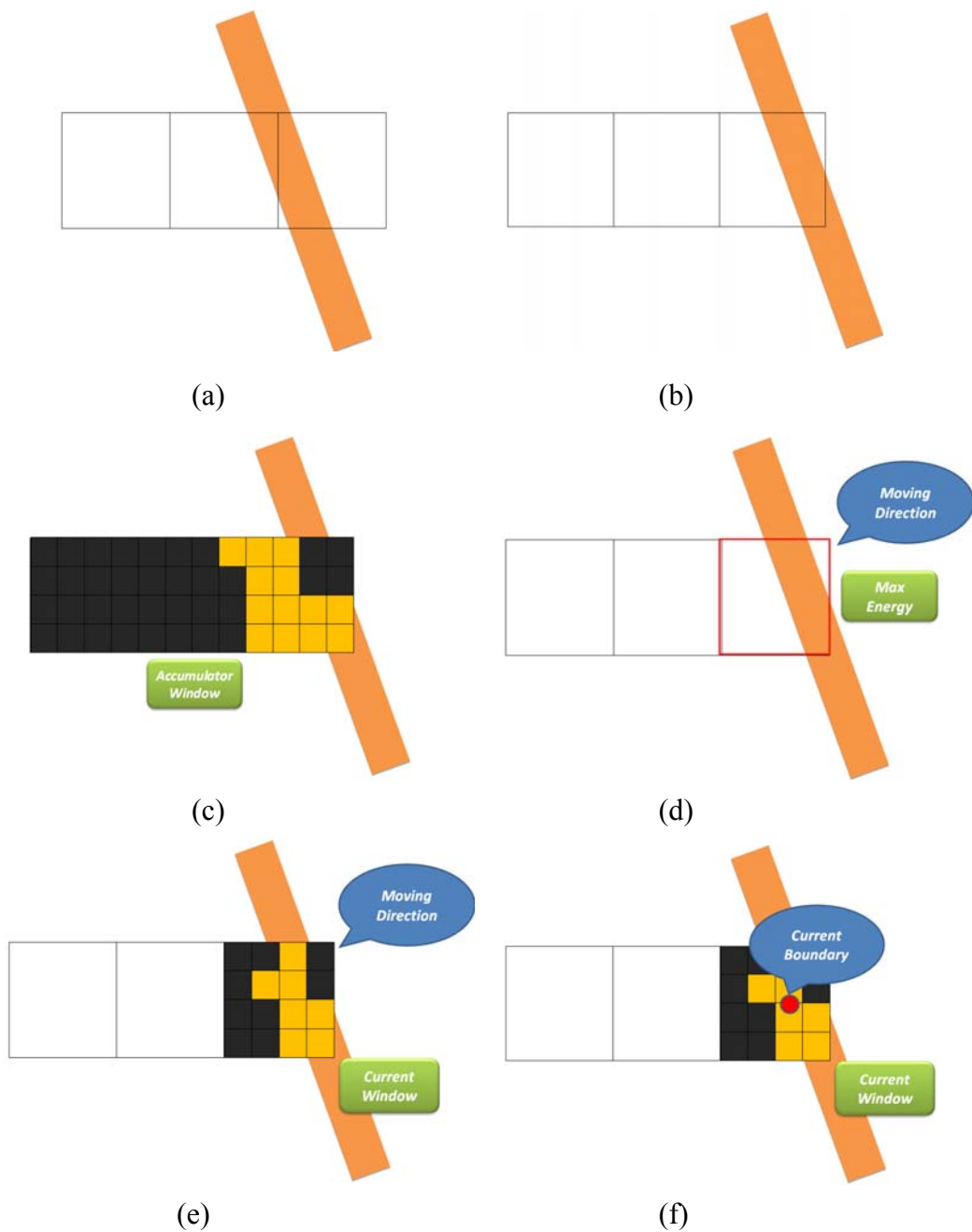
relative moving direction, $MovingDirection(i)$ shown in (4-4), of boundary and LMR boundary window is determined by where the maximum boundary energy resides. If the $MovingDirection(i)$ is either R(right) or L(left), we relocate the position of boundary by following the $MovingDirection(i)$. Next, we will discuss the relocate method.

$$\begin{aligned}
 MovingEnergy(i) &= Max (BEnergy(L), BEnergy(M), BEnergy(R)) \\
 MovingDirection(i) &= arg_i [MovingEnergy(i)] \\
 i &\text{ represent any of the LMR boundary window}
 \end{aligned} \tag{4-4}$$

The relocating process is tracking the boundary location in the last frame of a series of subsequent Mn frames. We use an example to explain the relocating process. When the boundary moves to the right side of a LMR boundary window shown in Fig. 4-16 (a) to (b), the accumulator window have detected the boundary moving trace as yellow pixels in Fig. 4-16 (c). Therefore it processes boundary alignment. The $MovingDirection$ of a LMR boundary window is detected as R(right) because of its maximum boundary energy shown in Fig. 4-16 (d). As a result we have realized the boundary moves to right side and we will track the boundary from the position of R block. Firstly, we detect whether the boundary resides R block of LMR boundary window in the last frame called current window checking. If the current window detects the boundary shown as yellow pixels in current window shown in Fig. 4-16 (e), we find the last position of boundary by the mean of the x-axis position of every boundary pixels. Finally, we align the LMR boundary window to the last position as Fig. 4-16 (g) to (h) to complete the boundary alignment (optimization) process.

Moreover, if the position of boundary moves extensively in a series of subsequent Mn frames during encountering sharp curvature, the last boundary position might deviate from LMR boundary window shown as Fig. 4-16 (i) to (j). Therefore the current window setting at R block will detected no boundary

appearance shown in Fig. 4-16 (k). Then we set the current window on the outer of R block shown in Fig. 4-16 (l) , the size of current window is same as the block of this LMR boundary window, and keep tracking forwarding MovingDirecition, right side, until the current window detect the boundary and using the same method mentioned above to align the LMR boundary window to the last position as Fig. 4-16 (g) to (h) to complete the boundary alignment (optimization) process.



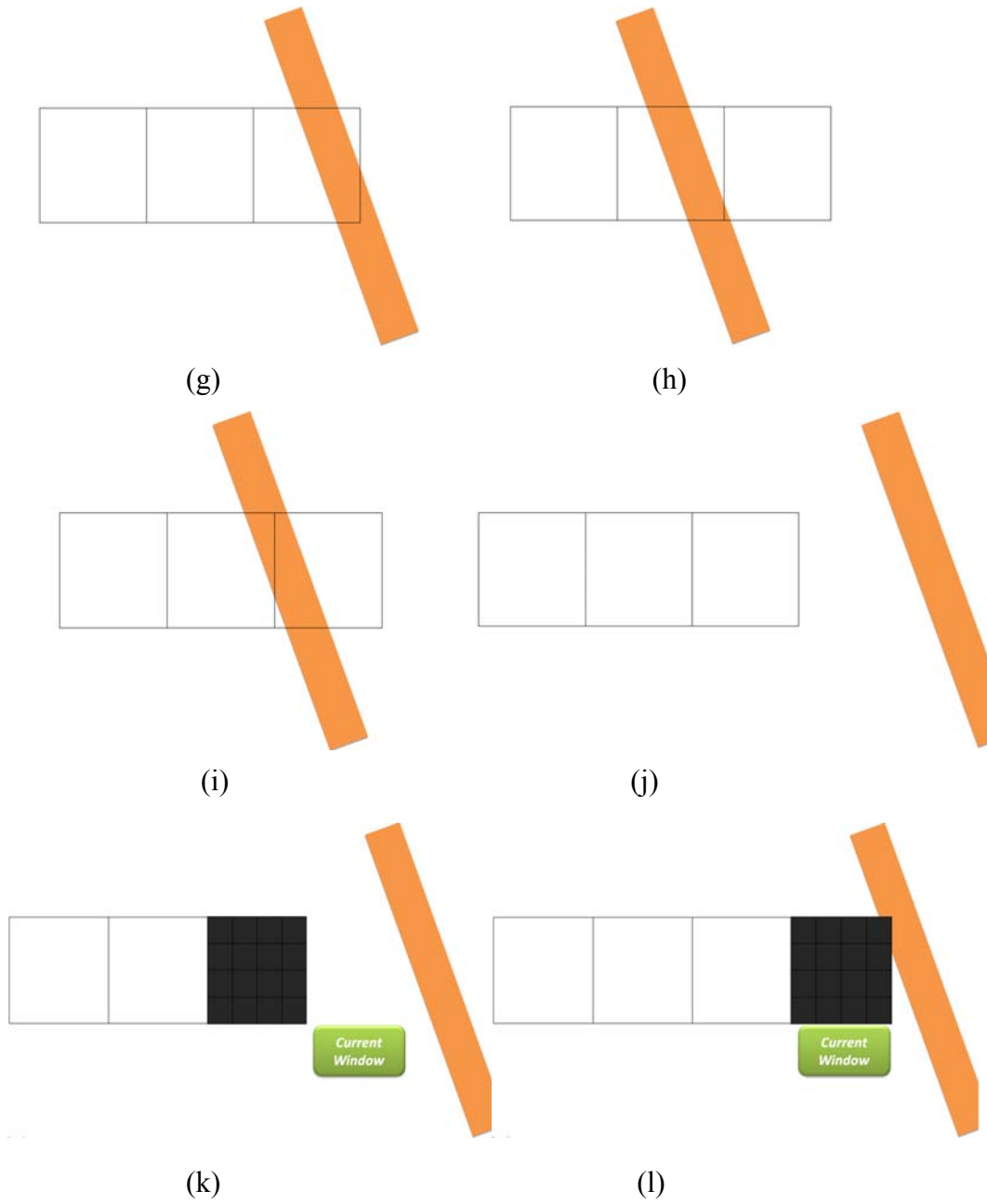


Fig. 4-16: The steps of boundary alignment (optimization)

II. Boundary Detection (Searching)

On the other hand, if there is no boundary passing through the LMR boundary window in the series of subsequent M_n frames, its accumulator window detects no boundary trace as yellow pixels in Fig. 4-16 (c) but as Fig. 4-17. This situation can be attributed by the following two reasons. Firstly, the boundary is constituted of broken line shown in Fig. 4-18 (a). Secondly, the boundary has deviated from LMR boundary

window shown in Fig. 4-19 (a). Therefore, it processes boundary detection (searching) procedure to find where the boundary deviates to or to check whether the broken line of boundary just have not passed through a LMR boundary window. We discuss the method in detail in the following.

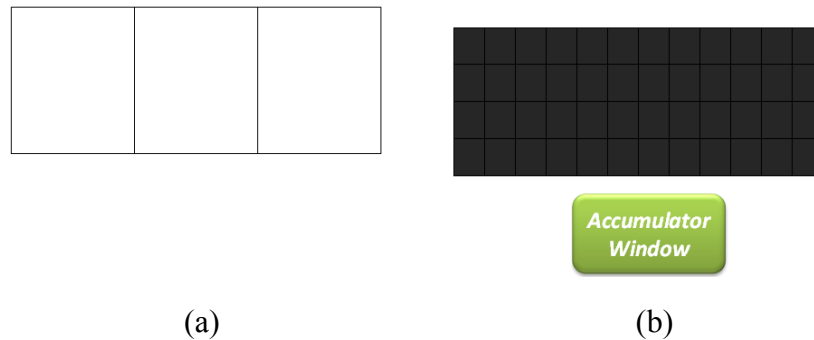
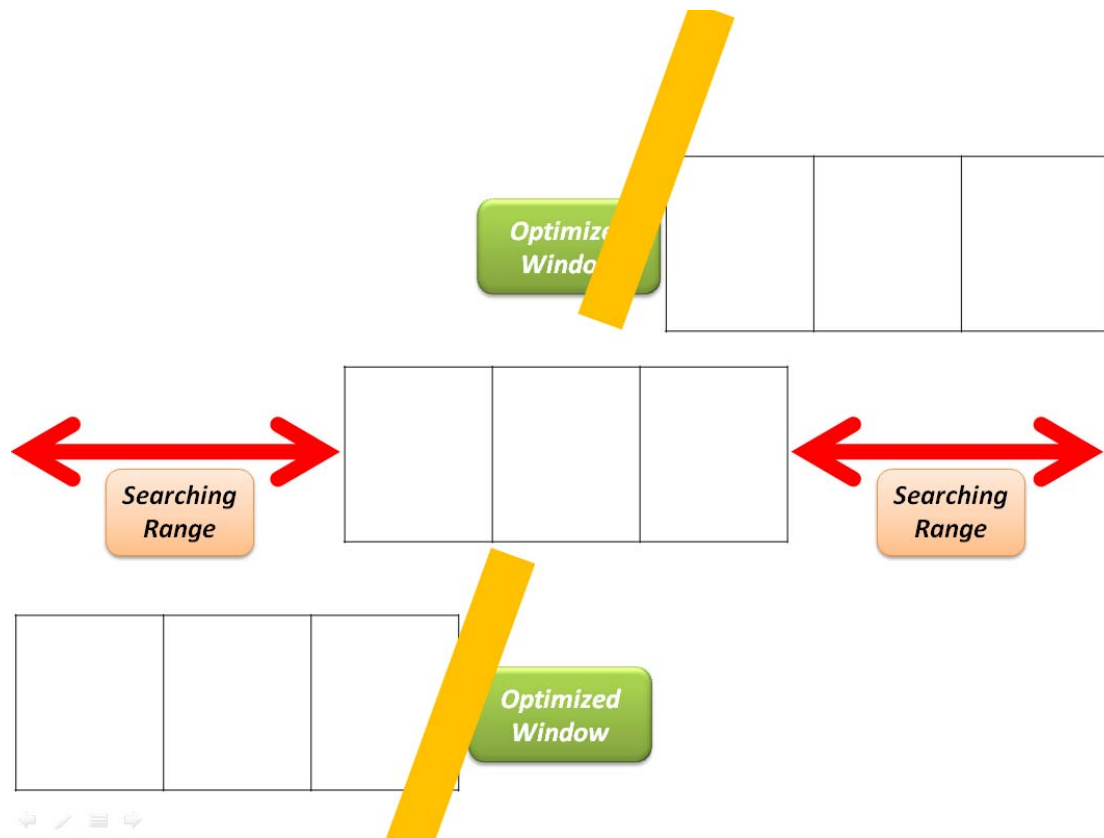


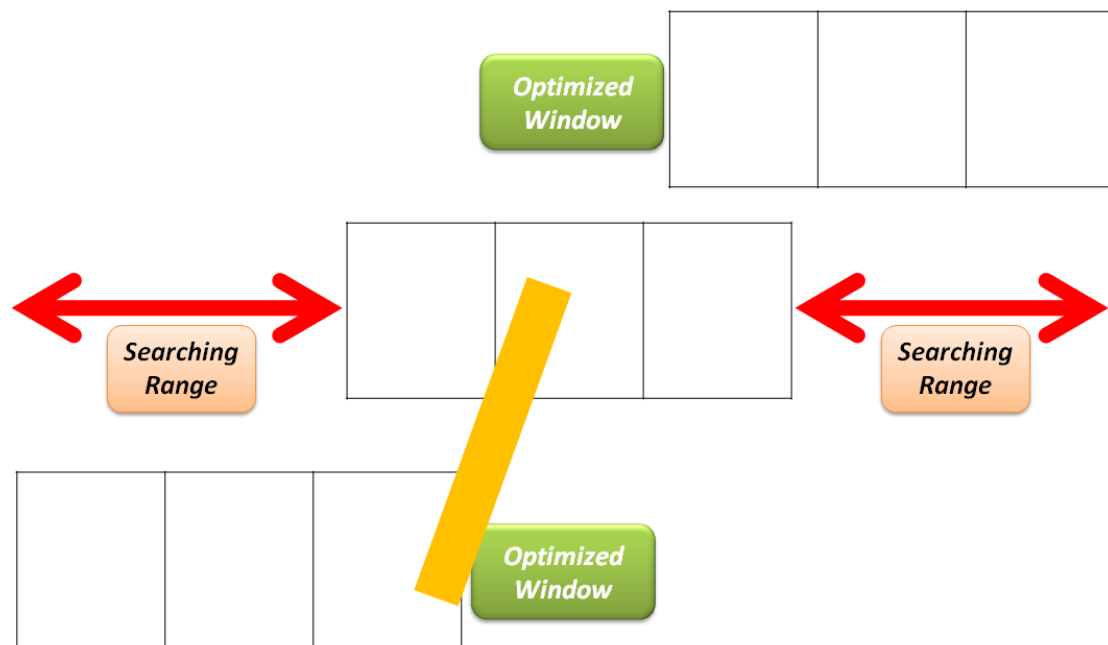
Fig. 4-17: (a) A LMR boundary and (b) its accumulator window with no boundary trace. Therefore, it performs the boundary detection procedure.

We utilize the last positions of far side and near side neighbor LMR boundary windows which have processed the boundary alignment (optimization) to complete searching process of the LMR boundary window. Firstly, we set the searching range for detecting the deviated boundary or to realizing the broken line condition. If the neighbor LMR boundary window which has process boundary alignment is right to it, the searching range is determined from the most extreme right side of this LMR boundary window to the center position of that neighbor LMR boundary window. On the other hand, If the neighbor LMR boundary window which has process boundary alignment is left to it, the searching range is determined from the most extreme left side of this LMR boundary window to the center position of that neighbor LMR boundary window. Secondly, we follow the similar tracking method to boundary alignment (optimization) process. The current window searches the boundary from near side to far side of the searching range. Once the current window detects the last boundary, we find the last position of boundary by the mean of the x-axis position of

every boundary pixels in the current frame. Finally, we align the LMR boundary window to the last position as shown in Fig. 4-19 (b). If the current can not find the boundary in the searching range, it represents that there is no boundary passing through the neighbor area in the series of subsequent M_n frames. It is because the segment of broken line boundary has not passed through yet. Therefore, The LMR boundary will remain at original position and wait for the incoming segment of the broken line shown in Fig. 4-18 (b).



(a)



(b)

Fig. 4-18: (a): The boundary is broken line so it detects no boundary appear in the searching range. (b): The LMR boundary will remain at original position and wait for the incoming segment of the broken line.

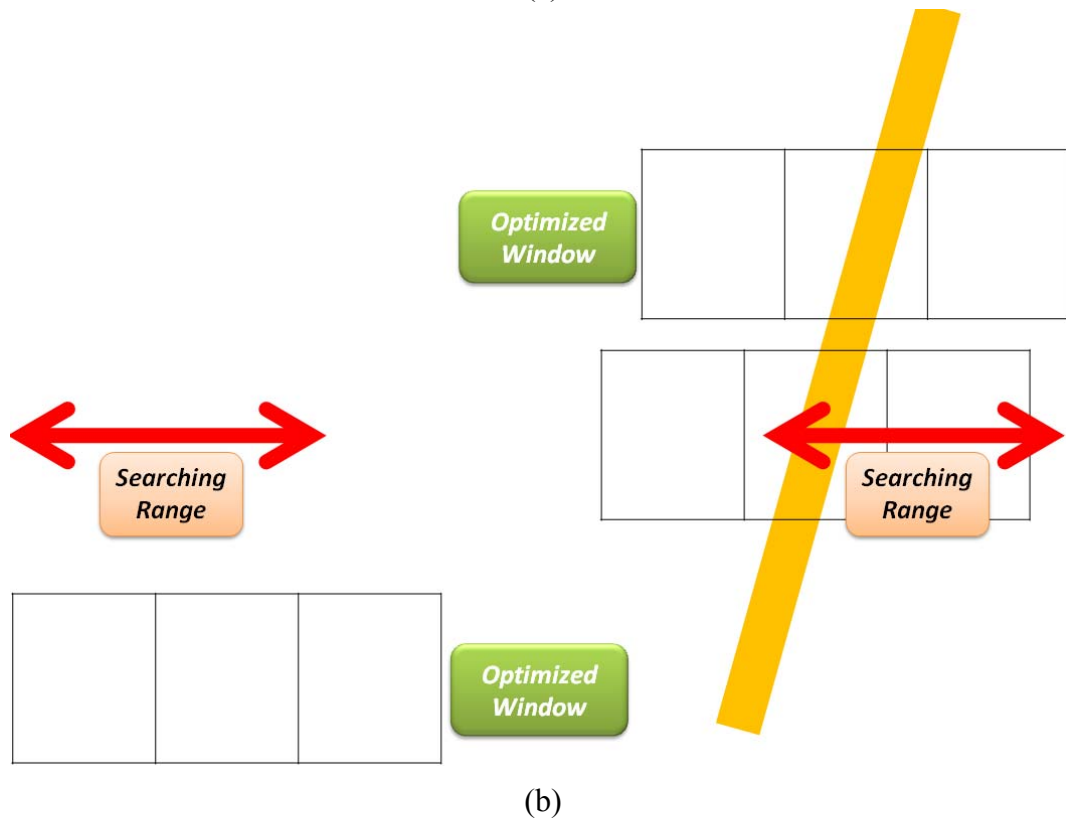
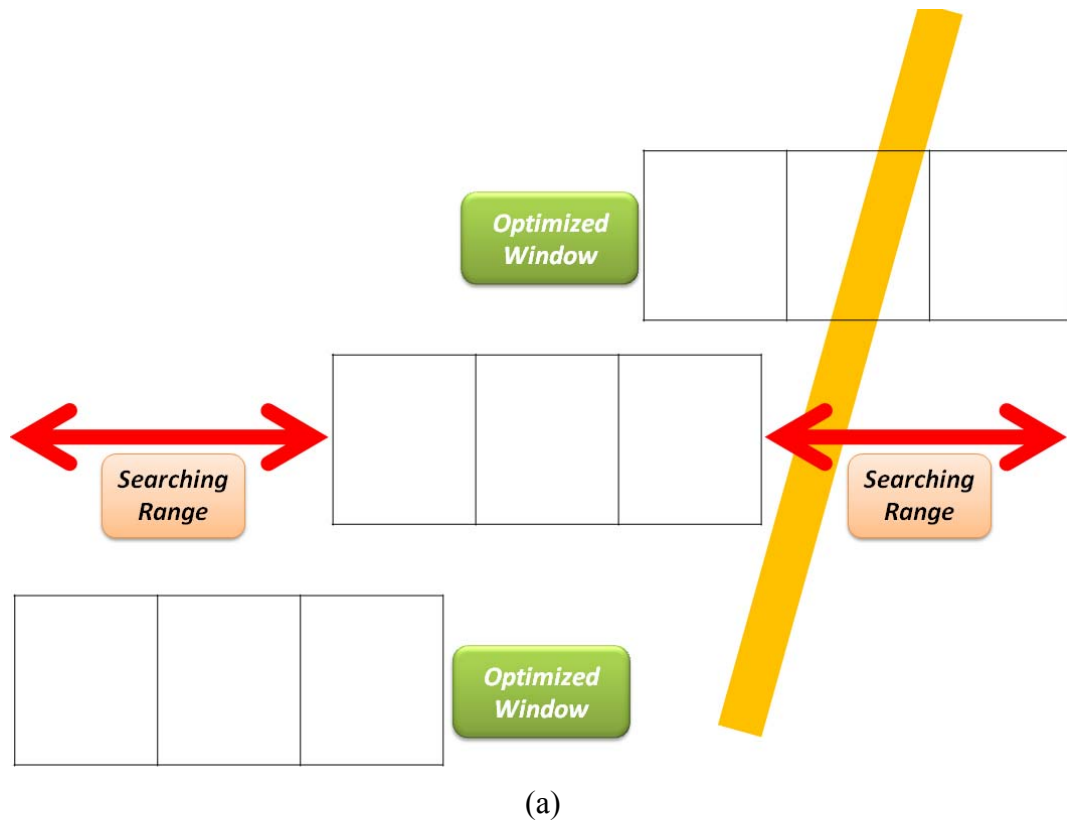


Fig. 4-19: (a): The boundary have deviated the range of LMR boundary window, so the accumulator window detected no boundary trace. (b): The LMR boundary tracks the last boundary position in the searching range and aligns to it.

III.Alignment checking

The boundary alignment (optimization) and boundary detecting (searching) just consider the boundary energy feature to realize the location and the relative moving direction of the boundary. The boundary energy categorized edge feature, but the feature is not robust enough when a LMR boundary window encounters strong edge intensity appearance which does not belong to road boundary. These edge are called fake edge, especially, the fake edge resulted from obvious shadow in the high light environment, a pond of water on the ground, and the objects around the path. These cause to mistake the boundary energy, so the LMR boundary will align to the wrong position. If a LMR boundary window correct alignment corresponds to position of road boundary, its ***Region Ratio Features, RoadPower*** and ***NonRoadPower***, will pass the alignment checking terms (4-5). On the contrary, one of the Region Ratio Features will be conflicted. If the new region ratio features pass the alignment checking, the region ratio will be updated in (4-6). We compare three examples to explain the problem of boundary alignment. In Fig. 4-20,

Fig. 4-21 and

Fig. 4-22, the pink area represents the road region inside of the driving path and the red region area is outside of driving path. The demarcation between the two areas is the driving path boundary. If the boundary energy is only attributed by road boundary, the LMR boundary window will align to the boundary. Then, the RoadPower and NonRoadPower can fit the thresholds, because the region around boundary always is with similar region ratio. Besides, the terms (4-5) take the last region ratio into account so it is very adaptive without any prior knowledge or predefined parameters. In Fig. 4-20, after alignment the RoadPower and NonRoadPower regress to about 90~100% and 65~80%. On the other hand, when LMR boundary window align to

wrong position, its region ratio must differ from previous one which fixed position on the road boundary shown as green blocks in Fig. 4-21 and Fig. 4-22. Therefore, the LMR boundary has to be checked by the Region Ratio Features after it alignment to boundary.

$$\text{If} \left(\begin{array}{l} |RoadPower(t+1) - RoadPower(t)| < TH_{Road} \\ or \\ |NonRoadPower(t+1) - NonRoadPower(t)| < TH_{NonRoad} \end{array} \right) \quad (4-5)$$

t+1: the region ratio of the LMR boundary window after alignment

t: the last region ratio of the LMR boundary passing the alignment checking

$$RoadPower(t) = RoadPower(t+1);$$

$$NonRoadPower(t) = NonRoadPower(t+1); \quad (4-6)$$

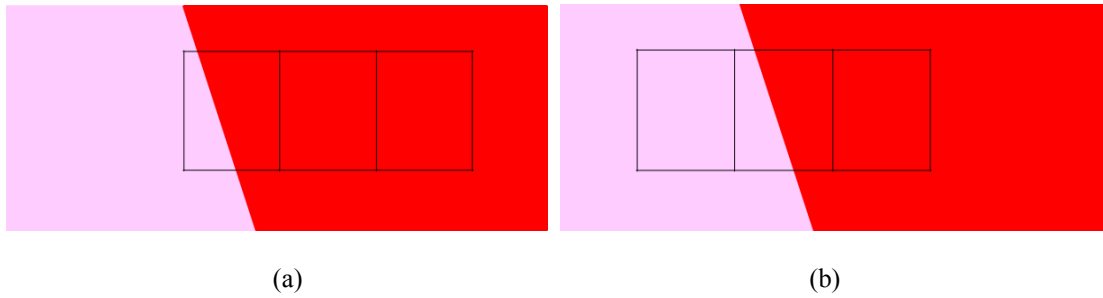
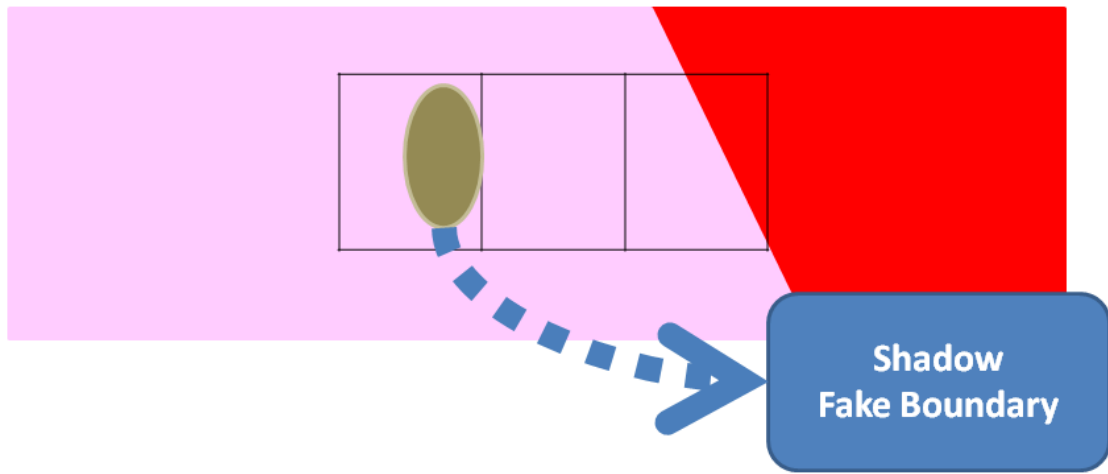
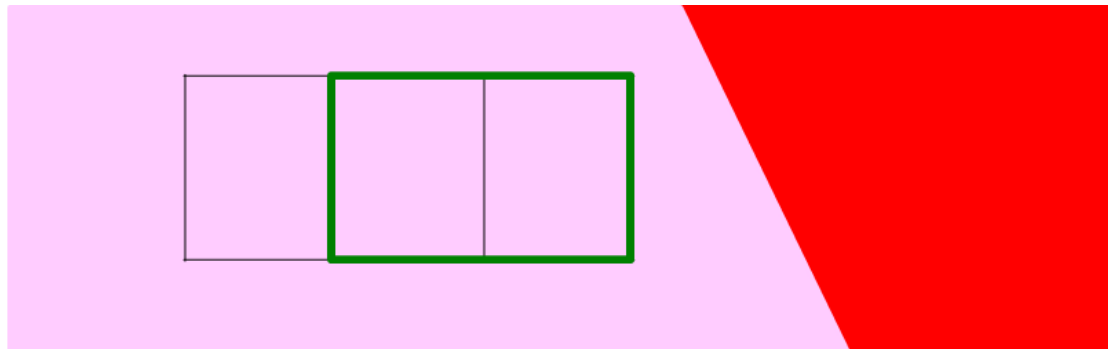


Fig. 4-20: (a): L block with the maximum boundary; (b): LMR boundary move toward left side to align to boundary. The correct alignment regress the region ratio, both **RoadPower and NonRoadPower**, to previous effective range.



(a)



(b)

Fig. 4-21: (a): The boundary alignment to left is in error because the fake edge resulted from the prominent shadow. (b): After the wrong alignment, the incorrect position of LMR boundary window can be checked by the *NonRoadPower* of *Region Ratio Feature* shown in green block.

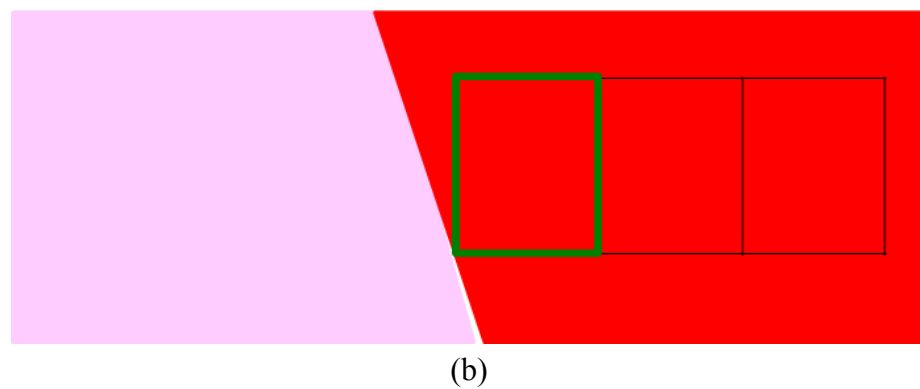
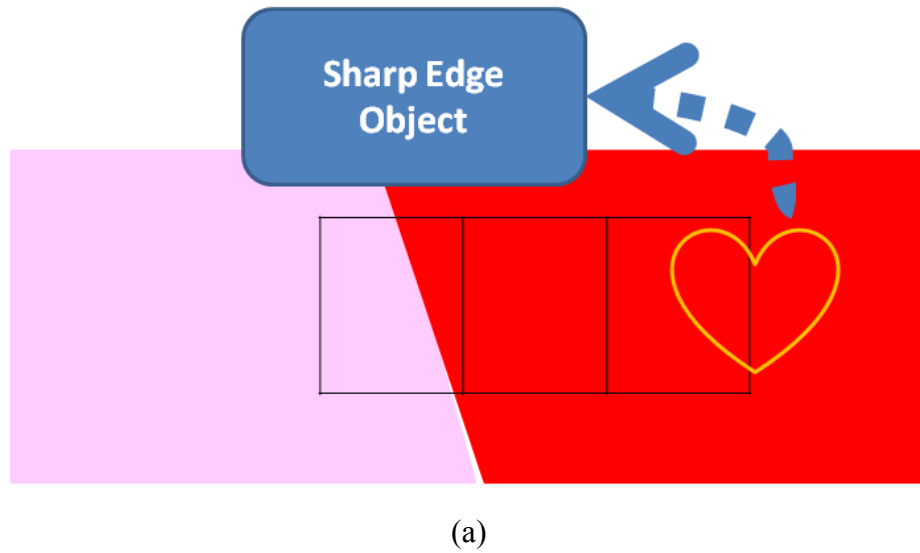


Fig. 4-22: (a): The boundary alignment to right is in error because the fake edge resulted from the object outside path. (b): After the wrong alignment, the incorrect position of LMR boundary window can be checked by the ***RoadPower*** of ***Region Ratio Feature*** shown in green block.

Chapter 5

Experimental Results

In this chapter, we will show our results of boundary detection and tracking algorithm in Section 5.1 and Section 5.2. We implemented our algorithm on the platform of PC with P4 2.66GHz and 1GB RAM. The software we used is Borland C++ Builder on Windows XP OS. All of the testing videos are uncompressed AVI standard files. The resolution of video frame is 320×240 .

We use CCD camera as the sensor which is mounted in the middle of the test vehicle's roof. The tested vehicle platform can be seen in Fig. 5-1.



Fig. 5-1: The tested vehicle

5.1 Experimental Results of Driving Path Boundary Detection

In the following , we show our experimental results of driving path boundary detection. It is the initial process in our system in order to find the boundaries and locate the LMR boundary window for tracking later. At first, we present the environment of the road as the left image of the raw in Fig. 5-2, and show the location of LMR boundary windows set by the driving path boundary detection algorithm as

the right image of the raw in Fig. 5-2. In Fig. 5-2, they include many type of road boundary and succeed in detecting the boundary in the night where is with or without street light. Besides, this initial process can work when encountering the curl road due to the modified methods proposed.



(a)



(b)



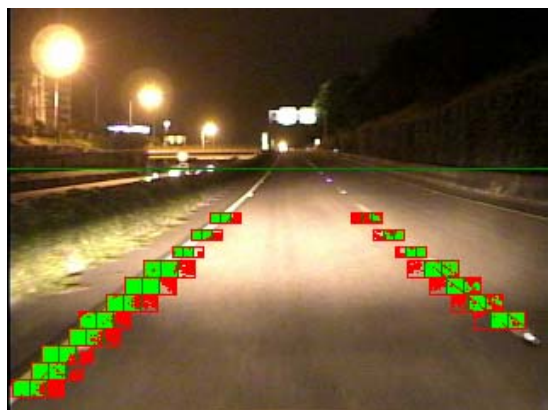
(c)



(d)



(e)



(f)



(g)



(h)



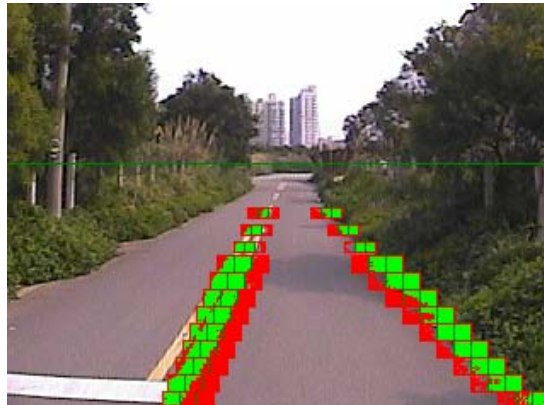
(i)



(j)



(k)



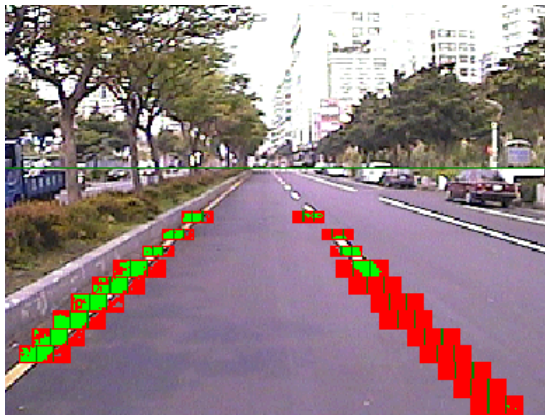
(l)

Fig. 5-2: Experimental results of driving boundary detection.

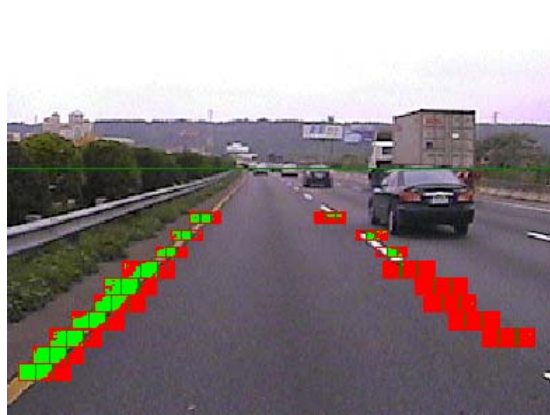
5.2 Experimental Results of Boundary Tracking

Here, we demonstrate some boundary tracking examples. In Fig. 5-3 we can see that the system can tracking steadily by align the LMR boundary to the last boundary

position. Moreover, it can work when encountering sharp curve (d)(e)(f) or in the night with or without street light (f)(h). If the vehicle is changing the line, the LMR boundary windows will follow the new path boundary to process the alignment. For example, when the vehicle attempts to change to the right line, the LMR boundary windows on the original right side will transfer to track the left boundary. On the other hand, the new right boundary appears on the right side of current lane, so the LMR boundary window will detect it immediately and continue tracking show in Fig. 5-4



(a)



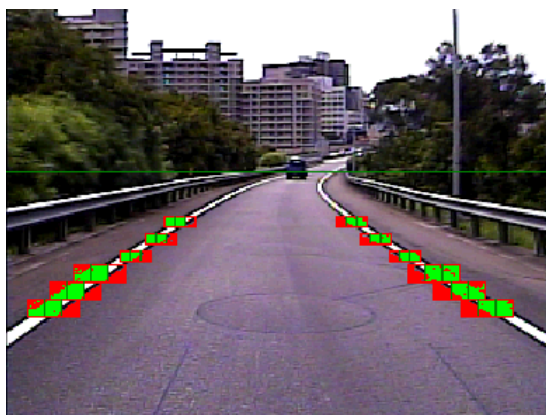
(b)



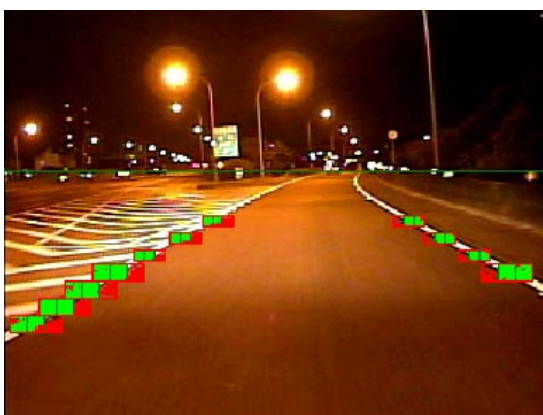
(c)



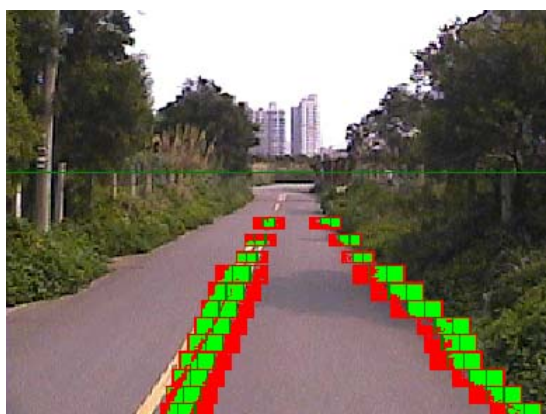
(d)



(e)



(f)

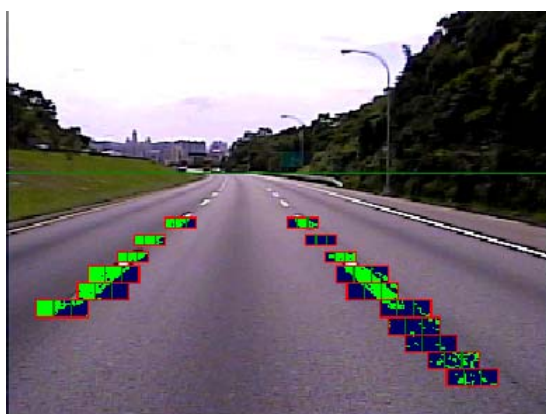


(g)

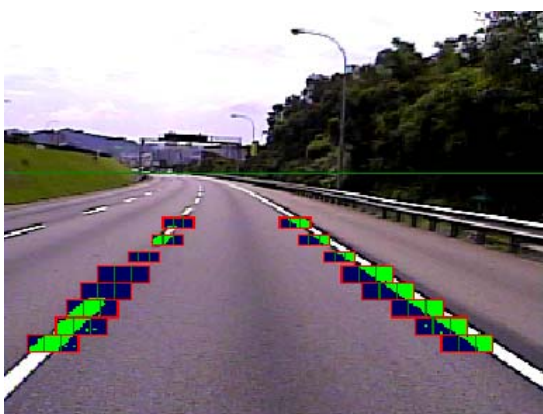


(h)

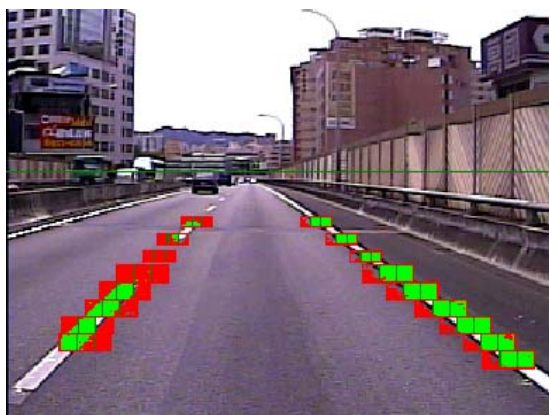
Fig. 5-3: Experimental results of boundary tracking



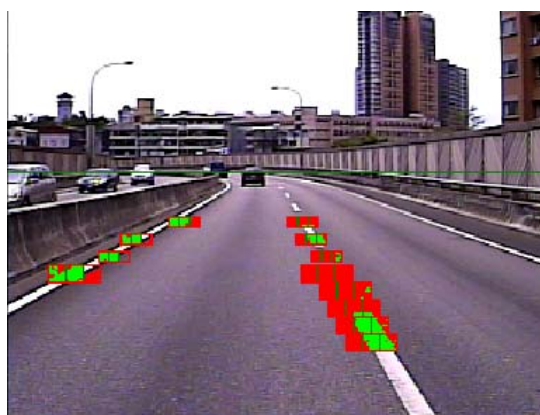
(a)



(b)



(b)



(d)

Fig. 5-4: Experimental results of change line situation.

Chapter 6

Conclusion

We present a new method of efficient road following to achieve real-time performance. At first, we locate the main driving path precisely, and then perform the boundary tracking algorithm. In the algorithm, we utilize the color distribution and edge difference features in the proposed LMR boundary window. By on-lined L*a*b color model, we can extract the road color region and observe the color distribution of the LMR boundary window. This approach is accomplished by using the proposed region ratio feature. We also present a boundary energy feature to find the most distinct edge difference which most matches the boundary property. Using these both main features, we can track the boundary efficiently and effectively. Therefore, the system integrates the concepts of region-based and boundary-based to realize the whole processing.

Experiments were conducted on different scenes which including the different type of road boundaries. The system succeeds in detecting and tracking the regular type of boundary with obvious lane line or complicated types of boundary including broken lane lines, absence of lane line and various types of country roads. Our system can also accommodate the lane change movement and relocate the new main driving path boundary. Through the experiment, we realize the overall excursion time is very swift and can handle driving in the night. In the future, we believe that when the system ports to the platform; it can achieve real-time performance as well.

6.1 Contribution

In this paper, we proposed a new method which combines color feature and edge feature. Therefore, we enhance the preciseness of the system. In addition, our algorithm has modified most disadvantages of the region-based and boundary-based methods. Firstly, we proposed the guide snake to sample the nearest region of each LMR boundary window. Therefore, it can solve the problem when encountering non-homogenous surface. Besides, the L^*a^*b color feature can resist the light variation and possess great luminance difference toleration. After applying the updating module, we can train the on-lined color model to increase its plasticity. As a result, the adaptive model can preserve accurate detection within unstable feature condition. On the other hand, we decrease the region of interest to LMR boundary windows so we can enhance the speed of computation time. Specifically for these disadvantages, we summarize the modification we mentioned above shown in Fig. 6-1. Secondly, we solve the main problems of boundary-based methods. The majority of the problems are caused by unconstructed road without lane line, strong edge of non-boundary objects, and the sharp curvature on the road. In our system, we focus solely on the LMR boundary windows and adopt the general boundary feature so we can prevent these problems occurred. Because the LMR boundary windows locate near by road boundary, it would not interfere with other stronger edge of the non-boundary objects. Furthermore, our tracking algorithm can effectively handle many complicated scenes such as broken lane line, and unclear boundary. Some boundary-based methods need many parameters and horizon position when converting to bird view image by inverse perspective mapping (IPM) or processing model template matching. It is very impractical due to it requires many prior knowledge. In fact, the pre-defined parameters and knowledge are not necessary in

our system. Specifically for the above disadvantages of boundary-based methods, we summarize the modification we mentioned above shown in Fig. 6-2.

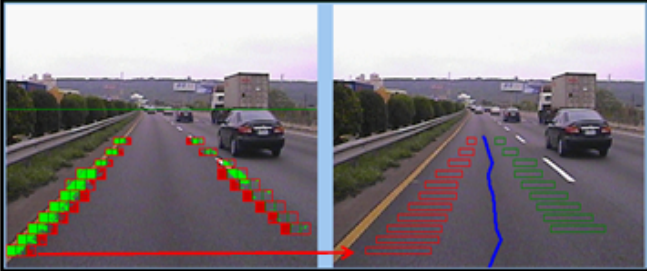
Feature	Disadvantage
<ol style="list-style-type: none"> 1. Color Feature 2. Texture Feature 	<ol style="list-style-type: none"> 1. Non-homogeneous 2. Light variation 3. Feature is not stable 4. Consuming Time
Modification (By On-lined Color Model)	
<ol style="list-style-type: none"> 1. <i>guide snake</i> 2. <i>L*a*b resist light variation</i> 3. <i>On-line updating module</i> 4. <i>Blob noise check</i> 5. <i>ROI select</i> 	

Fig. 6-1: The main features and disadvantages of region-based methods. Our algorithm can modify and solve the problems.

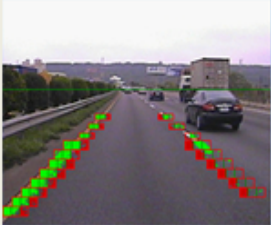
Feature	Disadvantage			
<ol style="list-style-type: none">1. Edge Feature2. Geometric Feature	<ol style="list-style-type: none">1. Unstructured road such as unmarked paved road2. Non-continuous markings3. Horizon is not stable when encounter camera shaking4. Edge produced by non-boundary5. Sharp curvature make incorrect6. IPM needs different parameters			
Modification (By setting LMR boundary window as ROI)				
<ol style="list-style-type: none">1. LMR boundary window2. Effective tracking algorithm3. Without parameters	 <table><tr><td>L</td><td>M</td><td>R</td></tr></table>	L	M	R
L	M	R		

Fig. 6-2: The main features and disadvantages of boundary-based methods. Our algorithm can modify and solve the problems.

6.2 Future Works

To further improve the performance and the robustness of our algorithm, some enhancements or trials can be made in the future. Firstly, if the edge boundary is very ambiguous, the edge information will be inconsistent. Therefore, we can aim at the LMR boundary windows to enhance the edge feature by stretching gradient difference. Secondly, we can express the boundary location more visually by using the least square error technique. The least square error method processes the boundary candidate pixels and finds the optimal boundary contour of linear or parabolic model which is with minimum errors. Thirdly, we can exploit the boundary detection results to realize the lateral departure warning (LDW) function and integrate into our system. Finally, with the effectiveness of locating boundary, the driver can clearly understand the vehicle position. Therefore, we will consolidate the proposed road features and the ability of driving path boundary detection to continue to research the obstacles and vehicles detection. According the boundary position, the system can comprehend the relative position, frontal or lateral, between the vehicle and these obstacles, Moreover, the distance estimation from the vehicle to the frontal or lateral objects is another important research topic. After accomplishing the overall functions mentioned above, the system will very applicable and we will transplant from PC based to DSP platform for more commercial certification and appealing.

References

- [1] J. Crisman and C. Thorpe, SCARF: A Color Vision System that Tracks Roads and Intersections, Conf. Proc. IEEE International Conf. on Robotics and Automation, IEEE, 1993
- [2] J. Crisman and C. Thorpe, UNSCARF: A Color Vision System for the Detection of Unstructured Roads, Conf. Proc. IEEE International Conf. on Robotics and Automation, 1991.
- [3] J. Huang and B. Kong, A New Method of Unstructured Road Detection Based on HSV Color Space and Road Features, International Conf. on Information Acquisition, 2007.
- [4] Y. Alon, A. Ferencz and A. Shashua, Off-road path following using region classification and geometric projection constraints, Computer Vision and Pattern Recognition, IEEE Computer Soc. Conf, 2006.
- [5] C. Rasmussen, Grouping dominant orientations for ill-structured road following, Proc. IEEE Soc. Conf. Computer Vision and Pattern Recognition, 2004.
- [6] Y. He, H. Wang, and B. Zhang, Color-Road Detection in urban traffic Scenes, IEEE Trans. Intelligent Transportation System, 2004.
- [7] D. Guo and M. Xie, Color Modeling by Spherical Influence Field in Sensing Driving Environment, IEEE Intelligent Vehicles Symposium, 2000.
- [8] C. Tan, T. Hong, T. Chang, and M. Shneier, Color Model-based Real-Time Learning for Road Following, IEEE Intelligent Transportation Systems Conf. 2006.
- [9] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, Vision and Navigation for Carnegie Mellon Navlab, Annual Review of Computer Science, 1999.
- [10] M. Bertozzi and A. Broggi, GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection, IEEE Trans. on Image Processing, 1998.
- [11] M. Bertozzi, A. Broggi, and Alessandra Fascioli, Stereo Inverse Perspective Mapping: Theory and Applications, Image and Vision Computing Journal, 1999.
- [12] Y. Wang, D. Shen, and E.K. Teoh, Lane detection using spline model, Pattern Recognition Letter, 2000.
- [13] Y. Wang, E.K. Teoh, D. Shen, Lane detection and tracking using B-Snake, Image and Vision Computing, 2003.
- [14] C.R. Jung and C.R. Kelber, Lane following and lane departure using a linear-parabolic model, Image Vision Computing, 2005.

- [15] J.W Park , J.W Lee, and K.Y Jhang, A lane-curve detection on an LCF, Pattern Recognition Letters, 2003.
- [16] E.J. Carmona, J. Martinez-Cantos, J. Mira, A new video segmentation method of moving objects based on blob-level knowledge, Pattern Recognition Letters, 2008.
- [17] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, Color transfer between images, IEEE Computer Graphics and Applications, 2001.
- [18] J.M. Alvarez, A. Lopez, and R. Baldrich, "Illuminant-Invariant Model-Based Road Segmentation" in IEEE Intelligent Vehicles Symposium, 2008.
- [19] C. Stauffer and W.E.L. Grimson,, Adaptive background mixture models for realtime tracking, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.
- [20] Taiwan Institute of Transportation, MOTC, <http://www.iot.gov.tw/>
- [21] AutoBlog, <http://www.chinese.autoblog.com/>