

# 國立交通大學

## 電機與控制工程學系

### 碩 士 論 文

利用車道資訊與停等特徵達成預先交疊偵測與解析  
之即時多車輛偵測與追蹤系統



A Real-Time Multiple-Vehicle Detection and Tracking System  
with Prior Occlusion Detection and Resolution by Lane  
Information and Queue Features

研 究 生：陳元馨

指 導 教 授：吳炳飛 教授

中 華 民 國 九 十 五 年 七 月

利用車道資訊與停等特徵達成預先交疊偵測與解析之即時多車輛偵測與追蹤系統

A Real-Time Multiple-Vehicle Detection and Tracking System with Prior Occlusion Detection and Resolution by Lane Information and Queue Features

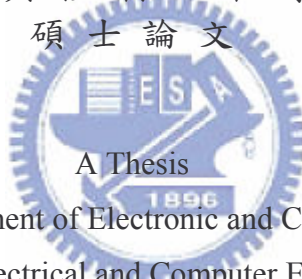
研究生：陳元馨

Student : Yuan-Hsin Chen

指導教授：吳炳飛

Advisor : Bing-Fei Wu

國立交通大學  
電機與控制工程學系  
碩士論文



A Thesis

Submitted to Department of Electronic and Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Electronic and Control Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

# Awards:

- 第十四屆全國自動化科技研討會最佳論文獎入圍



# Publication:

學術期刊 論文	1、Bing-Fei Wu, Shin-Ping Lin, Chung-Cheng Chiu, Chao-Jung Chen, and <u>Yuan-Hsin Chen</u> , "A Real-Time Multiple-Vehicle Detection and Tracking System for Traffic Monitoring"(submit to Elsevier)
學術會議 論文	<p>1、Bing-Fei Wu, Yen-Lin Chen, and <u>Yuan-Hsin Chen</u>, "Nighttime Vehicle Detection for Driver Assistance and Autonomous Vehicles", in <i>Proceedings of the 18<sup>th</sup> International Conference on Pattern Recognition</i>, Hong Kong, August, 2006.</p> <p>2、Bing-Fei Wu, Shin-Ping Lin, and <u>Yuan-Hsin Chen</u>, "A Real-Time Multiple-Vehicle Detection and Tracking System with Prior Occlusion Detection and Resolution, and Prior Queue Detection and Resolution" in <i>Proceedings of the 18<sup>th</sup> International Conference on Pattern Recognition</i>, Hong Kong, August, 2006.</p> <p>3、Bing-Fei Wu, Yen-Lin Chen, and <u>Yuan-Hsin Chen</u>" A Fast Intelligent Nighttime Vehicle-Light Recognition System Based on Computer Vision ", Vol. 2, pp.J-29-34, <i>Proceedings of 第十四屆全國自動化科技研討會</i> , Taiwan, June, 2006.</p> <p>4、Bing-Fei Wu, Shin-Ping Lin, and <u>Yuan-Hsin Chen</u> "A Real-Time Multiple-Vehicle Detection and Tracking System with Prior Occlusion Detection and Resolution", in <i>Proceedings of the 5<sup>th</sup> IEEE International Symposium on Signal Processing and Information Technology</i>, pp. 311-316 ,Athens, Greece, Dec. 2005.</p> <p>5、吳炳飛, 吳玉珍, 陳昭榮, 林欣平, <u>陳元馨</u>, 卓訓榮, 李霞, "The Study of Intelligent Vision Based Vehicle Detection", in <i>Proceedings of 中華民國運輸學會第20屆論文研討會</i>, pp. 951-966, Nov. 2005.</p> <p>6、Bing-Fei Wu, Yen-Lin Chen, <u>Yuan-Hsin Chen</u>, Chao-Jung Chen, and Chuan-Tsai Lin, "Real-Time Image Segmentation and Rule-Based Reasoning for Vehicle Head Light Detection on A Moving Vehicle", in <i>Proceedings of the 7<sup>th</sup> IASTED International Conference on Signal and Image Processing (SIP 2005)</i>, pp. 388-393, Honolulu, Hawaii, USA, Aug. 2005.</p>
專利	1、吳炳飛, 陳彥霖, <u>陳元馨</u> , 陳昭榮"一個基於電腦視覺的智慧型高速夜間車燈辨識系統"
研究計畫	<p>1、"汽車應用影像處理及辨識技術" 車輛研究測試中心(ARTC), 2004</p> <p>2、"國道替代道路路況資訊擴充之研究與實作" 交通部運輸研究所 (IOT), 2005</p>

## ● ”國道替代道路路況資訊擴充之研究與實作”新聞稿



交通部  
運輸研究所

新聞稿 94.9.9

### 國內自行研發影像式車輛偵測器技術突破 交通資訊普及將指日可待

交通部運輸研究所為了提供民眾更完善的路況資訊，特於近年著手研發適用國內複雜交通環境之車輛偵測器，其中影像式車輛偵測器已有重大突破，不但準確率提昇，並具機車辨識及自動化設定的功能，尤其大幅降低設備成本，對於未來普遍設置偵測器以提供塞車資訊及提昇產業競爭力皆有莫大助益。

車輛偵測器為收集交通量與速率等路況資訊之重要基礎設備，透過自動化車流偵測，不但有利交通管理，亦可提供民眾所關心的路況資訊。國內目前除了高速公路之偵測器佈設較為完整外，省縣道與都市道路則因偵測器價格昂貴及不易辨識機車的限制，而導致佈設數量不足。長久以來，國內使用的車輛偵測器多採用國外產品，不僅單價偏高，且維護不易，同時國外產品對於國內複雜的交通環境（如機車）在偵測上更有所限制。有鑑於此，94 年度交通部運研所與交通大學合作研發適用於臺灣交通環境特性並具合理成本之車輛偵測器，本年度以影像及微波式偵測技術為研發重點，而影像式車輛偵測器已成功完成研發。

交通部運研所研發之影像式偵測器係透過道路監視設備的影像辨識運算技術判別路況，其交通量辨識準確率達九成八以上，除可辨識一般汽車外，亦可適用於機車，同時可偵測雙向多車道，其功能較國外產品更適用於國內環境，並具有智慧型自動偵測道路環境特性能力。另在偵測區域內可對每一車輛全程追蹤其軌跡，若與交通執法系統結合，將可對車輛跨越分向限制線（雙白線）等違規行為予以有效取締。運研所表示，此項計畫研發成果未來經過技術轉移，與國內廠商合作達產品化階段後，將可投入市場大量生產與建置，創造上億元的產值，有利國內光電、通訊、資訊加值服務等相關產業的發展，未來在大幅降低偵測器成本而在一般道路普遍設置後，將可自動、即時地反映路況，藉由廣播、網站、導航系統與路側可變標誌提供民眾最新的交通訊息。

附件：交通部運輸研究所研發影像式車輛偵測器功能特色及畫面說明

新聞聯絡人：李霞  
電話：02-23496886  
e-mail：[lisa@iot.gov.tw](mailto:lisa@iot.gov.tw)

# 利用車道資訊與停等特徵達成預先交疊偵測與解析之即時多車輛偵測與追蹤系統

學生：陳元馨

指導教授：吳炳飛 博士

國立交通大學電機與控制工程學系 碩士班

## 摘要

道路車輛資訊的蒐集在先進交通控制系統中是不可或缺的重要環節；而車輛偵測系統為收集交通流量與行車速度等路況資訊之重要基礎設備。本論文提出一套利用車道資訊與停等特徵，達成預先交疊偵測與解析之即時多車輛偵測與追蹤系統；將前視(Forward-Looking)的彩色影像輸入動態前背景切割處理後，可將移動物體從影像中切割出，接著利用車輛追蹤技術，紀錄車輛在偵測範圍中的路徑；為了提高追蹤的正確率，我們利用自動車道偵測所得到之車道線資訊，提出了以車道為基礎的去陰影方法以及交疊偵測與切割。在得到車輛的追蹤軌跡之後，此系統可計算並提供行車速度、車流量統計等交通參數，並可分別偵測與辨識出大車、小車及機車，以及得到車輛的移動方向。除此之外，本研究亦提出了一套停等車輛偵測及分割演算法，能有效地解決車輛在停紅綠燈或是塞車的情況中，因為交疊而無法保留追蹤路徑問題，使得車輛追蹤能夠持續且正確地進行，以便後續交通參數的計算。

# A Real-Time Multiple-Vehicle Detection and Tracking System with Prior Occlusion Detection and Resolution by Lane Information and Queue Features

Student : Yuan-Hsin Chen

Advisor : Dr. Bing-Fei Wu

Department of Electronic and Control Engineering  
National Chiao Tung University

## ABSTRACT



A multiple-vehicle detection and tracking (MVDT) system with prior occlusion detection and resolution by lane information and queue features has various applications – tracking and classifying vehicles, determining traffic parameters, moreover, detecting violations of vehicles. The characteristics of MVDT system are real-time operation, robustness, precision, and ease of setup which are all important consideration for a vehicle detector. In this study, the proposed tracking reasoning is applied to track vehicles after the dynamic segmentation firstly. Next, some functional methods, such as lane-based run-length shadow suppression and prior lane-based occlusion detection and resolution, are proposed to enhance the accuracy of the tracking processing. Then, the edge-based queue detection and resolution is exploited to keep tracking trajectories when vehicles are waiting at the traffic lights or at traffic jam. Furthermore, tracking trajectories and the lane mask are applied to derive the values of traffic parameters and the direction of vehicle movement. Finally, for ease of setup, the adaptation of the system parameters is proposed.

# Acknowledgement:

首先，要感謝我的指導教授 吳炳飛博士二年來的教導，感謝您提供豐沛的研究資源，讓我的研究能順利進行，並讓我接計畫及寫 paper，給了我很多磨練的機會，讓我不管在專業或是心理層面上都有顯著的成長。另外，要感謝我的大學專題指導教授 蘇崇彥博士，是您帶領我進入影像處理的領域，您認真執著的態度，是我學習的榜樣。

此外，最要感謝的是欣平學長，在你耐心的教導下，讓我寫程式和解決問題的能力大大提升，每個禮拜的 meeting 時間，都能在愉快的討論下解決問題，並分享生活上的趣事和困難，你真的是一個很讚的 leader。還有，要感謝阿霖學長，雖然我們只合作一個多學期，但你對研究的熱情和堅持，讓我非常非常敬佩，你在研究上對學弟妹非常照顧並會盡力給予協助，讓我們節省了很多嘗試錯誤的時間，在生活上，會給予我中肯的建議，你真的是一個非常好的學長。另外，要感謝很照顧我的忠哥、旭哥、馬哥、全財學長、志鴻學長、鵬文學長和益賓，還有在實驗上給予協助的阿誠學長。

接著，要感謝我親愛的學姊映伶、同學宗堯和學長晏阡、重甫、培恭，有你們的陪伴，讓我待在新竹的日子，多了很多的樂趣，永遠都不會忘記我們一起看球、一起吃大餐、一起看電影、一起唱歌、一起打拼的日子，要當一輩子的好朋友喔！另外，要謝謝 CSSP 實驗室的伙伴們：leader 信元、一起打拼的同學們：岑偉、小熊、子萱、ppj、皓昱，熱心幫忙的可愛學弟們：秉宗、敏偉、晉源、至明、家維和崇瑞，和在交接很幫忙的志忠。

最後，感謝我親愛的父母，有你們無怨無悔的支持，我才能順利完成這篇論文，謝謝你們在我忙碌時，專程帶食物到新竹陪我吃飯，在我遇到挫折時，不斷給我鼓勵，並且給予我充分的資源，讓我去追逐我的夢想，在我心中，你們是最偉大的父母。還有我最貼心的妹妹，謝謝你在我在繁忙的研究生活中，付出更多的心力照顧關心家裡的事；還有要感謝我最愛的阿公阿媽和阿姨的支持。當然，不會忘了感謝我最可愛最溫柔體貼的男朋友 張嘉宏，謝謝你在我心情低落的時候逗我開心，在我有空的時候飛奔過來見我，謝謝你一直當我最重要的精神支柱。

感謝所有幫助過我的人，感謝上帝讓我一路順遂，謹以此論文獻給所有愛我的人。

陳元馨 于

新竹交通大學夏日午后



# Contents

AWARDS:.....	I
PUBLICATION: .....	II
摘要 .....	IV
ABSTRACT .....	V
ACKNOWLEDGEMENT: .....	VI
CONTENTS .....	VII
LIST OF FIGURES.....	IX
LIST OF TABLES .....	XI
CHAPTER 1 INTRODUCTION .....	1
1.1 MOTIVATION .....	1
1.2 BACKGROUND .....	2
1.3 SYSTEM OVERVIEW .....	3
CHAPTER 2 DYNAMIC MOVING OBJECT AND BACKGROUND SEGMENTATION.....	6
2.1 EXTRACTION OF COLOR BACKGROUND.....	6
2.1.1 <i>Creation and Initialization of Color Class</i> .....	7
2.1.2 <i>Updating Color Class</i> .....	8
2.1.3 <i>Update Converged Color Class to Background</i> .....	9
2.2 SEGMENTATION OF MOVING OBJECTS.....	10
2.3 ADAPTATION OF BACKGROUND EXTRACTION .....	10
2.3.1 <i>Adaptation in Illumination</i> .....	11
2.3.2 <i>Adaptive Thresholding for Moving Object Segmentation</i> .....	11
2.4 FAST CONNECTED-COMPONENT LABELING WITH EQUIVALENT LABEL.....	12
CHAPTER 3 RULE-BASED TRACKING REASONING.....	16
3.1 BACKGROUND COMPENSATION .....	16
3.2 PRIOR SPLIT BY LANE MASK .....	18
3.3 FILTER OUT FALSE DETECTED OBJECTS.....	21
3.4 SHADOW SUPPRESSION .....	28
3.4.1 <i>Prior Shadow Suppression in the ROI</i> .....	28
3.4.2 <i>Shadow Suppression in Each Label</i> .....	32
3.5 UPDATE TRAJECTORIES AND ELIMINATE VIBRATED MOVING OBJECTS .....	34
3.6 TRAFFIC PARAMETERS CALCULATION.....	36
3.7 PARAMETER ADAPTATION .....	39
3.8 TRACKING STABILITY IMPROVEMENT.....	40

CHAPTER 4 QUEUE RESOLUTION.....	43
4.1 QUEUE DETECTION.....	43
4.2 VEHICLE HEAD AND REAR EXTRACTION.....	43
4.2.1 <i>The Edge Property of Vehicle Head and Rear</i> .....	44
4.2.2 <i>Color Space Analysis for Gradient Computation</i> .....	46
<b>4.2.2.1 YUV Color Space</b> .....	47
<b>4.2.2.2 HSI Color Space</b> .....	48
4.3 QUEUE SPLITTING REFINEMENT.....	50
CHAPTER 5 EXPERIMENTAL RESULTS.....	55
5.1 RESULTS OF VEHICLE DETECTION.....	55
5.2 RESULTS OF TRAFFIC PARAMETER CALCULATION.....	60
5.3 PARAMETER ADAPTATION.....	64
5.4 RESULTS OF QUEUE RESOLUTION.....	67
5.5 ON-LINE DEMONSTRATION.....	68
CHAPTER 6 CONCLUSIONS AND FUTURE WORK.....	70
6.1 CONCLUSION.....	70
6.2 FUTURE WORK.....	71
REFERENCES:.....	72



# List of Figures

FIG. 1-1 BLOCK DIAGRAM OF THE PROPOSED MVDT SYSTEM .....	4
FIG. 1-2 FLOWCHART OF THE PROPOSED MVDT SYSTEM; DASHED-LINE REGIONS 1 AND 2 ARE THE SUB-FLOWCHARTS OF DYNAMIC SEGMENTATION AND RULE-BASED TRACKING REASONING RESPECTIVELY. .....	5
FIG. 2-1 THE FLOWCHART OF BACKGROUND EXTRACTION PROCESSING.....	7
FIG. 2-2 AN EXAMPLE OF FCLEL.....	15
FIG. 3-1. (A) ORIGINAL IMAGE WITH LOTS OF STOPPED VEHICLES. (B) BACKGROUND IMAGE OF (A) WITH ERROR UPDATE. (C) CORRECT BACKGROUND IMAGE OF (A). .....	18
FIG. 3-2 (A) ONE OF THE DETECTION SCENE AND THE ROI BOUNDED BY MAGENTA BOUNDING BOX. (B) THE LANE MASK OF (A). .....	19
FIG. 3-3. VEHICLES MAY BE OCCLUDED SIDE BY SIDE HORIZONTALLY ACROSS THE LANE.....	19
FIG. 3-4 (A) TWO BLUE ELLIPSES REPRESENTED TWO OCCLUDED VEHICLES BOUNDED BY RED BOUNDING BOX LOCATE ON TWO DIFFERENT LANES. (B) THE STATISTICS OF TWO ELLIPSES IN (A). .....	20
FIG. 3-5 THE OCCLUDED VEHICLES ARE SPLIT BY LANE INFORMATION SUCCESSFULLY. ....	21
FIG. 3-6 (A) THE ORIGINAL IMAGE (B) THE MOVING OBJECT IMAGE THAT MIDDLE PART OF MOVING OBJECT IS ELIMINATED.....	23
FIG. 3-7 THE BLUE REGION OF THIS SQUARE IS THE REGION WE CALCULATED FOR DENSITY JUDGMENT. ....	23
FIG. 3-8. A VISUAL EXAMPLE OF BOUNDARY MARGINS AND REJECTIONS. ....	26
FIG. 3-9. THE INCORRECT TRACKING PROCESSING OF IMAGE SEQUENCE NO.1040 TO NO.1044 IN THE FIRST LANE SHOWS THE TRACKING TRAJECTORY IS DISAPPEARED IF THE MOVING OBJECT CONNECTED TO THE BOUNDARY. .....	27
FIG. 3-10. THE COMPLETE AND CORRECT TRACKING PROCESSING OF IMAGE SEQUENCE NO.1040~NO.1044 IN THAT THE MOVING OBJECT CONNECTED TO THE BOUNDARY IS SEPARATED FROM THE BOUNDARY OBJECT. ....	28
FIG. 3-11 THE FLOWCHART OF SHADOW SUPPRESSION IN THE ROI .....	29
FIG. 3-12 (A) THE DETECTED VEHICLES ARE FRAGMENTED WITH INEXACT SHADOW JUDGMENT. (B) THE DETECTED VEHICLES ARE WITH SOME SHADOWS WHICH ARE NOT ELIMINATED CLEARLY.....	30
FIG. 3-13 THE PROCESS OF LANE-BASED RUN-LENGTH SHADOW CONFIRMING PROCESSING .....	32
FIG. 3-14 THE RESULT OF SHADOW SUPPRESSION IN THE ROI.....	32
FIG. 3-15 THE SAMPLES OF THE EFFECT OF LARGE SHADOWS.....	33
FIG. 3-16 THE RESULT OF SHADOW SUPPRESSION IN EACH LABEL. ....	34
FIG. 3-17 THE DIAGRAM OF THE RELATION BETWEEN TRACKING TRAJECTORY AND CURRENT MOVING OBJECT. ....	35
FIG. 3-18 THE DIAGRAM OF INTERSECTION OF THE CURRENT MOVING OBJECT AND THE LAST NODE OF THE EXISTED TRAJECTORY .....	36
FIG. 3-19 THE DIAGRAM OF DIRECTION CLASSIFICATION. ....	39
FIG. 3-20 THE STATISTICS OF VEHICLE WIDTH AND HEIGHT. ....	39

FIG. 3-21 (A)~(F) THE IMAGE SEQUENCE WHICH THE SILVER VEHICLE IS NOT EXTRACTED IN (D) AND (E). .....	40
FIG. 3-22 THE TRACKING TRAJECTORY IS DELETED IMMEDIATELY IF THE NEXT TRACKING NODE IS NOT FOUND IN (A) AND (B) IS THE RESULT OF RESERVING TRACKING NODE. ....	41
FIG. 3-23 TRACKING NODES FROM VEHICLE MOVING SLOWLY OR STOPPING. ....	42
FIG. 3-24 AN EXAMPLE OF SQUEEZE TRACKING NODES FOR VEHICLE STOPPING OR MOVING SLOWLY CASE. ....	42
FIG. 4-1 THE EXAMPLES AT QUEUING TIME .....	44
FIG. 4-2 THE 3X3 MASK FOR GRADIENT CALCULATION .....	45
FIG. 4-3 THE BLACK LINES IN THESE IMAGES REPRESENT EDGES .....	45
FIG. 4-4 THE RED LINES IN (B) ARE THE RESULTS OF GREEN LINES IN (A) WITH RUN-LENGTH PROCESSING. ....	46
FIG. 4-5 THE PRELIMINARY RESULTS OF STOPPED VEHICLE SPLIT. ....	46
FIG. 4-6 THE GENERAL COLORS OF VEHICLES. ....	47
FIG. 4-7 TWO GENERAL COLOR BLOCKS COMBINED REPRESENT TWO OCCLUDED VEHICLES. ....	47
FIG. 4-8 THE HSI COLOR MODEL BASED ON (A) CIRCULAR COLOR PLANES (B) HEXAGON COLOR PLANES. ....	49
FIG. 4-9 THE RESULTS OF GRADIENT CALCULATION IN G, H, S, I, Y, U, AND V COLOR PLANES. ....	50
FIG. 4-10 THE WRONG CASES OF OCCLUDED VEHICLE SPLIT. ....	50
FIG. 4-11 THE WRONG CASES OF OCCLUDED VEHICLE SPLIT. ....	51
FIG. 4-12 A GREEN LINE LOCATED ON WINDSHIELD IN (A) IS ELIMINATED IN (B). ....	51
FIG. 4-13 THE CHARACTERISTICS OF SUNROOF, THAT IS, THE POINTS OF VERTICAL GRADIENT ARE CONNECTED ON BY ONE HORIZONTALLY. ....	52
FIG. 4-14 THE FAKE GREEN SPLIT LINE LOCATED ON A VEHICLE SUNROOF IN (A) IS REMOVED IN (B). ....	52
FIG. 4-15 THE STOPPED VEHICLE BEFORE AND AFTER THE SMOOTHING PROCESSING .....	53
FIG. 4-16 THE RESULT OF FAKE SPLIT LINE ELIMINATION. ....	54
FIG. 5-1 ONE OF THE EXPERIMENTAL ENVIRONMENTS AND EXPERIMENTAL EQUIPMENTS. ....	56
FIG. 5-2 RESULTS OF VEHICLE DETECTION IN THE URBAN WITH SUNLIGHT. ....	56
FIG. 5-3 RESULTS OF VEHICLE DETECTION IN THE URBAN WITH SERIOUS SHADOWS OF VEHICLES AND ROADSIDE BUILDINGS. ....	57
FIG. 5-4 RESULTS OF VEHICLE DETECTION IN THE URBAN WITH VEHICLE HEADLIGHTS AND RAIN DROPS. ....	58
FIG. 5-5 RESULTS OF VEHICLE DETECTION IN THE EXPRESSWAY. ....	59
FIG. 5-6 RESULTS OF VEHICLE DETECTION IN THE URBAN WITH CURVE AND STRAIGHT LANES. ....	60
FIG. 5-7 RESULTS OF VEHICLE DETECTION AND TRAFFIC PARAMETER CALCULATION IN THE EXPRESSWAY. ....	61
FIG. 5-8 RESULTS OF VEHICLE DETECTION AND TRAFFIC PARAMETER CALCULATION IN THE URBAN. ....	61
FIG. 5-9 THE HISTOGRAM OF VEHICLE (A) WIDTH MEAN, AND (B) HEIGHT MEAN WITH RESPECT TO THE FRAME NUMBER. ....	66
FIG. 5-10 THE HISTOGRAM OF VEHICLE (A) WIDTH MEAN, (B) AND HEIGHT MEAN WITH RESPECTIVE TO THE VEHICLE NUMBER .....	67
FIG. 5-11 STOPPED VEHICLES ARE DETECTED AND SPLIT BY THE PROPOSED METHODS OF QUEUE DETECTION AND RESOLUTION. ....	68
FIG. 5-12 THE ACCURACY OF DETECTION OF COMPRESSED IMAGES WITH BITRATE 128KBPS TO 1152KBPS WITH FRAME RATE 30 FPS, 15 FPS, AND 10 FPS. ....	69

# List of Tables

TABLE 3-1 THE PROCEDURES OF HOW TO FILTER OUT THE FALSE DETECTED OBJECTS.....	24
TABLE 3-2 EQUATIONS USED TO CALCULATE TRAFFIC PARAMETERS. ....	37
TABLE 5-1 AVERAGE DETECTION AND TRACKING RATE OF TOTAL QUANTITIES OF VEHICLES .....	62
TABLE 5-2 AVERAGE CLASSIFICATION RATE OF SMALL OR LARGE VEHICLES.....	62
TABLE 5-3 AVERAGE SPEED DETECTION RATE OF VEHICLES .....	63
TABLE 5-4 OVERALL EVALUATION OF PROPOSED SYSTEM AT 6 DIFFERENT TEST CONDITIONS.....	63
TABLE 5-5 COMPARISON TO OTHER TECHNIQUES.....	63

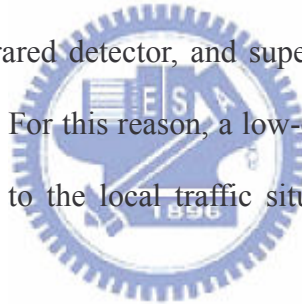


# Chapter 1 Introduction

## 1.1 Motivation

When the transportation is getting growth with years, the traffic information becomes more and more important for drivers and traffic supervisors in that the traffic is becoming more and more serious. To provide more comprehensive travel information, the vehicle detector has to be developed necessarily for the complicated traffic environment in Taiwan with lots of motorcycles and traffic jam.

There are also other kinds of vehicle detector existed to be used practically, such as loop detector, microwave detector, infrared detector, and supersonic detector etc. which are high cost and inconvenient installment. For this reason, a low-cost and simple-equipped resolution should be developed conforming to the local traffic situation for generalization of vehicle detection.



A vision-based vehicle detector is one of the most applicable devices to collect traffic information, e.g. traffic volume and speed. By automatic detection and tracking, the traffic management is more efficient that not only increases the recognition accuracy but also decreases the equipment cost. Therefore, the traffic information service more reliable and accessible any time. However, except the highway systems, the quantity of detectors in their motorways is very insufficient due to high installation cost and the poor recognition ability on motorcycles.

Also, two situations are always occurred both in the urban area and in the expressway-stopping in front of traffic lights and the traffic jam. The tracking processing is hard to apply to above situations in that the stopped moving objects would connect to the

boundary object which would be deleted. So, we propose the method to resolve and detect the queue in order that the tracking processing would be kept during the queue time.

## 1.2 Background

In vehicle detection processing, virtual slit [9] and virtual loop [1] exploits the concept of inductive loop [13] to detect vehicle passing by monitoring illumination change in pre-specified regions of a frame. As the kind of processing checks the pre-specified regions of frame only, its processing speed is fast. However, it is hard to setup, expensive, and functional limited. Another alternative uses double-difference operator [5] with gradient magnitude to detect vehicles. Although the kind of processing is more complicated than previous one, it can gather more vehicle information.

For vehicle tracking, Kato et al. [8], Kamijo et al. [9] and Tao et al. [20] employed maximum a posterior (MAP) to track occluded vehicles based on Markov random field (MRF), spatiotemporal MRF (ST-MRF) and dynamic layer shape, motion, and the appearance model, respectively. However, MAP requires much computational power. For complexity reduction, Li et al. [15] and Smith et al. [18] used sequential importance sampling (SIS), which is a class of Monte Carlo method and the Sum-of-Squared Differences (SSD) with dynamic pyramiding, to reduce the amount of inputs (samples). Even with these improvements, MAP can not quite deliver full real-time performance. The extended Kalman filter (EKF)-based techniques are faster [13], [1], [21]. These techniques estimated the positions and velocities (states) of vehicles that are represented using dynamic models. Lou et al. [7] proposed the modified EKF to reduce the sensitivity of the filter to the uncertainty of motion model. Although EKF-based techniques are robust when tracking objects with random invocations and noisy measurements, most invocations (operations of drivers) and measurements (movements of vehicles) for tracking vehicles are not very random or noisy.

Furthermore, the approach may converge to wrong states if the vehicles are occluded. Therefore, Benjamin et al. [4] utilized a feature based tracker to obtain feature trajectories and select one trajectory to represent a group of common motions. However, the motions of vehicle features will be indistinguishable if two or more occluded vehicles perform similar movement. Consequently, the rule-based reasoning method is applied to reduce the processing time and to solve the occlusion problem. For instance, Cucchiara et al. [5] applied a forward chaining production rule and urban traffic rules and Gupte et al. [6] analyzed the states of moving regions to create, delete, extend, split, or merge trajectories. However, the former merely resolves only up to two occluded vehicles. Furthermore, it requires much time to test the intersection of any previous-vehicle and current-vehicle pair. Although the latter does not limit the number of occluded vehicles, it requires a much longer processing time as the number of occluded vehicles increases.

In queue detection, the difference of the amplitude of FFT between empty road and occupied road is utilized to detect queues in [16]. Therefore, queue length is computed by edge-based vehicle detection method. Otherwise, Zanin et al. [17] proposed a queue detection system based on vehicle presence detection and movement analysis. Nevertheless, the research of queue resolution is rare in that if vehicles are not moving objects, it is hard to detect or track them.

### 1.3 System Overview

The real-time multiple-vehicle detection and tracking (MVDT) system is proposed in this thesis which can successfully identify the traffic condition by a forward looking CCD camera. The proposed MVDT system contains 3 major procedures such as dynamic segmentation, tracking processing, and queue resolution. In the dynamic segmentation, moving objects are segmented from video frames with reference to a regularly updated background and previous



trajectories. In rule-based tracking reasoning, spatiotemporal characteristics of the moving objects are utilized to update trajectories. Furthermore, the queue would be detected and split in the queue resolution. The block diagram is shown in Fig. 1-1. Consequently, the system flowchart is shown in Fig. 1-2.

The work is organized as follows. Chapter 2 and Chapter 3 introduce the methods of dynamic segmentation and rule-based tracking reasoning, respectively. Chapter 4 presents the methods of queue detection and resolution. Chapter 5 addresses the experimental results obtained by the proposed system, and Chapter 6 draws conclusions.

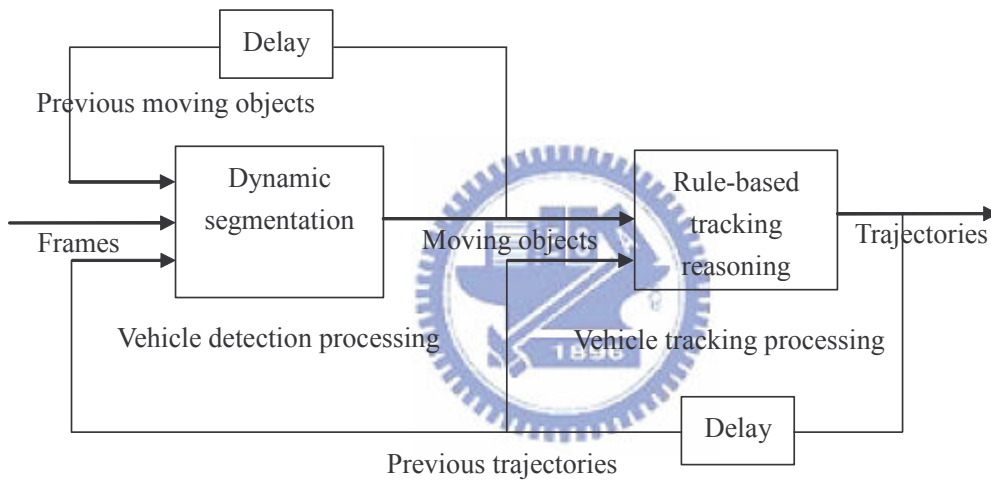


Fig. 1-1 Block diagram of the proposed MVDT system

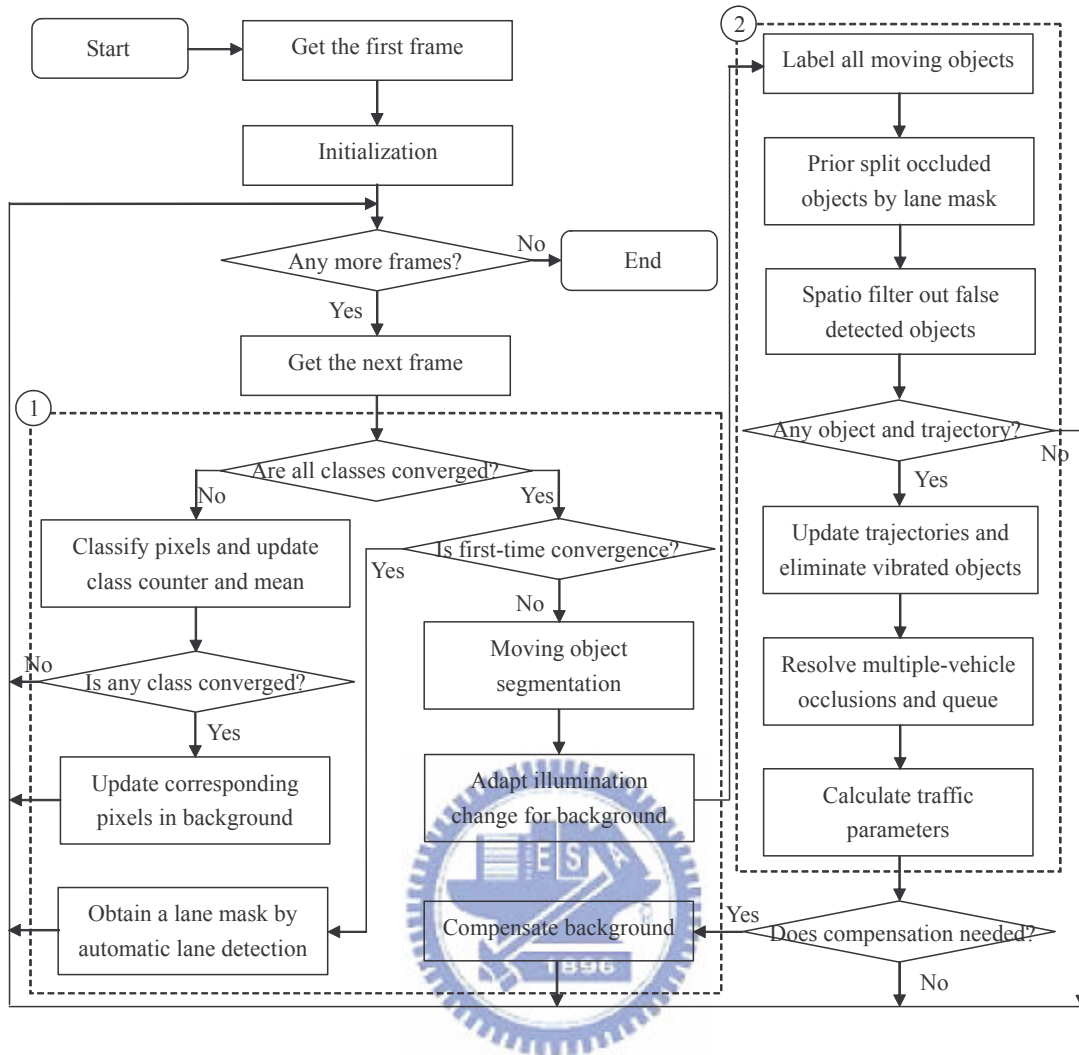


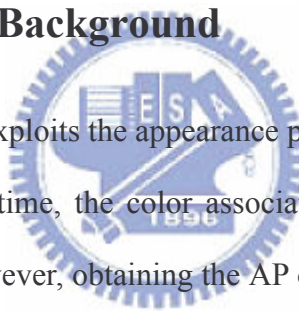
Fig. 1-2 Flowchart of the proposed MVDT system; dashed-line regions 1 and 2 are the sub-flowcharts of dynamic segmentation and rule-based tracking reasoning respectively.

# Chapter 2 Dynamic Moving Object and Background Segmentation

In order to segment desired moving objects, a color background of the image sequence should be extracted firstly. In this chapter, a pre-processing of the vehicle detection system will be introduced. Next, a color background extraction method will be described. Then, a moving object segmentation will be presented. Finally, the segmented moving object will be refined by a compensation method.

## 2.1 Extraction of Color Background

Color background extraction exploits the appearance probability (AP) of each pixel's color. That is, after a sufficiently long time, the color associated with the maximum AP is most probably a background color. However, obtaining the AP of each pixel's color requires a large memory. Hence, the AP of each pixel's color class is adopted instead. The flowchart of the color background extraction is shown in Fig. 2-1



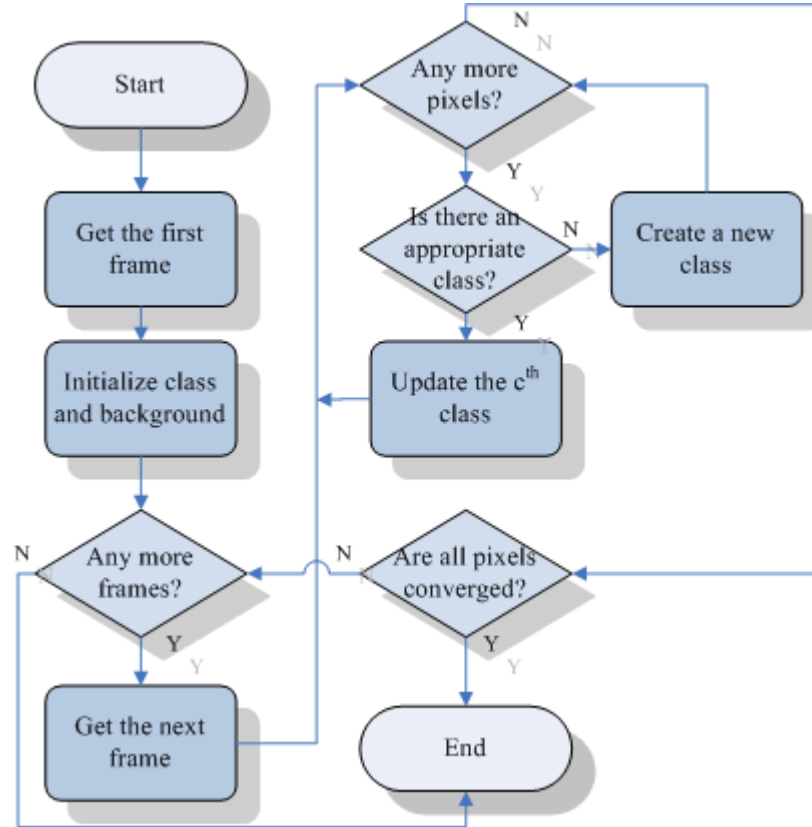


Fig. 2-1 The flowchart of background extraction processing

### 2.1.1 Creation and Initialization of Color Class

A color class located at coordinate  $(x, y)$  is uniquely identified by an ordered number. A color counter,  $CC(x, y, c)$ , and a color mean:  $CM(x, y, c)$  of the class are created to calculate AP and to classify the pixel's color in RGB (R: red, G: green, and B: blue) color space. For convenience,  $CM(x, y, c)$  is defined as a vector with three RGB color components:  $[CM_R(x, y), CM_G(x, y), CM_B(x, y)]^T$ . The total number of classes at a particular coordinate is  $NC(x, y)$ .

Initially, only one class (the 0<sup>th</sup> class) is created for each pixel. The total number of classes, color counter and the color mean of the 0<sup>th</sup> class are given as

$$NC(x, y) = 1, \quad (2-1)$$

$$CC(x, y, 0) = 1, \quad (2-2)$$

and

$$\mathbf{CM}(x, y, 0) = \mathbf{p}(x, y, t_0) \quad (2-3)$$

respectively, where  $\mathbf{p}(x, y, t_0)$  is a frame pixel located at coordinate  $(x, y)$  and sampled at the initial time  $t_0$ . Again,  $\mathbf{p}(x, y, t_0)$  is a vector with three RGB color components  $-[p_R(x, y, t_0), p_G(x, y, t_0), p_B(x, y, t_0)]^T$ .

A color background  $BG(x, y)$  which is a vector with three RGB color components  $[BG_R(x, y), BG_G(x, y), BG_B(x, y)]^T$  is required to store converged color means.

$BG_R(x, y)$ ,  $BG_G(x, y)$ , and  $BG_B(x, y)$  are initialized as

$$\begin{aligned} BG_R(x, y) &= -1 \\ BG_G(x, y) &= -1 \\ BG_B(x, y) &= -1 \end{aligned} \quad (2-4)$$

to specify that all pixels of the background have not yet converged.



### 2.1.2 Updating Color Class

The sum of absolute the differences ( $SAD$ ) between a frame pixel sampled at current time  $t$  and the corresponding  $c^{\text{th}}$  color mean is calculated as follows, to determine whether the pixel is in the  $c^{\text{th}}$  class or whether a new class must be created for it;

$$SAD(x, y, c) = \sum_{i=R,G,B} |p_i(x, y, t) - CM_i(x, y, c)| \quad (2-5)$$

, which is a decision function.

First, the decision function classifies the pixel into a class  $j$  according to

$$j = \arg \min_{0 \leq c < NC(x, y)} SAD(x, y, c). \quad (2-6)$$

After that,  $SAD(x, y, j)$  is compared with a fixed threshold  $TH_1$ . If  $SAD(x, y, j)$  is less than  $TH_1$ , then  $CM(x, y, j)$  and  $CC(x, y, j)$  are updated according to

$$CM(x, y, j) = \frac{CC(x, y, j) \times CM(x, y, j) + p(x, y, t)}{CC(x, y, j) + 1} \quad (2-7)$$

and

$$CC(x, y, j) = CC(x, y, j) + 1 \quad (2-8)$$

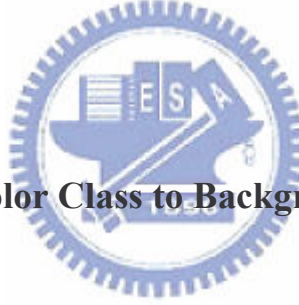
respectively. Otherwise, a new class is created by

$$CM(x, y, NC(x, y)) = p(x, y, t), \quad (2-9)$$

$$CC(x, y, NC(x, y)) = 1, \quad (2-10)$$

and

$$NC(x, y) = NC(x, y) + 1. \quad (2-11)$$



### 2.1.3 Update Converged Color Class to Background

As time passes, the color counter that belongs to the background increases rapidly. Accordingly, the  $c^{\text{th}}$  color counter updated at time  $t$  is utilized to derive the AP of the class defined as

$$AP(x, y, c) = \frac{CC(x, y, c)}{\sum_{i=0}^{NC(x, y)-1} CC(x, y, i)} = \frac{CC(x, y, c)}{t+1} \quad (2-12)$$

The AP of  $k^{\text{th}}$  class that is most probably classified as the background is given by

$$k = \arg \max_{0 \leq c < NC(x, y)} AP(x, y, c). \quad (2-13)$$

Hence, by applying a dynamic threshold  $TH_2$ , the background can converge according to

$$\mathbf{BG}(x, y) = [\mathbf{CM}(x, y, k)]$$

if

$$BG_r(x, y) = -1 \text{ and } AP(x, y, k) \geq TH_2 \quad (2-14)$$

where  $[\ ]$  is a rounding operation for each tuple in a vector.

## 2.2 Segmentation of Moving Objects

Once the background has been extracted, the moving objects can be deleted by checking the *SAD* between the background and the input frame

$$SAD(x, y) = \sum_{i=R,G,B} |(p_i(x, y, t) - BG_i(x, y))| \quad (2-15)$$

Given two dynamic thresholds of extracting moving object,  $MTH_L$  and  $MTH_H$ , a binary mask on the moving objects is obtained according to

$$MM(x, y) = \begin{cases} 1 & , \text{ } MSD(x, y) < MTH_L \text{ and } MSD(x, y) > MTH_H \\ 0 & , \text{ otherwise} \end{cases} \quad (2-16)$$

The following section describes the method proposed to calculate  $MTH_L$  and  $MTH_H$ .

## 2.3 Adaptation of Background Extraction

The changeable environment will highly affect the background extraction, such as illumination. On the other hand, the illumination varying also influences on the segmentation of moving object. Consequently, the illumination adaptation methods are presented in the following.

### 2.3.1 Adaptation in Illumination

A background mask  $BM(x, y)$  defined as the complement of  $MM(x, y)$  is used to select the regions of a background that must be updated to adapt to a change in illumination. Given a predefined  $n$ , the background is updated by

$$\mathbf{BG}(x, y) = \frac{n-1}{n} \times \mathbf{BG}(x, y) + \frac{1}{n} \times p(x, y, t), \quad (2-17)$$

when  $BM(x, y) = 1$ . If  $n$  is large, then the background will not easily adapt to a slow change in illumination change. However, if  $n$  is small, then the background will be easily affected by moving objects and noise. In our experience,  $n = 8$  is a good compromise.

### 2.3.2 Adaptive Thresholding for Moving Object Segmentation

As the background is updated at each frame, the dynamic segmentation can overcome the slow change in illumination, such as that associated with daylight or weather, with a fixed threshold. However, for rapid illumination change, a fixed threshold will cause false detection. Therefore, an adaptive thresholding procedure is developed to find the low-valley  $\mathbf{VL}=[VL_R, VL_G, VL_B]^T$  and high-valley  $\mathbf{VH}=[VH_R, VH_G, VH_B]^T$  of the filtered difference distribution  $\mathbf{FD}(n)=[FD_R(n), FD_G(n), FD_B(n)]^T$  between the background and the input frame.

The filtered difference distribution that depends on a difference distribution

$$\mathbf{D}(n) = \left[ \sum_{p_R(x,y,t)-BG_R(x,y)=n} 1, \sum_{p_G(x,y,t)-BG_G(x,y)=n} 1, \sum_{p_B(x,y,t)-BG_B(x,y)=n} 1 \right]^T \quad (2-18)$$

is obtained by

$$\mathbf{FD}(n) = \frac{\sum_{i=n-p}^{n+p} \mathbf{D}(i)}{2p+1} \quad (2-19)$$



where  $(2p+1)$  is the filter order of the specified moving average filter. The reason for not using  $\mathbf{D}(n)$  to find the valleys directly is due to that  $\mathbf{D}(n)$  is noisy. With the filtered difference distribution, the Laplacian operator

$$\nabla^2 \mathbf{FD}(n) = \mathbf{FD}(n+1) - 2\mathbf{FD}(n) + \mathbf{FD}(n-1) \quad (2-20)$$

is utilized to find the correct valleys

$$\mathbf{VL} = \min \left( \arg \left( \nabla^2 \mathbf{FD}(n) = 0 \right) \right), \quad (2-21)$$

and

$$\mathbf{VH} = \max \left( \arg \left( \nabla^2 \mathbf{FD}(n) = 0 \right) \right) \quad (2-22)$$

where  $\min()$ ,  $\max()$ , and  $\arg()$  are tuple-wised operation.

The finding of valleys of  $\mathbf{FD}(n)$  is based on the observation that the most frequently appearing differences in  $\mathbf{D}(n)$  are associated the background. From the valleys, the dynamic thresholds  $MTH_L$  and  $MTH_H$  can be obtained by

$$\begin{aligned} MTH_L &= VL_R + VL_G + VL_B \\ MTH_H &= VH_R + VH_G + VH_B \end{aligned} \quad (2-23)$$

## 2.4 Fast Connected-Component Labeling with Equivalent Label

The moving object mask  $MM(x, y)$  is generated after processing of moving object segmentation. Each 8-connected component in  $MM(x, y)$  is thought as a moving object. The moving object may be a distinct vehicle, two or more vehicles overlapped as single object, a part of a distinct vehicle (mis-detected objects), or noises (false detected objects). Each moving object is given a label by proposed fast connected-component labeling with equivalent label (FCLEL) algorithm.

The FCLEL is a sequential connected-component labeling algorithm. In comparison with the parallel connected-component labeling algorithm, the sequential one is more suitable for the proposed MVDT system. The reason is due to that the proposed MVDT system is a PC-based system which can only execute operations and access data sequentially. Similar to [10], the difference is that FCLEL records equivalent labels for each partition in the first phase, and uses the equivalent labels to speed up the partitions combination in the second phase. The detail FCLEL algorithm is described in the following.

The modified connected component labeling method involves two phases. In the first phase, a frame pixel  $p(x, y)$  located at coordinate  $(x, y)$  is assumed to represent the image of size  $W \times H$  segmented after pre-processing, where  $g(x, y) \in \mathbf{Z}$  is a label image of  $p(x, y)$ , and  $L \in \mathbf{N}$  is a label counter, which stores the current minimum illegal label value obtained by the raster scanning from top to bottom and then from left to right;  $E_q(l) \in \mathbf{Z}$  represents the equivalence linkage of a label  $l < L$ ;  $(x_0, y_0)$  is the coordinates of the pixel encountered during the raster, and  $N(x_0, y_0)$  denotes the coordinates of the four previously scanned pixels, based on the coordinates of the central pixel  $(x_0, y_0)$ :

$$N(x_0, y_0) = \{(x_0 - 1, y_0 - 1), (x_0 - 1, y_0), (x_0, y_0 - 1), (x_0 + 1, y_0 - 1)\}; \quad (2-24)$$

Initially,  $L$  is set to one to indicate that no label has yet been created. During raster scanning, whenever the encountered pixel is in the background ( $MM(x_0, y_0) = 255$ ), the corresponding coordinates of the label of the image are set to 0 ( $g(x_0, y_0) = 0$ ). However, if the encountered pixel is in the foreground and disjointed to the previous foreground pixels, a new component is generated by assigning the label counter to the coordinates of the label image ( $g(x_0, y_0) = L$ ) and then increasing the label counter by one ( $L = L + 1$ ). The equivalent linkage of the label thus generated is initialized to 0 ( $E_q(g(x_0, y_0)) = 0$ ; where 0 denotes unlabeled). Otherwise, if the encountered pixel is in the foreground and is joined to previously scanned foreground pixels, then the coordinates of the corresponding label image are set to the

minimum equivalent linkage roots of the image's neighbors  $g(x_0, y_0) = \min_{(x,y) \in N(x_0, y_0)} E_Q(g(x, y))$ ;

$E_Q(l)$  finds the root of the equivalent linkage for a specified label  $l$ .

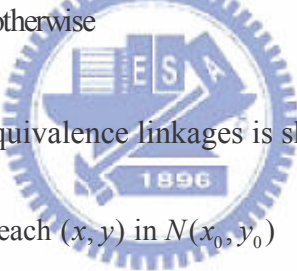
$$g(x_0, y_0) = \begin{cases} 0 & \text{if } MM(x_0, y_0) = 255, \\ L, (L=L+1) & \text{else if } g(x, y) = 0 \quad \forall \{(x, y) \in N(x_0, y_0)\}, \\ \min_{(x,y) \in N(x_0, y_0)} E_Q(g(x, y)) & \text{otherwise.} \end{cases} \quad (2-25)$$

$$E_Q(l) = \begin{cases} E_Q(E_q(l)) & \text{if } E_q(l) \neq 0, \\ l & \text{otherwise.} \end{cases} \quad (2-26)$$

Additionally, the equivalent linkages of the neighbors of encountered pixel in the label image must be updated by:

$$\begin{cases} \text{no operation} & \text{if } MM(x_0, y_0) = 255, \\ E_q(g(x_0, y_0)) = 0 & \text{else if } g(x, y) = 0 \quad \forall \{(x, y) \in N(x_0, y_0)\}, \\ \text{update equivalent linkages} & \text{otherwise} \end{cases} \quad (2-27)$$

where the pseudo code of update equivalence linkages is shown below:



```

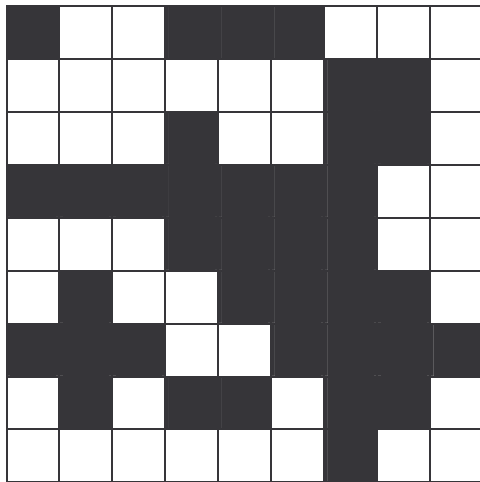
for each (x, y) in N(x_0, y_0)
{
    E = E_q(g(x, y));
    if (E_Q(E) ≠ g(x_0, y_0))
    {
        do
        {
            E' = E_q(E);
            E_q(E) = g(x_0, y_0);
            E = E';
        } while (E is unequal to 0);
    }
}

```

The first phase resolves all equivalent linkages but not the label image. Consequently, in the second phase, labels must be reassigned to resolve the label values stored in the label image. The label reassignment for each pixel with coordinates  $(x_0, y_0)$  is described:

$$g(x_0, y_0) = \begin{cases} \text{no operation} & \text{if } g(x_0, y_0) = 0 \text{ or } E_q(g(x_0, y_0)) = 0, \\ E_Q(g(x_0, y_0)) & \text{otherwise.} \end{cases} \quad (2-28)$$

After the second phase, all connected components are labeled.



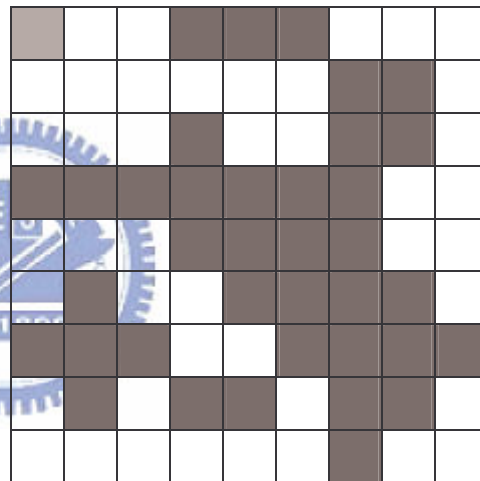
(a)  $MM(x, y)$

1	0	0	2	2	2	0	0	0
0	0	0	0	0	0	2	2	0
0	0	0	3	0	0	2	2	0
4	4	3	3	3	2	2	0	0
0	0	0	3	2	2	2	0	0
0	5	0	0	2	2	2	2	0
5	5	5	0	0	2	2	2	2
0	5	0	5	2	0	2	2	0
0	0	0	0	0	0	2	0	0

(b)  $g(x, y)$  after the first phase

1	0	0	2	2	2	0	0	0
0	0	0	0	0	0	2	2	0
0	0	0	2	0	0	2	2	0
2	2	2	2	2	2	2	0	0
0	0	0	2	2	2	2	0	0
0	2	0	0	2	2	2	2	0
2	2	2	0	0	2	2	2	2
0	2	0	2	2	0	2	2	0
0	0	0	0	0	0	2	0	0

(c)  $g(x, y)$  after the second phase



(d) The colorful representation of connected components (component 1: light gray and component 2: dark gray)

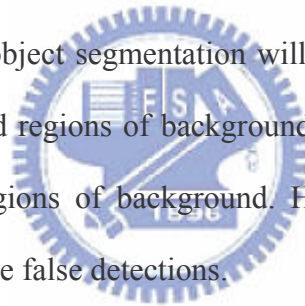
Fig. 2-2 An example of FCLEL.

# Chapter 3 Rule-based Tracking

## Reasoning

### 3.1 Background Compensation

Although the background extraction can obtain the initial background rapidly and robustly, it maybe contains some vehicles as the initial background. That is because some vehicles park on the roadside or stop in queue during the background extraction. If the stopped vehicles start to move away, the moving object segmentation will false detect the regions as moving objects. Further, the false detected regions of background will never be updated because we do not update moving object regions of background. Hence, a background compensation technique is proposed to correct the false detections.



The background compensation technique uses the trajectories feedback from vehicle tracking processing to decide whether the moving objects are false detected or not. If the moving objects are false detected, the following three situations will occur:

1. The centers of moving objects does not change too much for a period of time;
2. Before update, the edge property of moving object would be checked. If it contains obvious edge property, the vehicle-like tracking node can't be updated in the background.
3. The starting nodes of trajectories are not near the boundary of a frame;

If any trajectory of moving objects satisfies the three situations, the regions of moving objects will be set as background and re-initial the region.

Otherwise, some stopped vehicles are considered as false detected objects in tracking processing. So, the above method is not effect to compensate the background. The difference between the current image and the preceding background ( $D_{UB}$ ) is checked:

$$D_{UB} = \sum_{x,y} p(x, y, t) - \mathbf{BG}(x, y) \quad (3-1)$$

If the difference is small, the background would be updated; if the difference is large enough, the background would not be renewed. This would protect background update from interference of moving objects. Fig. 3-1(b) is the background update without above processing with Fig. 3-1(a) that Fig. 3-1(c) demonstrates this background compensation is effective.

A background mask  $BM(x, y)$  defined as the complement of  $MM(x, y)$  is used to select the regions of a background that must be updated to adapt to a change in illumination. Given a predefined  $n$ , the background is updated by

$$\mathbf{BG}(x, y) = \frac{n-1}{n} \times \mathbf{BG}(x, y) + \frac{1}{n} \times p(x, y, t) \quad (3-2)$$

when  $BM(x, y) = 1$ . If  $n$  is large, then the background will not easily adapt to a slow change in illumination change. However, if  $n$  is small, then the background will be easily affected by moving objects and noise. In our experience,  $n = 8$  is a good compromise.



(a)

(b)



(c)

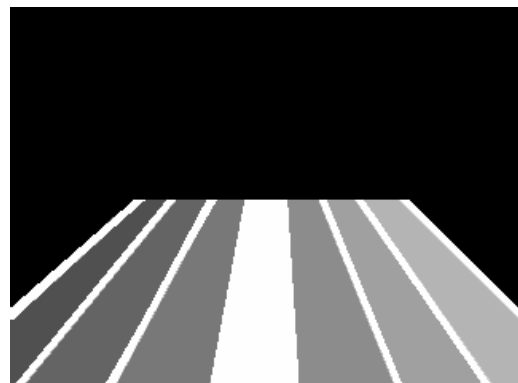
Fig. 3-1. (a) Original image with lots of stopped vehicles. (b) Background image of (a) with error update. (c) Correct background image of (a).

### 3.2 Prior Split by Lane Mask

If vehicles are occluded when they just entrance to the frame, the tracking processing will have trouble to create correct trajectories. Even the post splitting technique could not split the kind of occlusion. Therefore, a splitting technique is required to resolve the occlusion prior to tracking processing. In the study, a lane mask as shown in Fig. 3-2 is used as a reference of separations of occluded vehicles. The concept is-most vehicles are occluded side by side horizontally across adjacent lanes as shown in Fig. 3-3 when they are moving vertically. In the following, the proposed prior occlusion detection and resolution based on the concept are described.



(a)



(b)

Fig. 3-2 (a) One of the detection scene and the ROI bounded by magenta bounding box. (b)

The lane mask of (a).

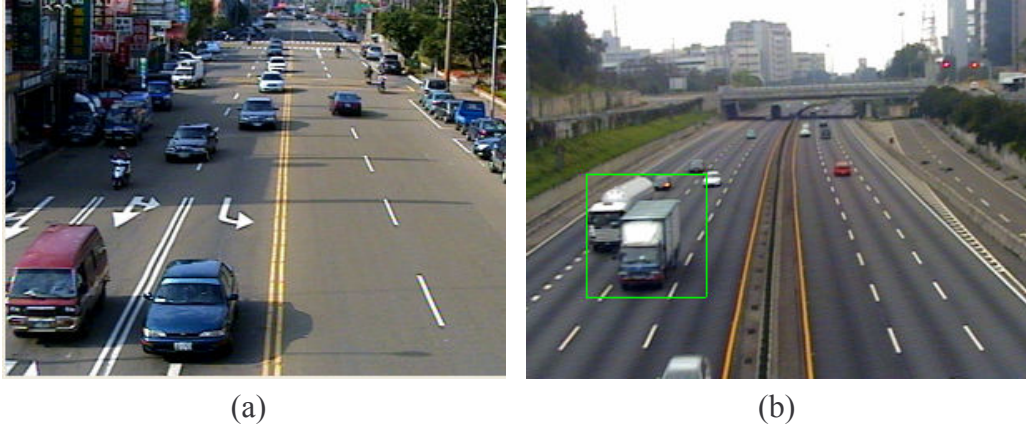


Fig. 3-3. Vehicles may be occluded side by side horizontally across the lane.

Before occlusion detection, the label ID  $l$  and the label ID image  $g(x, y)$  of a moving object obtained after connected-component labeling and the lane mask  $LM(x, y)$  are used. Also, the spatial properties used top-most coordinate  $T(l)$ , left-most coordinate  $L(l)$ , bottom-most coordinate  $B(l)$ , right-most coordinate  $R(l)$ , width  $W(l)$ , and height  $H(l)$ . Recalling that, the values in lane mask are: -1 (ignored), 0 (separators or boundaries), 1 (first lane), 2 (second lane), and so on.

For occlusion detection, first of all, we judge whether the bounding box of the label contains occluded vehicles shown in Fig. 3-4(a) by:

$$MM(x, y) = 0$$

$$\text{when } H(l)/W(l) > TH_{HW}, g(x, y) = l \text{ and } LM(x, y) = 0 \quad (3-3)$$

before splitting, where  $TH_{HW}$  is the threshold of the height-width ratio.



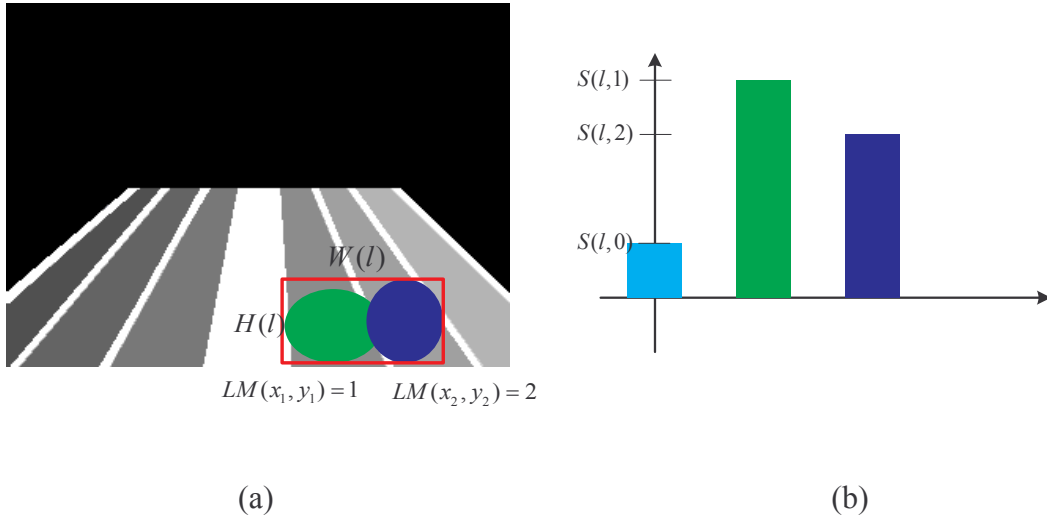


Fig. 3-4 (a) Two blue ellipses represented two occluded vehicles bounded by red bounding box locate on two different lanes. (b) The statistics of two ellipses in (a).

For occlusion detection, each pixel of the  $l^{\text{th}}$  moving object is checked to determine whether the pixel should amount to a histogram or not by

$$S(l, LM(x, y)) = S(l, LM(x, y)) + 1$$

when  $g(x, y) = l$  and  $LM(x, y) \neq -1$  and  $LM(x, y) \neq 0$  (3-4)

satisfied.

Certainly, each element in the histograms should be assigned to 0 in advance. With the histograms, the occurrence of occlusion can be detected by

$$|S(l, h) - S(l, h+1)| < TH5 \quad (3-5)$$

where, in authors' experience, a reasonable value of threshold  $TH5$  is 5.

For occlusion resolution, the  $l^{\text{th}}$  moving object is split by

$$MM(x, y) = 0$$

if  $g(x, y) = l$  and  $LM(x, y) = 0$ . (3-6)

The occluded vehicles shown in Fig. 3-5(a) and (c) are split by lane information successfully which are shown in Fig. 3-5 (b) and (d).

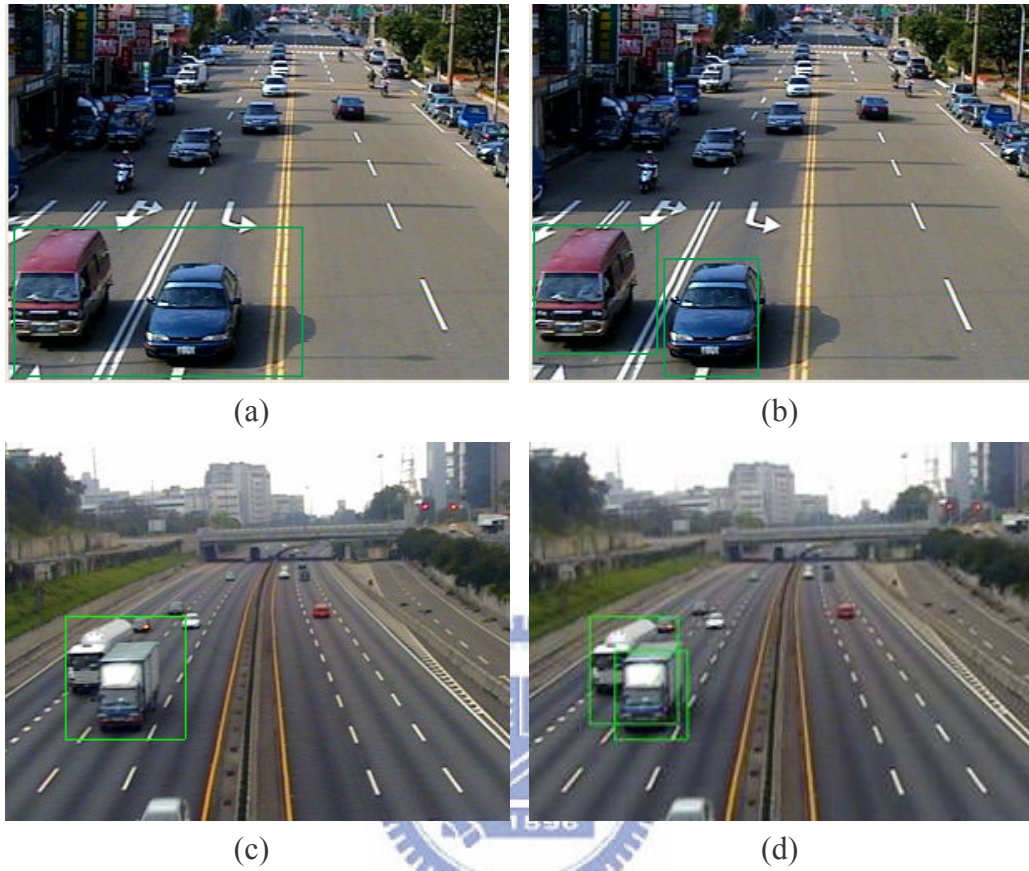


Fig. 3-5 The occluded vehicles are split by lane information successfully.

### 3.3 Filter Out False Detected Objects

In general, false detected objects can be eliminated by the spatial properties obtained after connected component labeling ([19]). The spatial properties used top-most coordinate  $T(l)$ , left-most coordinate  $L(l)$ , bottom-most coordinate  $B(l)$ , right-most coordinate  $R(l)$ , total pixel  $p(l)$ , area  $A(l)$ , width  $W(l)$ , height  $H(l)$ , aspect ratio  $AR(l)$ , size  $S(l)$ , and density  $D(l)$ , where  $l$  is label ID. Among the spatial properties, the first 5 properties can be obtained during connected-component labeling. The others are derived from the 5 properties as shown below:

$$W(l) = R(l) - L(l) + 1 \quad (3-7)$$

$$H(l) = B(l) - T(l) + 1 \quad (3-8)$$

$$AR(l) = \frac{H(l)}{W(l)} \quad (3-9)$$

$$A(l) = W(l) \times H(l) \quad (3-10)$$

$$S(l) = \sum_{L \in l} p(L) \quad (3-11)$$

$$D(l) = \frac{S(l)}{A(l)} \quad (3-12)$$

Moreover, some moving objects have general width-height ratio but small density considered as a noise in that the bounding box of vehicles and motorcycles have large density. However, some regions of moving objects would be eliminated after shadow suppression or background extraction as shown in Fig. 3-6, because of the similarity of color between vehicle windows and shadows or the ground. Therefore, the shape of the moving object is the most important consideration. So, the bounding box of the moving object is divided into 9 parts as shown in Fig. 3-7. We only count up the number of pixels of the blue region in Fig. 3-7, that is, the edge of the moving object. The new density definition of the moving object is:

$$D(l) = \frac{S_e(l)}{\frac{8}{9} \times A(l)} \quad (3-13)$$

where  $S_e(l)$  is the amount of the pixel of moving object edge.



Fig. 3-6(a) The original image (b) The moving object image that middle part of moving object is eliminated.

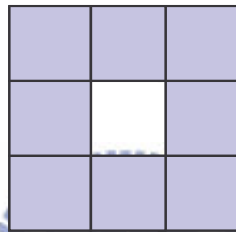


Fig. 3-7 The blue region of this square is the region we calculated for density judgment.

The spatial properties are used to filter out the false detected objects by using a thresholding method. The thresholding operators are shown in Table 3-1.

Table 3-1 The procedures of how to filter out the false detected objects

Properties	Procedure
$W(l) \in \mathbf{N}$	$\begin{cases} \text{Eliminate the label } l & \text{if }  W(l) - WM(l,t)  \leq 2 \times \sqrt{WV(l,t)}, \\ \text{Do nothing} & \text{otherwise.} \end{cases}$ <p>where the <math>WM(l,t)</math> and <math>WV(l,t)</math> are mean and variance of the width of all moving objects found until now.</p>
$H(l) \in \mathbf{N}$	$\begin{cases} \text{Eliminate the label } l & \text{if }  H(l) - HM(l,t)  \leq 2 \times \sqrt{HV(l,t)}, \\ \text{Do nothing} & \text{otherwise.} \end{cases}$ <p>where the <math>HM(l,t)</math> and <math>HV(l,t)</math> are mean and variance of the height of all moving objects found until now.</p>
$AR(l) \in \mathbf{N}$	$\begin{cases} \text{Eliminate the label } l & \text{if } \left  \frac{AR(l) - HM(l,t)}{WM(l,t)} \right  \leq 2 \times \sqrt{\frac{HV(l,t)}{WV(l,t)}}, \\ \text{Do nothing} & \text{otherwise.} \end{cases}$
$A(l) \in \mathbf{N}$	$\begin{cases} \text{Eliminate the label } l & \text{if }  A(l) - HM(l,t) \times WM(l,t)  \leq 2 \times \sqrt{HM(l,t) \times WM(l,t)}, \\ \text{Do nothing} & \text{otherwise.} \end{cases}$
$D(l) \in \mathbf{N}$	$\begin{cases} \text{Eliminate the label } l & \text{if } D(l) \leq TH_d, \\ \text{Do nothing} & \text{otherwise.} \end{cases}$ <p>where <math>TH_d</math> is the threshold of density.</p>

For spatiotemporal noises, the spatiotemporal properties that contain inter-relationships between a frame and its previous frames are used to eliminate them.

The first case of spatiotemporal noise is the roadside noises. To delete the roadside noises due to the original stopped vehicles moving away, the three conditions considered in the background compensation are used.

The second case of spatiotemporal noise is called vibrated moving objects, such as the leaves which are vibrated by the wind. Although these moving objects can not be classified to

false detected objects, they are undesired in the vehicle tracking processing. For such a reason, the kind of spatiotemporal noises is stated in the sub-section H.

The last case of spatiotemporal noise is called uncompleted objects when parts of the moving objects are out of the frame. Such kind of moving objects might cause wrong judgment in the tracking processing. In Eq.(3-13), the inequalities formed by  $MT \in \mathbf{N}$ ,  $ML \in \mathbf{N}$ ,  $MB \in \mathbf{N}$ , and  $MR \in \mathbf{N}$  are used to eliminate the uncompleted objects.

Eliminate the boundary object with label  $l$ ,

$$\begin{aligned} \text{if } \{ [T(l) < MT \text{ and } B(l) < RT] \text{ or } [L(l) < ML \text{ and } R(l) < RL] \\ \text{or } [B(l) > MB \text{ and } T(l) > RB] \text{ or } [R(l) > MR \text{ and } L(l) > RR] \} \end{aligned} \quad (3-14)$$

However, for large moving objects, such as trucks and buses, parts of the objects are mostly out of the frame. Therefore, an additional criterion is used to reject large moving objects from being eliminated. In Eq.(32), the inequalities formed by  $RT \in \mathbf{N}$ ,  $RL \in \mathbf{N}$ ,  $RB \in \mathbf{N}$ , and  $RR \in \mathbf{N}$  are used to reject large moving objects from being eliminated even when they are close to the boundary. A visual example of the boundary margins and boundary rejections are shown in Fig. 3-8. The red region is formed by boundary margins and green region is formed by boundary rejections. In Fig. 3-8(a), the  $B(l)$  of the truck and  $B(l)$  of the car are close to the boundary, and both vehicles will be eliminated before tracking processing. In Fig. 3-8(b), the car will not be eliminated because it is not close to the boundary. Although the  $B(l)$  of the bus is still close to the boundary, the bus will not be eliminated because the  $B(l)$  of the bus is greater than  $RT$ .

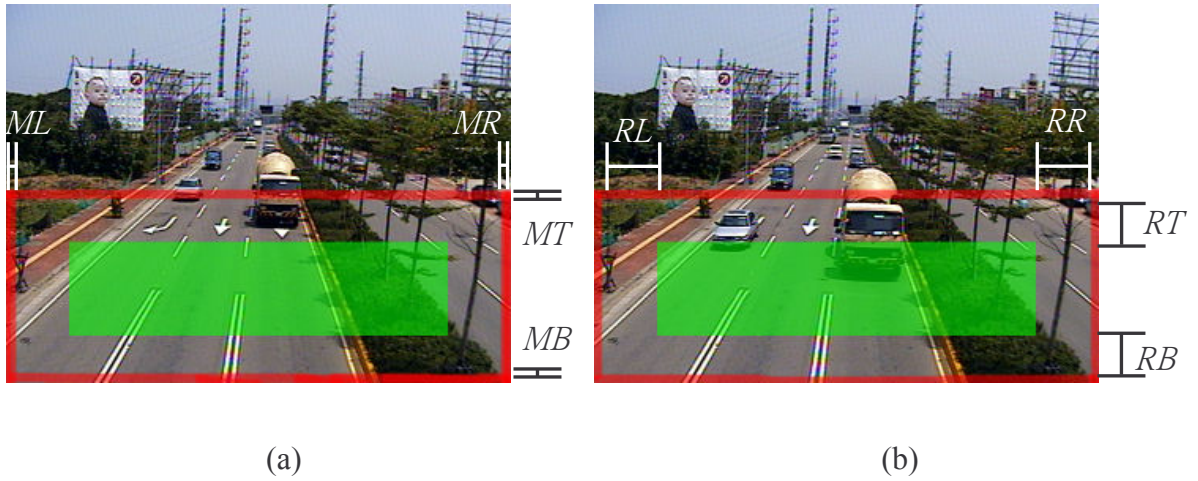


Fig. 3-8. A visual example of boundary margins and rejections.

According to the filtering out the uncompleted objects method, the vehicle near the boundary would be deleted. But at the queue time, more than one vehicles occluded near the boundary would be eliminated that the tracking process of following vehicles would be interrupted as shown in Fig. 3-9. This sequence shows that when a rear vehicle approaches to the fore stopped preceding vehicle near the boundary, the trajectory of the rear vehicle (magenta line) would be deleted because of the lack of the next tracking nodes.

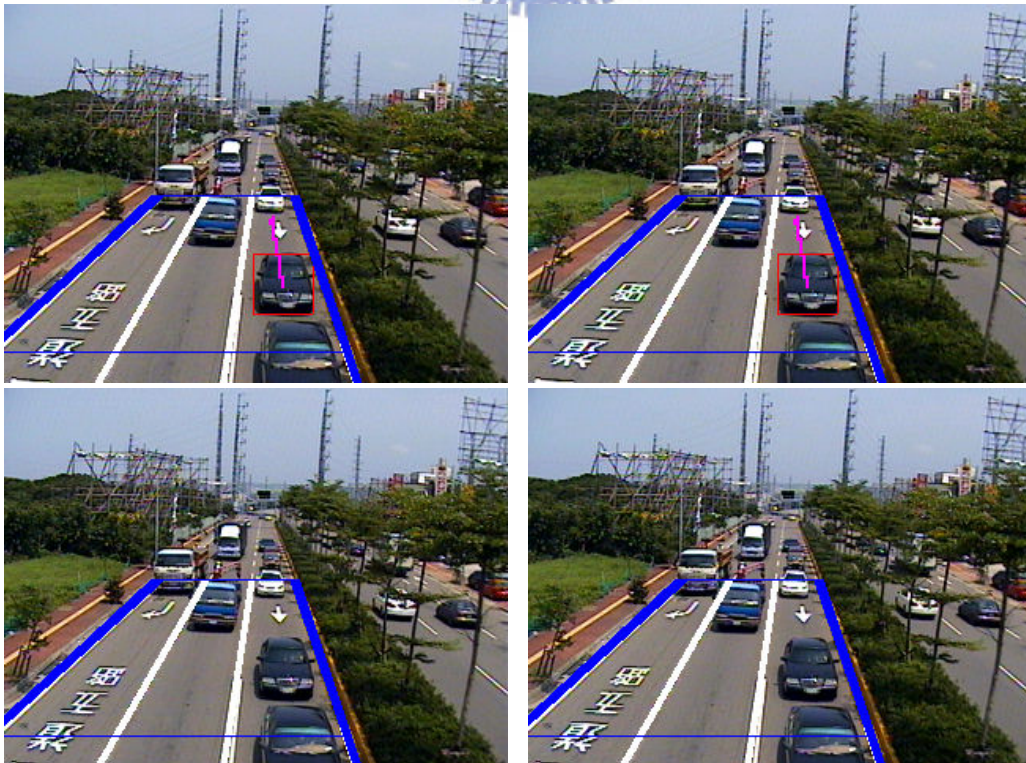




Fig. 3-9. The incorrect tracking processing of image sequence no.1040 to no.1044 in the first lane shows the tracking trajectory is disappeared if the moving object connected to the boundary.

Therefore, if the bounding boxes of moving objects are connected to the boundary of the detection zone and the speeds of moving objects become slower, the boundary occlusion resolution is utilized. If the intersection of a moving object and a last trajectory node is large enough, the moving object would be split based on the last trajectory node. Consequently, the queue of stopped vehicle would not be deleted and the tracking trajectory is kept as shown in

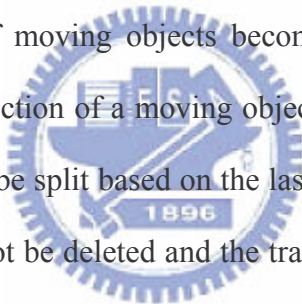
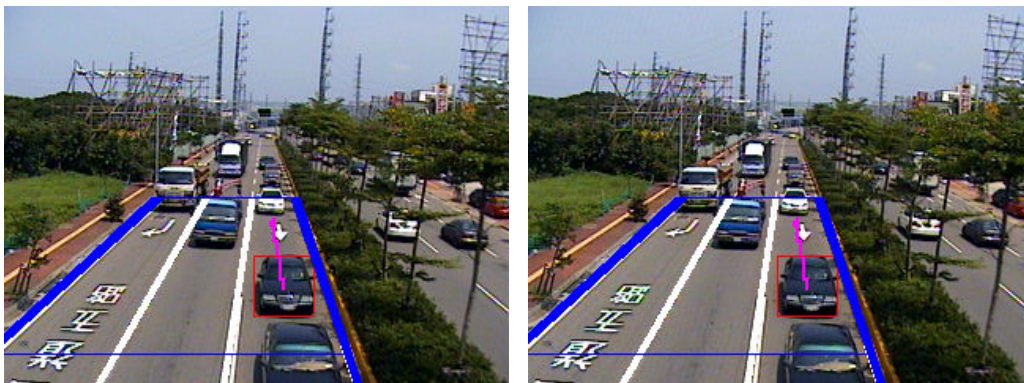


Fig. 3-10





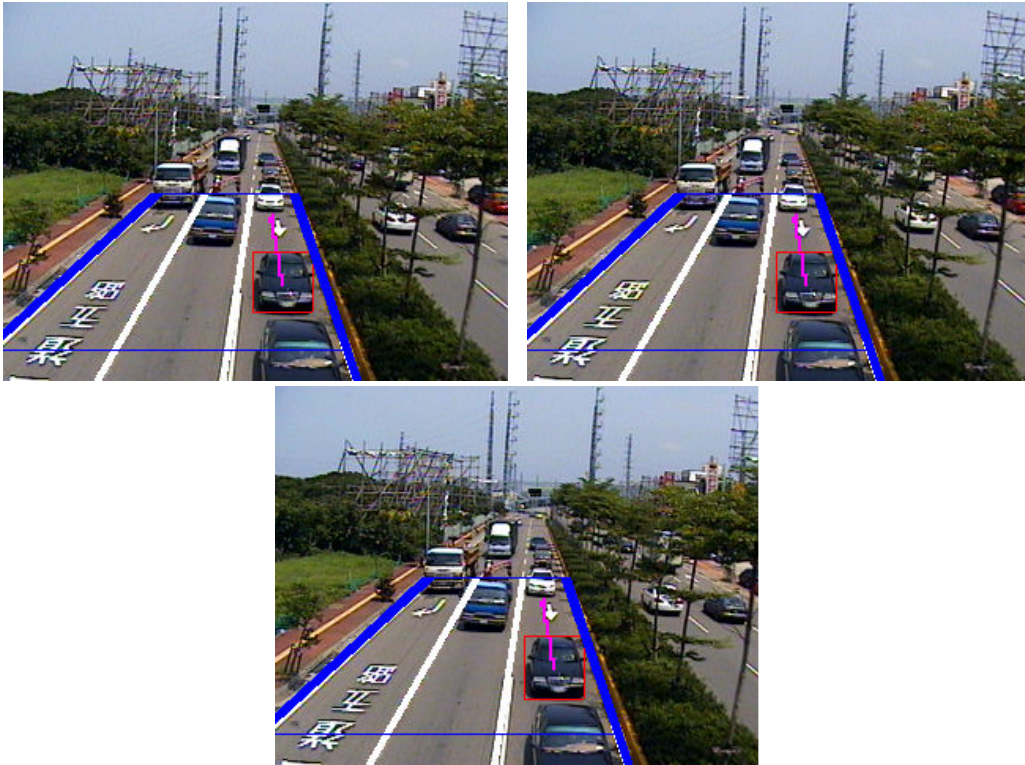
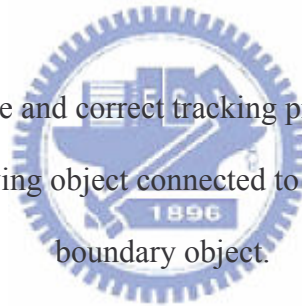


Fig. 3-10. The complete and correct tracking processing of image sequence no.1040~no.1044 in that the moving object connected to the boundary is separated from the boundary object.



### 3.4 Shadow Suppression

Shadows are always the serious problem of tracking processing in that the occlusion and error detection are occurred frequently. Consequently, the methods of shadow suppression are introduced in this subsection.

#### 3.4.1 Prior Shadow Suppression in the ROI

After background extraction, the information of a moving object mask will be refined by shadow suppression. The purpose of refining a moving object mask is to resolve some occluded vehicles caused by shadows. The flowchart of processing of shadow suppression is

shown in Fig. 3-11.

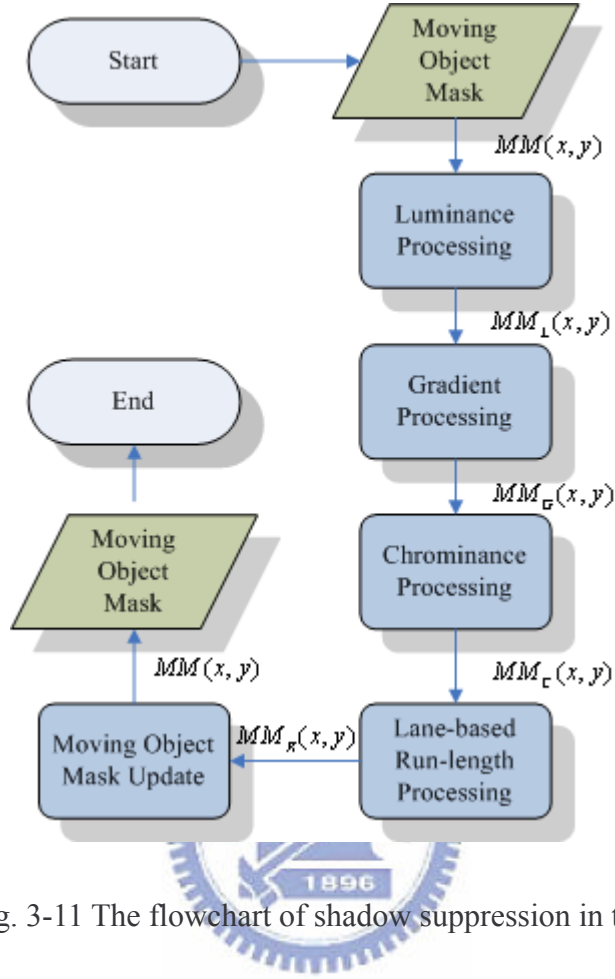


Fig. 3-11 The flowchart of shadow suppression in the ROI

In Fig. 3-11, the luminance processing reserves low luminance regions based on the moving object mask  $MM(x, y)$  by

$$MM_L(x, y) = \begin{cases} 1 & , \text{ If } (MM(x, y) = 1 \text{ and } I(x, y) < TH_L) \\ 0 & , \text{ otherwise} \end{cases} \quad (3-15)$$

where the  $I(x, y)$  is the transformation from color to luminance.

Next, the gradient processing eliminates edges of the remained regions  $MM_L(x, y)$  by

$$MM_G(x, y) = \begin{cases} 1 & , \text{ If } (MM_L(x, y) = 1 \text{ and } G(x, y) < TH_G) \\ 0 & , \text{ otherwise} \end{cases} \quad (3-16)$$

where the  $G(x, y)$  is the gradient value after the Sobel operator.

Then, the chrominance processing further pertains to low chrominance regions of the remained regions  $MM_G(x,y)$  by

$$MM_C(x,y) = \begin{cases} 1 & , \text{ If } (MM_G(x,y) = 1 \text{ and } C(x,y) < TH_C) \\ 0 & , \text{ otherwise} \end{cases} \quad (3-17)$$

where the  $C(x,y)$  is the sum of chrominance value,  $C_b + C_r$ .

The corresponding values  $TH_L$ ,  $TH_G$ , and  $TH_C$  are fixed thresholds of maximum allowable luminance, gradient, and chrominance, respectively.

$MM_C$  is considered as a shadow-candidate mask. Nevertheless, it is difficult to set the proper  $TH_L$ ,  $TH_G$ , and  $TH_C$  for all cases to eliminate the shadow. Because there are some dark parts of vehicles with similar above properties to shadows, some detected vehicles are fragmented in Fig. 3-12(a) or some shadows are not suppressed in different threshold in Fig. 3-12(b). As shown in Fig. 3-13(a), most vehicles are moving along the lane, so the lane-based run-length method is proposed to suppress shadows.



Fig. 3-12(a)The detected vehicles are fragmented with inexact shadow judgment. (b)The detected vehicles are with some shadows which are not eliminated clearly.

First of all, the label of a moving object is classified to a lane ID. The method used to classify a label to a lane ID is specified as

$$h^*(l) = \arg \max_h S(l, h) \quad (3-18)$$

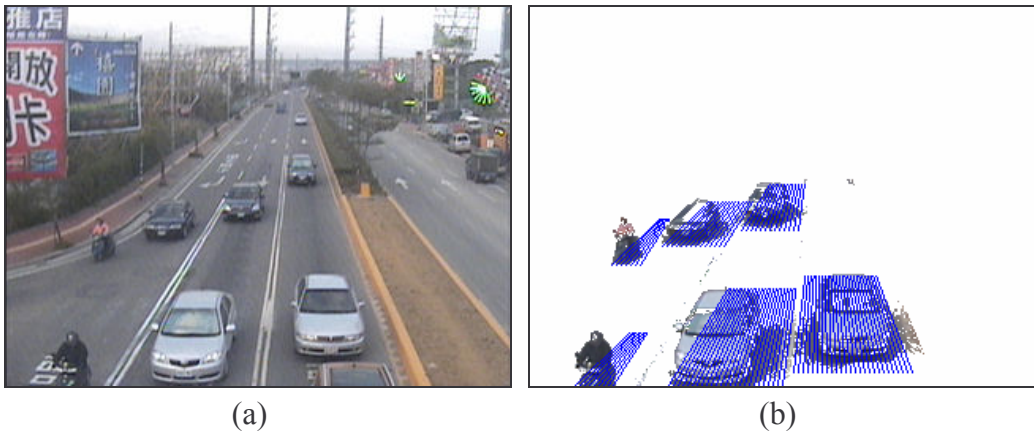
where the way to obtain the histogram  $S(l, h)$  is the same as (3-17). Then, prior splitting by lane information is utilized to avoid interference with occlusion before deciding the search region depending on the bounding box of a label. Scan over the lane along the lane direction represented by blue lines in Fig. 3-13(b) and (c) in which the magenta parts represent labels. If the amount of continuous points of the scan line in shadow-candidate mask is large enough, the line segment of this scanline is considered as a shadow and shadow mask  $R(x, y)$  will be set as 1. The lane-base run-length processing filters the shadow-like mask  $MM_R$ .

$$MM_R(x, y) = \begin{cases} 1 & , \text{ If } (MM_C(x, y) = 1 \text{ and } R(x, y) = 1) \\ 0 & , \text{ otherwise} \end{cases} \quad (3-19)$$

Finally, the remained regions, the shadows,  $MM_R(x, y)$  are used to update the moving object mask by

$$\begin{cases} MM(x, y) = 0 & , \text{ If } (MM_R(x, y) = 1 \text{ and } MM(x, y) = 1) \\ \text{do nothing} & , \text{ otherwise} \end{cases} \quad (3-20)$$

As shown in Fig. 3-13(d), shadows are presented by cyan color. And then the horizontal run-length process is performed on the lane mask  $R(x, y)$ . Eventually, the result of shadow suppression in the ROI is shown in Fig. 3-14.



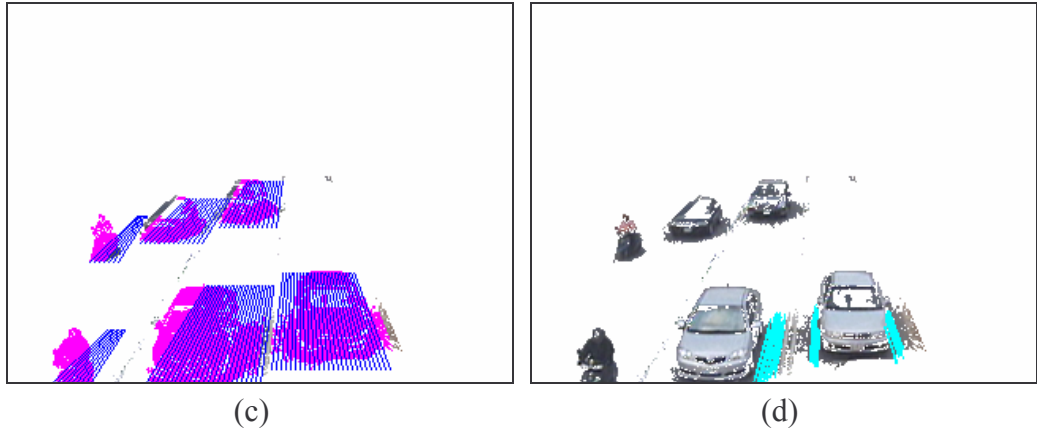


Fig. 3-13 The process of lane-based run-length shadow confirming processing



Fig. 3-14 The result of shadow suppression in the ROI

### 3.4.2 Shadow Suppression in Each Label

On the other hand, if the proposed shadow suppression method above is not effective in some cases, there are some false moving objects (shadows) appeared in the tracking process as shown in Fig. 3-15. In general, shadows of vehicles split by lane information would be thought as another moving object. It is hard to eliminate the shadow moving object in that its size and width-height ratio is similar to a vehicle.



Fig. 3-15 The samples of the effect of large shadows

Checking the gradient of every tracking node, the normalized gradient (NG):

$$NG(l) = \sum_{G(x,y) \in l} G(x,y) / p(l) \quad (3-21)$$

is utilized, where  $l$  is a Label ID, and  $p(l)$  is the total pixels of the bounding box with the same Label ID  $l$ . According to Fig. 3-16(b) and (d), the shadow moving object is deleted.

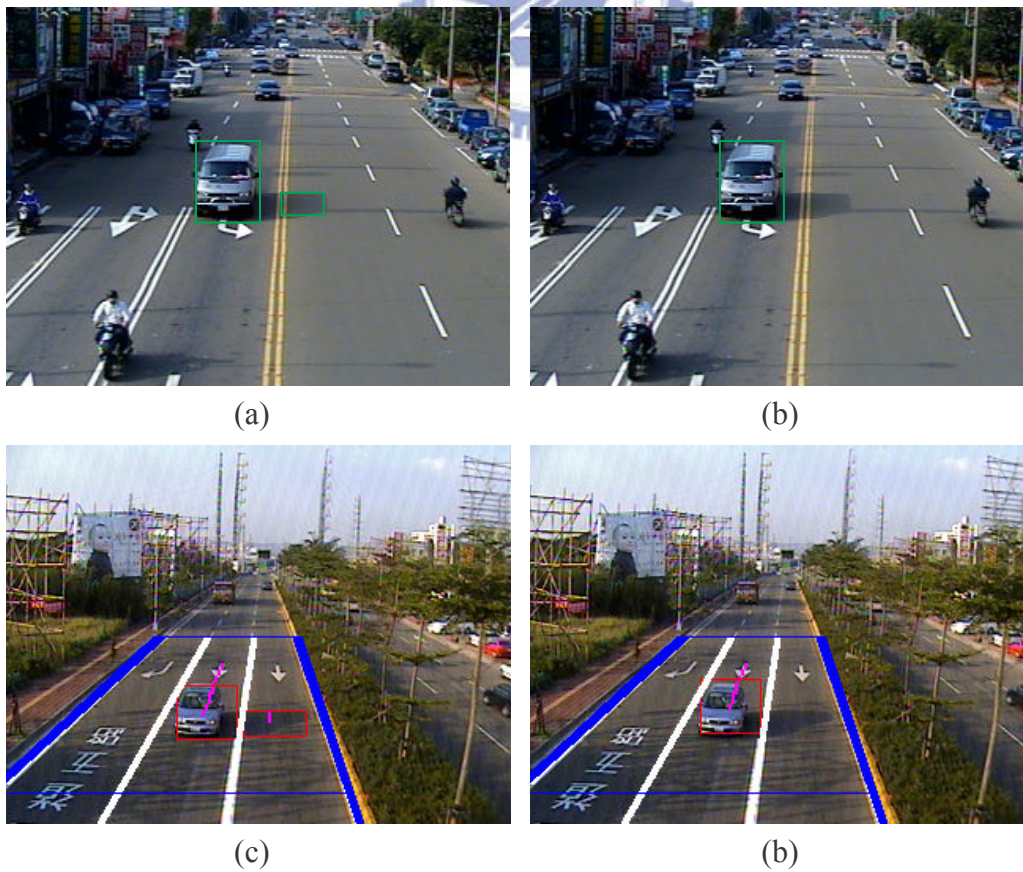
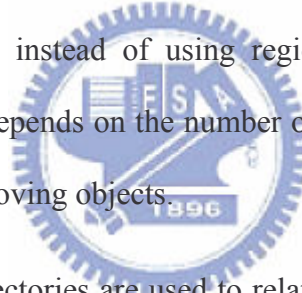


Fig. 3-16 The result of shadow suppression in each label.

### 3.5 Update Trajectories and Eliminate Vibrated Moving Objects

A tracking trajectory is simply a sequence of subsequent tracking nodes (moving objects) which satisfy the relation constraints. The label ID  $l$  should be extended to  $l(k, t)$  to become a function of the trajectory ID  $k < NT(t) \in \mathbf{N}$  and the time instance  $t$  where  $NT(t)$  is the number of trajectories at the time instance  $t$ . Then, the  $k^{\text{th}}$  trajectory at time instance  $t$  with  $NN(k, t)$  nodes is denoted as  $T(k, t) = \{l(k, t - NN(k, t) + 1), \dots, l(k, t - 1), l(k, t)\} = T(k, t - 1) \cup \{l(k, t)\}$ , where  $NN(k, t)$  is the number of nodes of  $k^{\text{th}}$  trajectory at time instance  $t$ .

In order to archive real-time issue, the center of each moving object is used. The benefit of using centers of moving objects instead of using regions of moving objects is that the computation complexity merely depends on the number of moving objects but the product of current and previous number of moving objects.



First of all, the centers of trajectories are used to relate current moving objects to current existed trajectory. The creation and extension of a trajectory is executed by checking the relation between the current moving object and the last node of the existed trajectory. The *Euclidean distance*:

$$Euclidean(\mathbf{P}(l, k, t)) = |\mathbf{P}(l, k, t)| = \sqrt{|C(l) - C(k, t - 1)|^2} \quad (3-22)$$

is utilized to check the relation between the center of current moving object and the last trajectory node, where

$C_L(l)$  is the related center of the  $l^{\text{th}}$  moving object,

$C_T(k, t)$  is the related center of a node in the  $k^{\text{th}}$  existed trajectory at time instance  $t$ , and

$$\mathbf{P}(l, k, t) = \mathbf{C}_L(l) - \mathbf{C}_T(k, t-1).$$

In addition, the angle between the current moving object and the last node of the existed trajectory is checked in that the vehicle-like moving object wouldn't move in opposite direction immediately. If  $NN(k, t)$  exceeds one then the angle with respect to the  $l^{\text{th}}$  moving object,

$$AC(l, k, t) = \mathbf{P}(l, k, t) \cdot \mathbf{Q}(k, t) - |\mathbf{P}(l, k, t)| |\mathbf{Q}(k, t)| \cos(TH_\theta) \quad (3-23)$$

where

$$\mathbf{Q}(k, t) = \mathbf{C}_T(k, t-1) - \mathbf{C}_T(k, t-2). \quad (3-24)$$

are checked. If  $AC(l, k, t) > 0$ , then the  $l^{\text{th}}$  moving object will satisfy the angle constraint  $TH_\theta$  with the  $k^{\text{th}}$  trajectory at time  $t$ . The diagram (Fig. 3-17) illustrates the distance and angle constraint. Accordingly, in authors' experience, a reasonable value of  $TH_\theta$  is  $60^\circ$ .

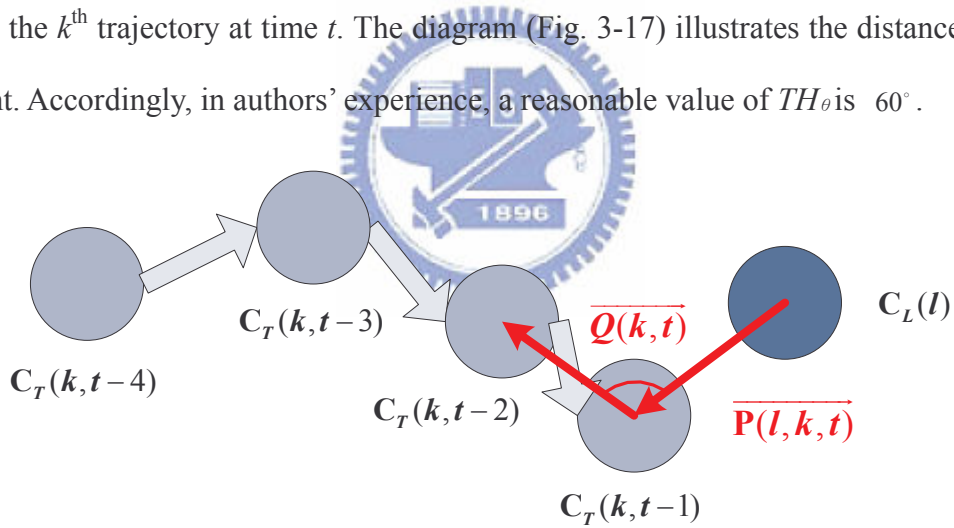


Fig. 3-17 The diagram of the relation between tracking trajectory and current moving object.

If the distance and angle constraint are both not conformed, the further judgment for updating the tracking trajectory -intersection of the current moving object and the last node of the existed trajectory -is offered.

In order to reducing the computing power,  $\Delta x_T$ ,  $\Delta y_T$ ,  $\Delta x_B$ , and  $\Delta y_B$  in Fig. 3-18 represented



the degree of intersection are utilized; i.e. if these four distance parameter are small enough, the intersection of  $C(l)$  and  $C(k,t-1)$  are large that we can safely arrive that  $C(l)$  is the next tracking node of the tracking trajectory  $C(k,t)$  without calculating the real intersection.

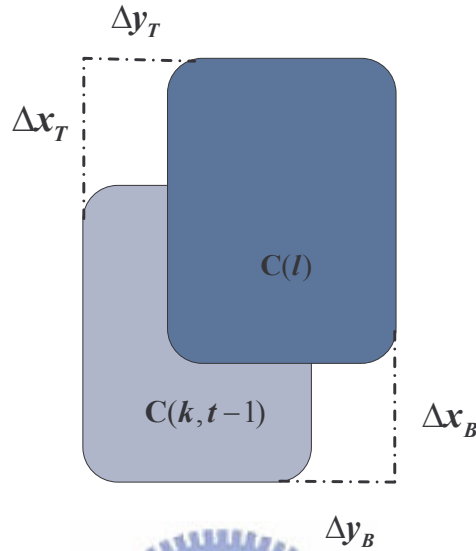


Fig. 3-18 The diagram of intersection of the current moving object and the last node of the existed trajectory

However, if a moving object satisfies the distance constraint but does not satisfy the angle constraint, then a vibration counter associated with the trajectory will be increased by 1. If the vibration counter of the trajectory exceeds 3, then the trajectory will be thought as that of vibrating moving objects. That is, the trajectory will be ignored.

Eventually, any moving object that can not be related to an existed trajectory should further check whether it is occluded or not. The schemes applied to detect occlusion and to resolve the occlusion are presented in the following sub-section. Otherwise, a new tracking trajectory will be created.

### 3.6 Traffic Parameters Calculation

In general, most traffic parameters can be derived by the tracking trajectories. However,

each trajectory has to classify to a lane ID before calculating traffic parameters. The method used to classify a trajectory to a lane ID is specified as (3-17), where the way to obtain the histogram  $S(k, t, h)$  is similar to (3-17) where  $l$  is extended to  $l(k, t-1)$  to indicate the  $k^{\text{th}}$  trajectory at time instance  $(t-1)$

Besides, there are some major traffic parameters: speed ( $VS$ ), quantity ( $VQ$ ), headway ( $VH$ ), volume ( $VV$ ), and occupancy ( $VO$ ). The speed ( $VS$ ) of a detected vehicle obtained by tracking trajectory is the average speed in the ROI. The quantity ( $VQ$ ) of each lane means the amount of vehicles in each lane and the headway ( $VH$ ) is the distance between the rear of a front vehicle and the head of a rear vehicle. In addition, the amount of vehicles divided by time is volume ( $VV$ ) and the occupancy ( $VO$ ) is the percentage of the amount of vehicles divided by time.

Equations used to calculate traffic parameters are listed in Table 3-2. Note that the moment we calculate the traffic parameters is at the moment we delete the trajectory. Hence, the last node of the just deleted trajectory is at time instance  $t-1$ .

Table 3-2 Equations used to calculate traffic parameters.

Traffic Parameters	Equations
Speed: $VS(h^*)$	$VS(h^*(k,t)) = \frac{1}{8}VS(h^*(k,t-1)) + \frac{7}{8} \times \frac{ C(k,t-1) - C(k,t-N(k,t-1)) }{N(k,t-1) \times FPH} \times \frac{0.005}{\bar{W}(k)}$ <p>where <math>FPH</math> is the number of frame per hour; <math>N(k, t)</math> is the number of nodes in the <math>k^{\text{th}}</math> trajectory at time instance <math>t</math>; <math>C(l)</math> is the center of the <math>l^{\text{th}}</math> moving object; <math>\bar{W}(k)</math> is the average width of nodes in the <math>k^{\text{th}}</math> trajectory.</p>
Quantity: $VQ(h^*)$	<p>If <math>N(k, t-1) &gt; 3</math>, then</p> $VQ(h^*(k, t)) = VQ(h^*(k, t-1)) + 1$

Headway: $VH(h^*)$	<p>First, initialize <math>t_H(h) = 0</math> for all lane ID <math>h</math>.</p> $VH(h^*(k,t)) = \frac{t - t_H(h^*(k,t-1))}{FPH} \times VS(h^*(k,t-1))$ $t_H(h^*(k,t)) = t$
Volume: $VV(h^*)$	$VV(h^*(k,t)) = \frac{VQ(h^*(k,t-1)) \times FPH}{t}$
Occupancy: $VO$	<p>If <math>N_t(k,t-1) &gt; 3</math>, then</p> $OF = OF + 1$ $VO = \frac{OF}{t} \times 100\%$

The direction of vehicle movement is also important traffic information. In the cases of two-direction road scene, the direction of vehicle movement will help us to count the quantities in or out correctly that the traffic parameter will more make sense. In Fig. 3-19,  $x$  and  $y$  are the coordinate of the center of a vehicle, and there are four possible directions in general: top to bottom, bottom to top, left to right, and right to left. The four directions are defined by two lines:  $\cot \frac{\varphi}{2} x + y = 0$  and  $\cot \frac{\varphi}{2} x - y = 0$  with angle  $\varphi$ , furthermore, the regions of four directions are shown in Fig. 3-19.

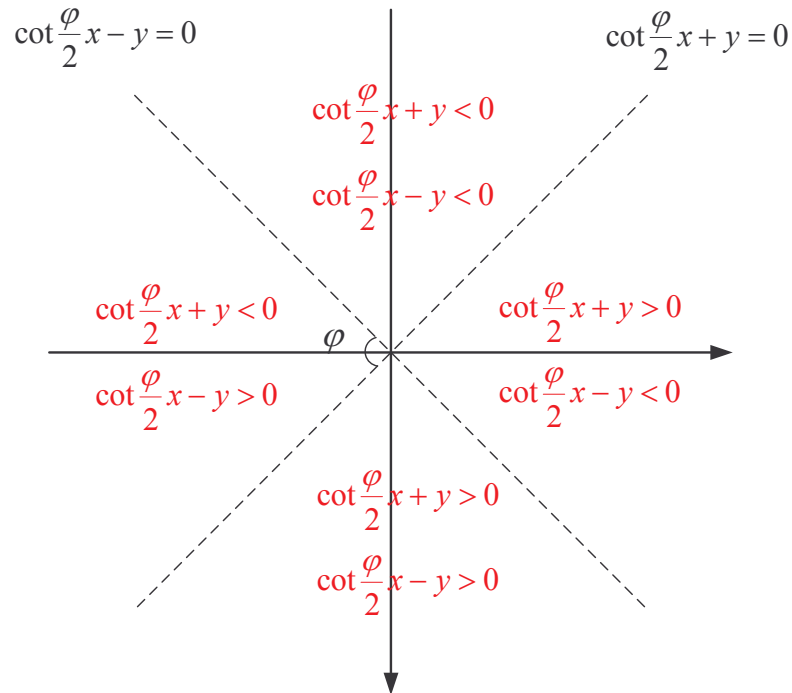


Fig. 3-19 The diagram of direction classification.

The angle  $\phi$  in Fig. 3-19 is represented as the acceptable range of each four directions. That is, if  $\phi$  is small, the acceptable range of direction of bottom-to-top and top-to-bottom vehicles is boarder than left-to-right or right-to-left vehicles.

### 3.7 Parameter Adaptation

In order to adapt the MVDT system to diverse capture view conditions, such as lane direction and vehicle size, all parameters applied to the system must be decided adaptively. In this work, the average of weighted small vehicle width or height mean is referred to tune the system parameters which are thought as a separation between the width of small vehicles and large vehicles. This parameter automation processing is configured each time before the system starts-up. Selecting the center region (green line to magenta line in Fig. 3-20(a)) of half image, the vehicle height and width statistics are gathered in the region. The statistic result is shown in Fig. 3-20(b) which can calculate the mean of vehicle width and height. The mean of the vehicle width or height is thought as a separation between small vehicles and large vehicles.

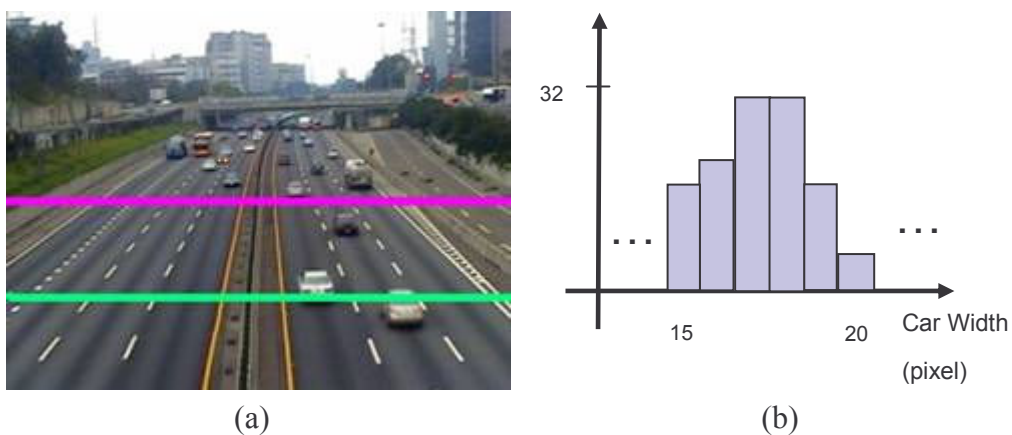


Fig. 3-20 The statistics of vehicle width and height.

### 3.8 Tracking Stability improvement

In order to improve the accuracy of traffic parameter and ensure the stability of the tracking processing, some serious problem should be resolved such as losing the tracking node during tracking processing, over counting, and so on.

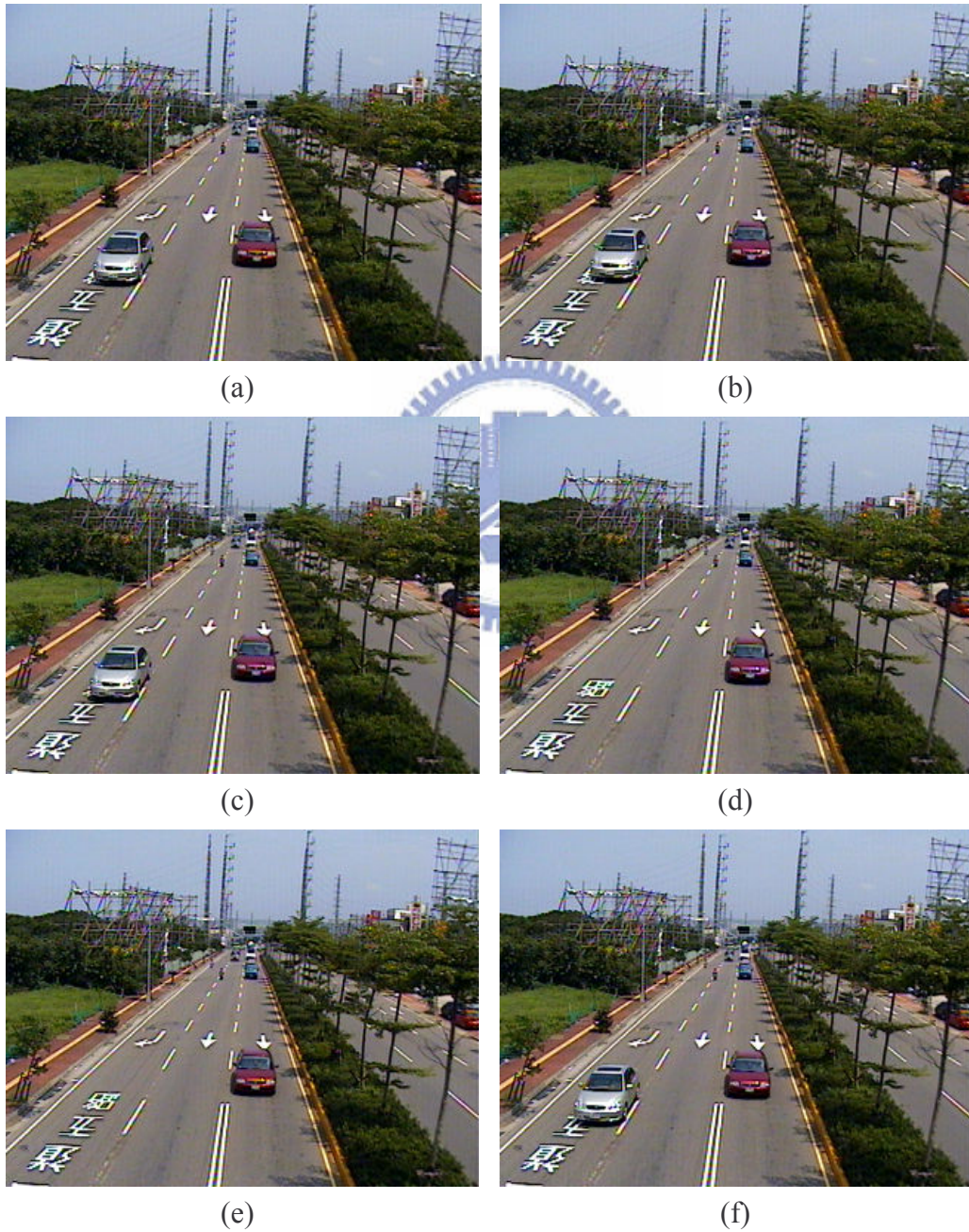


Fig. 3-21 (a)~(f) the image sequence which the silver vehicle is not extracted in (d) and

(e).

First, over counting is a serious problem due to mis-detected vehicles. If moving objects are not extracted during segmentation or not found during tracking processing, the according trajectories would be deleted as shown in Fig. 3-21(a)~(f). Tracking trajectories would be reserved until the moving objects are not found or segmented for two or three frames in that the moving might not be segmented in the previous processing due to the unapparent information of original images. Fig. 3-21(a) shows that the tracking trajectory is deleted immediately if the next tracking node is not found in (a). And Fig. 3-21(b) is the result of reserving tracking node.

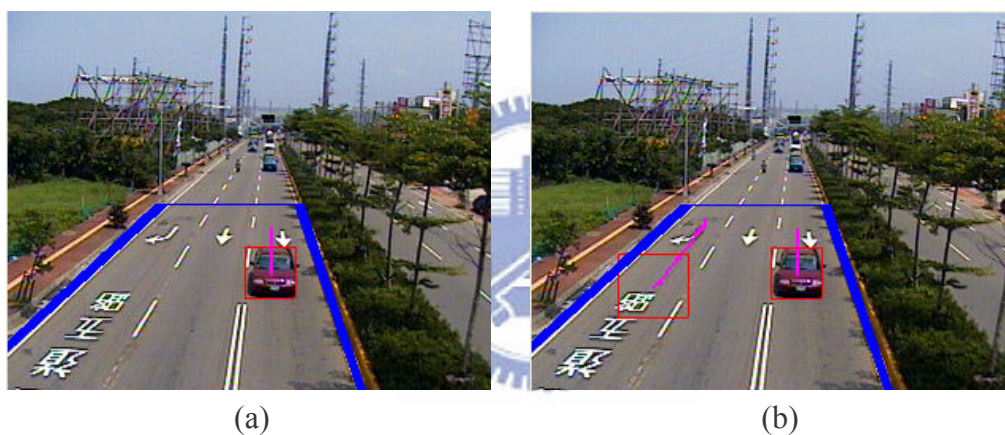


Fig. 3-22 The tracking trajectory is deleted immediately if the next tracking node is not found in (a) and (b) is the result of reserving tracking node.

In addition, the accuracy of speed is another important problem. When the vehicles stopped during the tracking processing, tracking nodes might be in the same position. It would be incorrect to calculate speed based on the traffic parameter calculating method because of the amount of tracking nodes.

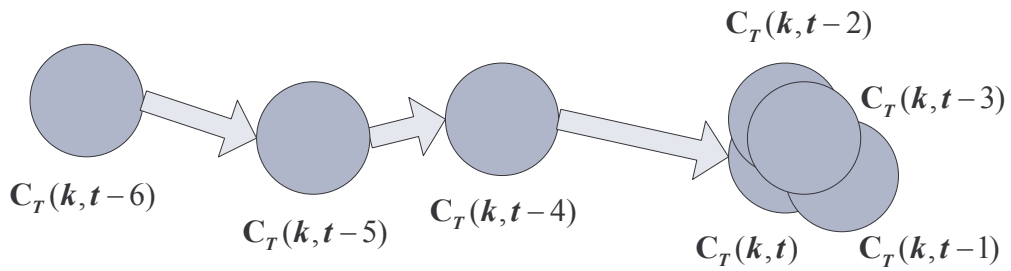


Fig. 3-23 Tracking nodes from vehicle moving slowly or stopping.

As shown in Fig. 3-23, there are some tracking nodes of a vehicle which is attempting to stop in ROI (region of interest). The distance and the amount of tracking nodes are used to calculate the speed of the vehicle. Therefore, the serial number of tracking nodes should be given the same one, if the vehicle is stopping. Also, the number of squeeze nodes ( $Sq$ ) of each tracking trajectory is recorded to assure of the tracking processing as shown in Fig. 3-24.

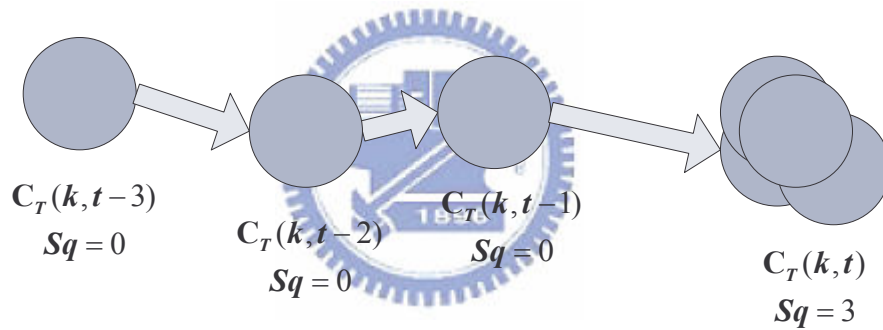


Fig. 3-24 An example of squeeze tracking nodes for vehicle stopping or moving slowly case.

# Chapter 4 Queue Resolution

## 4.1 Queue Detection

For queue detection, the trajectory information is used. If the distance between two subsequent trajectory nodes is shorter than a previous given threshold, vehicle queue might be occurred:

$$\begin{cases} \text{Queue Resolution} & \text{if } |C_T(k,t) - C_T(k,t-1)| \leq TH_d \\ \text{Tracking Processing} & \text{otherwise} \end{cases} \quad (4-1)$$

where  $TH_d$  is the threshold of the distance between two subsequent trajectory nodes.

Otherwise, the tracking trajectory might be disappeared if the moving objects are not fitting the tracking conditions. Then, the height of the label is utilized to decide the queue-like objects:

$$\begin{cases} \text{Queue Resolution} & \text{if } H(l) \geq TH_H \\ \text{Tracking Processing} & \text{otherwise} \end{cases} \quad (4-2)$$

where  $TH_H$  is the threshold of the reasonable height of the label.

## 4.2 Vehicle Head and Rear Extraction

As shown in Fig. 4-1, vehicles are always occluded at queuing. That is because the rear of a vehicle is occluded to the head of the next vehicle. There is no tracking information to use to separate distinguishing vehicles at this time. For such a reason, a prior queue detection and resolution technique that utilizes the edge feature of vehicle head or rear is proposed.





Fig. 4-1 The examples at queuing time

### 4.2.1 The Edge Property of Vehicle Head and Rear

The first stopped vehicle for each lane is always outside the detection zone. However, boundary objects would be deleted in the tracking processing. If the first stopped vehicle occluded with other vehicles in back of it, all vehicles in the lane would be deleted and not be counted. Therefore, the edge-based queue resolution is proposed.

The edge of a car rear and the edge of the next car head are used to split two occluded vehicles because it is obvious even when two or more vehicles are occluded. The Sobel horizontal gradient

$$G_x(Z_5) = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) \quad (4-3)$$

is firstly used to find the horizontal edges per lane, where  $Z_n, n=1 \dots 9$ , is defined in Fig. 4-2.

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

Fig. 4-2 The 3x3 mask for gradient calculation

The edges of a label are defined by:

$$\begin{cases} g(l,r,t) = 1 & \text{if } Gx \geq TH_G \\ g(l,r,t) = 0 & \text{if } Gx \leq TH_G \end{cases} \quad (4-4)$$

where  $g(l,r,t)$  is an edge mask of label  $l$  at time instance  $t$  and  $r$  is the row of the label  $l$ .

The row  $r$  in the edge mask  $g(l,r,t)$  with the amount of horizontal gradient greater than  $TH_G$  is considered as an available edge line shown with black lines in Fig. 4-3.



Fig. 4-3 The black lines in these images represent edges

Nevertheless, the available edge lines are not shown in every row even the row is an edge. A run-length method is utilized to group the found edges in edge mask  $g(l,r,t)$ . Fig. 4-4(a) shows edge lines without run-length processing and Fig. 4-4(b) shows the result of run-length processing. Therefore, the threshold of run-length is 4 in authors' experimental.

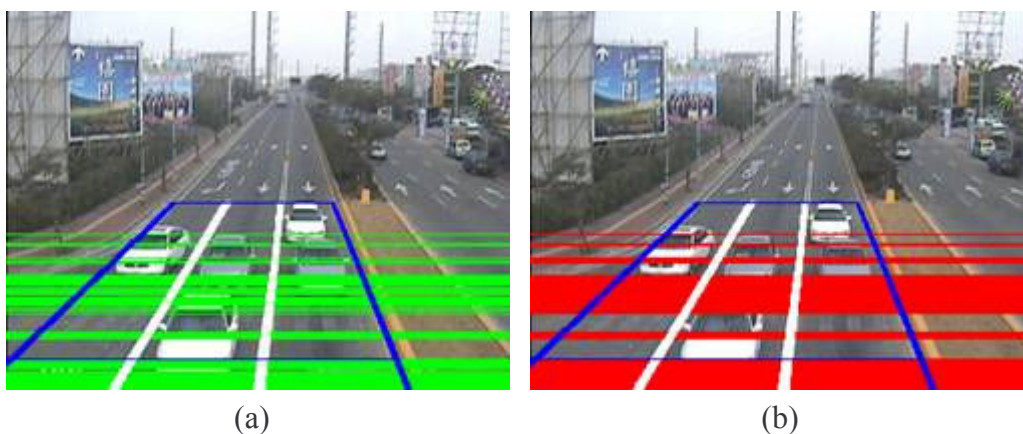


Fig. 4-4 The red lines in (b) are the results of green lines in (a) with run-length processing.

After run-length processing, the preliminary split line is located on the middle of notably grouped edge lines. The edges of vehicles are represented as magenta lines in each lane in Fig. 4-5.



Fig. 4-5 The preliminary results of stopped vehicle split.

## 4.2.2 Color Space Analysis for Gradient Computation

In order to calculate gradient to find edges of vehicles, some candidate color spaces are selected to experiment with which one is more discriminative. In our experiment, we collect common colors of vehicles (Fig. 4-6) and some combinations of them (Fig. 4-7).

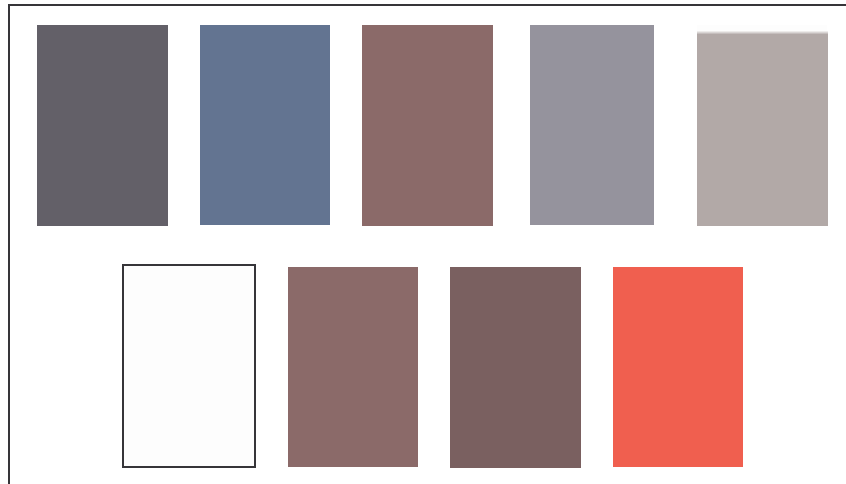


Fig. 4-6 The general colors of vehicles.

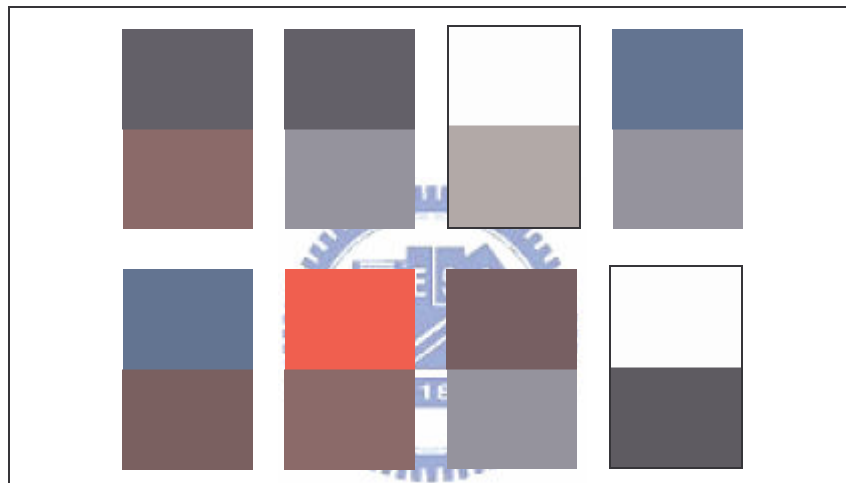


Fig. 4-7 Two general color blocks combined represent two occluded vehicles.

#### 4.2.2.1 YUV Color Space

The YUV model defines a color space in terms of one luminance and two chrominance components. YUV is used in the PAL and NTSC systems of television broadcasting, which is the standard in much of the world.

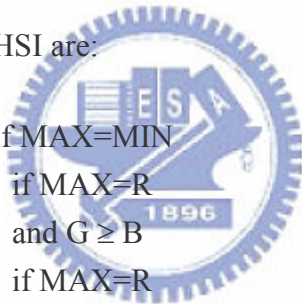
YUV models human perception of color more closely than the standard RGB model used in computer graphics hardware. YUV signals are created from an original RGB (red, green and blue) source. The weighted values of R, G and B are added together to produce a single Y

signal, representing the overall brightness, or luminance, of that spot. The U signal is then created by subtracting the Y from the blue signal of the original RGB, and then scaling; and V by subtracting the Y from the red, and then scaling by a different factor.

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ U &= 0.492(B - Y) \\ V &= 0.877(R - Y) \end{aligned} \tag{4-5}$$

#### 4.2.2.2 HSI Color Space

The HSI color space which is non-linear deformations of the RGB color cube stands for Hue, Saturation, Intensity. The angular parameter corresponds to hue, distance from the axis corresponds to saturation, and distance along the black-white axis corresponds to intensity (Fig. 4-7). The equations of RGB-HSI are:



$$H = \begin{cases} \text{undefined} & \text{if MAX=MIN} \\ 60 \times \frac{G-B}{\text{MAX-MIN}} + 0, & \text{if MAX=R} \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{\text{MAX-MIN}} + 360 & \text{if MAX=R} \\ & \text{and } G < B \\ 60 \times \frac{B-R}{\text{MAX-MIN}} + 120 & \text{if MAX=G} \\ 60 \times \frac{R-G}{\text{MAX-MIN}} + 240 & \text{if MAX=B} \end{cases}$$

$$S = \begin{cases} 0 & \text{if MIN=MAX} \\ \frac{\text{MAX-MIN}}{\text{MAX+MIN}} = \frac{\text{MAX-MIN}}{2L} & \text{if } 0 < L \leq \frac{1}{2} \\ \frac{\text{MAX-MIN}}{2-(\text{MAX+MIN})} = \frac{\text{MAX-MIN}}{2-2L} & \text{if } L > \frac{1}{2} \end{cases} \tag{4-6}$$

$$I = \frac{(\text{MAX+MIN})}{2}$$

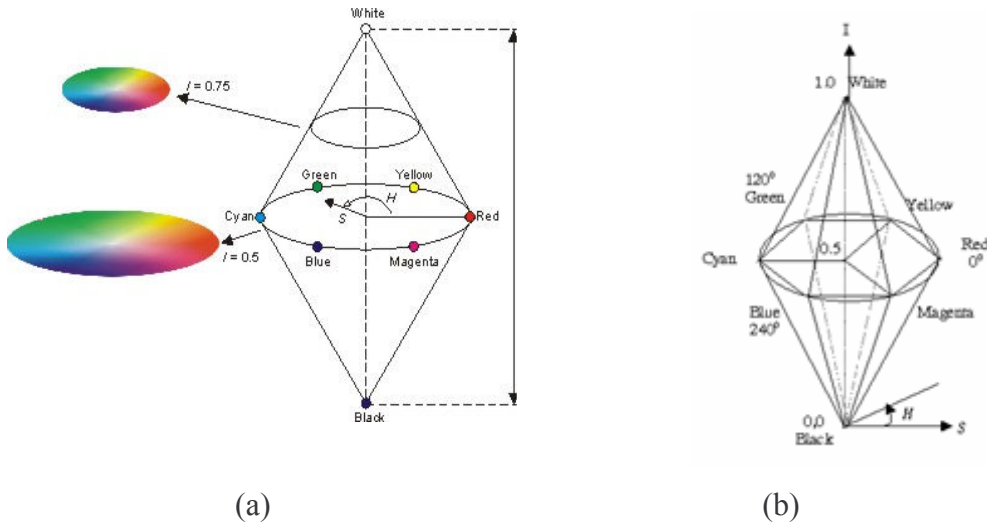


Fig. 4-8 The HSI color model based on (a) circular color planes (b) hexagon color planes.

The range of Hue is  $0^\circ \sim 360^\circ$  as shown in Fig. 4-8(a). However, the angle of two similar colors might be large. So, we formulate the gradient of Hue in:

$$\text{Min}(\text{abs}(H_A - H_B), \text{abs}(H_A - H_B - 360^\circ)) \quad \text{where } H_A > H_B \quad (4-7)$$

where  $H_A$  and  $H_B$  are hue of two different colors.

Fig. 4-9 shows the results of calculating gradients in G, H, S, I, Y, U, and V color planes. The range of gradient is normalized from 0 to 255 according to the intensity of gradient value. If the edge property is more obvious, the value of gradient is closed to 255. Based on Fig. 4-9, it is conspicuous that H, I, and U color planes are more discriminative than others.

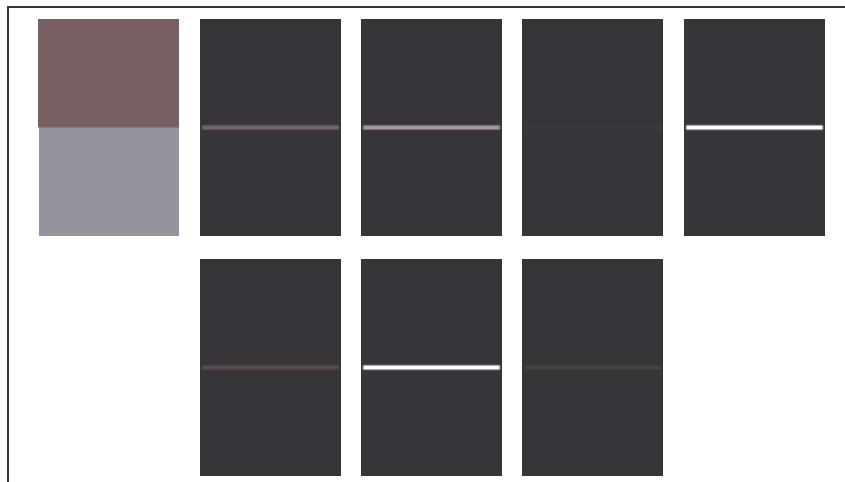


Fig. 4-9 The results of gradient calculation in G, H, S, I, Y, U, and V color planes.

### 4.3 Queue Splitting Refinement

However, it is difficult to distinguish which split lines are the ones on vehicle heads or rears. There are some fake split lines on the vehicles. That is because large vehicles are with different edge properties from small vehicles or some vehicles have a lot of strong edges in their windows and sunroofs. The wrong cases are shown in Fig. 4-10.



Fig. 4-10 The wrong cases of occluded vehicle split.

First, a region below the center of a split line is selected to apply horizontal gradient. Then, if the selected region is smooth, a split line nearby windows (i.e. a fake split line) will be detected and will be removed. Otherwise, the split line is utilized to separate occluded vehicles. A queued vehicle with a split line (the magenta line) and a region (the cyan dashed-line box) is shown in Fig. 4-11(a) and (b). The horizontal gradient of Fig. 4-11(a) without the split line and the region is shown in Fig. 4-11(b).



Fig. 4-11 The wrong cases of occluded vehicle split.

After filtering out fake lines on the window, the correct split line is obtained. Fig. 4-12(a) displays a fake line on the white vehicle window and the fake line eliminated by the above processing is shown in Fig. 4-12(b).

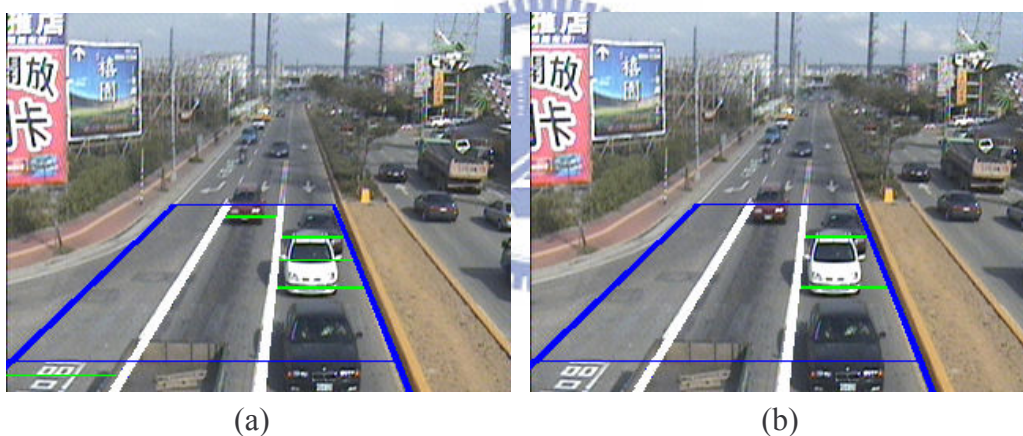


Fig. 4-12 A green line located on windshield in (a) is eliminated in (b).

There are also some edges located on vehicle sunroofs which should not be split. Because the edge property of a sunroof is similar to the one of a vehicle head, it is hard to be eliminated. However, a sunroof has a notable horizontal edge property but unapparent vertical property comparing to a car head. So, the vertical gradient of a small region is applied to eliminate fake split lines on car sunroofs. Before applying vertical gradient, a rectangular region below the center of a candidate split line is picked. But, the feature of vertical gradient is easily affected by moving direction. Fig. 4-13(a), the enlarged drawing of Fig. 4-13(b),



shows the color of a sunroof changing gradually resulted in the appearance of adjacent vertical gradients. Consequently, the region with vertical gradients connected one by one horizontally is filtered out (red points in Fig. 4-13(b)). The fake split line located on a car sunroof is removed in Fig. 4-14.



Fig. 4-13 The characteristics of sunroof, that is, the points of vertical gradient are connected on by one horizontally.



Fig. 4-14 The fake green split line located on a vehicle sunroof in (a) is removed in (b).

After connected-component labeling processing, non-vehicle objects would be filtered out. Then, queue detecting processing and object-oriented queue resolution are applied to each label.

After choosing the highly discriminant color space to calculate gradient, there are some edges of noise objects would be enhanced. Smoothing the moving object to reduce the edges

of noise, the captured image with illuminate variation on the single-chrominance part will be smoothed. Using the 13 colors to represent the vehicle in the original image in Fig. 4-15, the smoothed image (Fig. 4-15(b)) shows that the middle region in the red bounding box are presented by only one color which means there is no edge property to interfere our judgment. Otherwise, in Fig. 4-15(a), the same region is shown by different color to demonstrate that this region is not smooth.

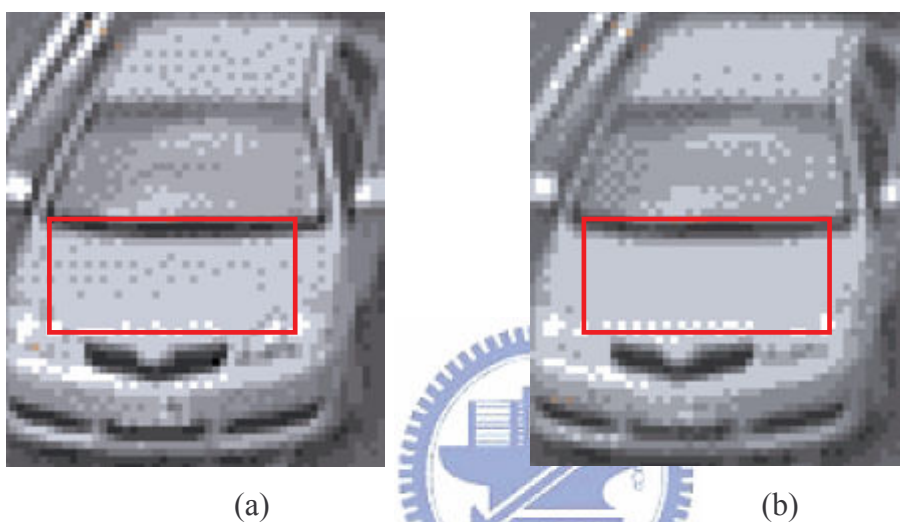


Fig. 4-15 The stopped vehicle before and after the smoothing processing

According to the filtering-out-the-fake-line processing, the front of vehicle window should be smooth to distinguish the real split line and fake split line accurately. The result is shown in Fig. 4-16 which the fake split line located on the vehicle window in Fig. 4-16 (a) is eliminated in Fig. 4-16(b)



Fig. 4-16 The result of fake split line elimination.



# Chapter 5 Experimental Results

## 5.1 Results of Vehicle Detection

In order to capture the whole scene of lanes and vehicles, CCD should be built on the high location as shown in Fig. 5-1. This experimental platform on which we can stand is more convenient to execute real-time experiment and demonstration than the platform which CCD is built on street nameplates (Fig. 5-1). But the platform which CCD is built on street nameplates is suitable for collecting diverse test data because of the computer is placed near the road as shown in Fig. 5-1.



(a)



(b)



(c)

Fig. 5-1 One of the experimental environments and experimental equipments

In our experiment, some different road sections, weather conditions, and viewpoints of CCD are tested to demonstrate the robustness of the vehicle detection system. To begin with, Fig. 5-2 shows the detected result with bright sun. Since this test condition is almost the ideal one without any interference, such as vehicle shadow and sufficient light, the result is satisfactory that small vehicles, large vehicles, and motorcycles are all detected in Fig. 5-2(a)~(d). Therefore, the lane information is applied to split occluded vehicles across a lane but vehicle on a lane as shown in Fig. 5-2(b).

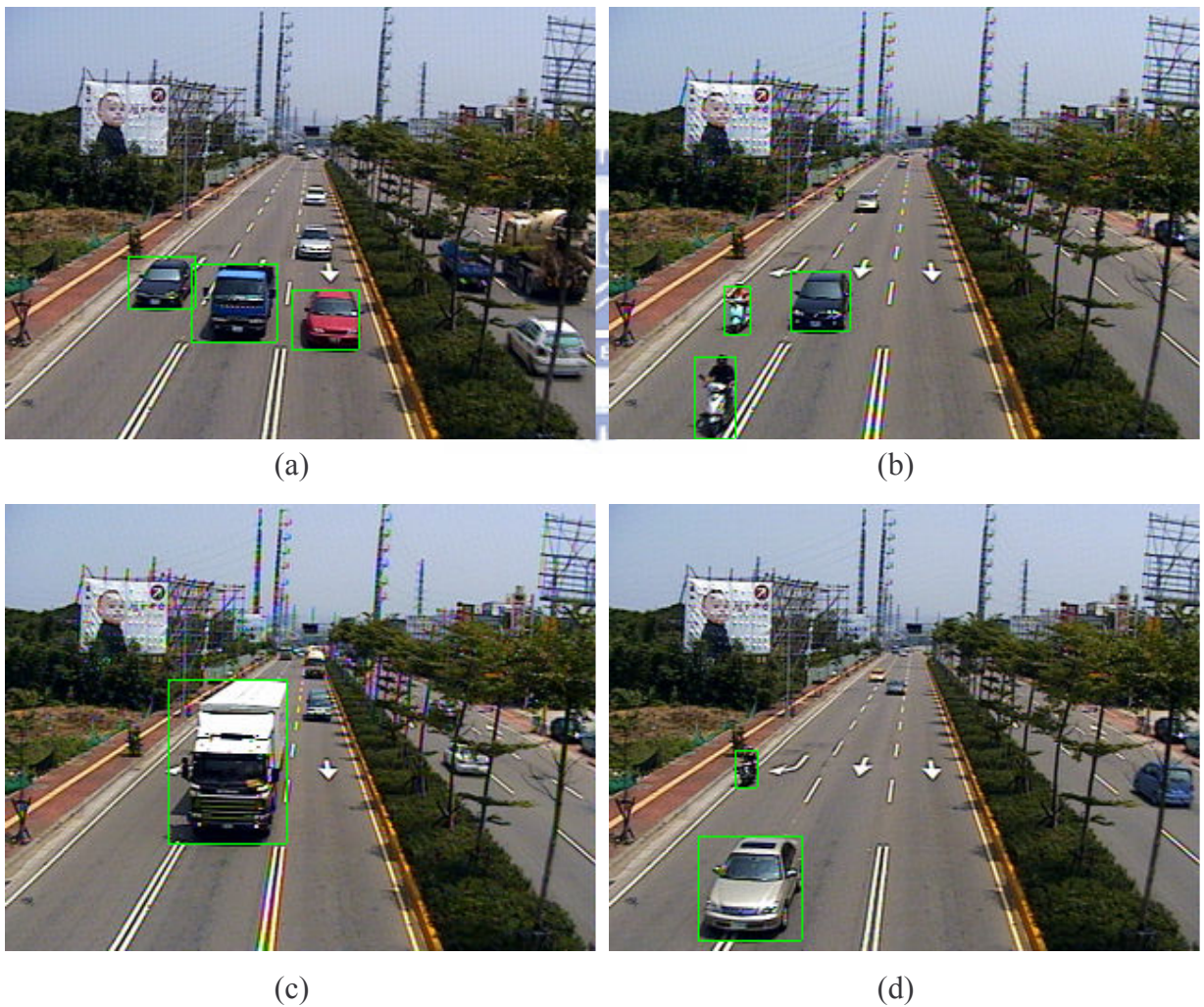


Fig. 5-2 Results of vehicle detection in the urban with sunlight.

In addition, the vehicle detection system is used in the sunset scene that the shadow effect is so serious both in the background and in the moving objects. Fig. 5-3(a)~(d) are all in the same image sequence within 2 minutes, but the illumination are varying immensely and the shadows of vehicles are enormous. However, both serious problems are resolved by the proposed methods above, and the result of the detected vehicle is highly accurate.



Fig. 5-3 Results of vehicle detection in the urban with serious shadows of vehicles and roadside buildings.

In the rainy weather, rain drops on the CCD lens will blur or hide some parts of image randomly and changeably. The proposed system is successfully to detect and identify the vehicles in the rainy day as shown in Fig. 5-4. On the other hand, the vehicle turning on the

head lights is also detected as shown in Fig. 5-4(c) and (d). There is a glob located not on the ROI in order that vehicles still can be detected successfully; however if a glob is on the ROI, vehicles are difficult to be noted by most image processing rules.

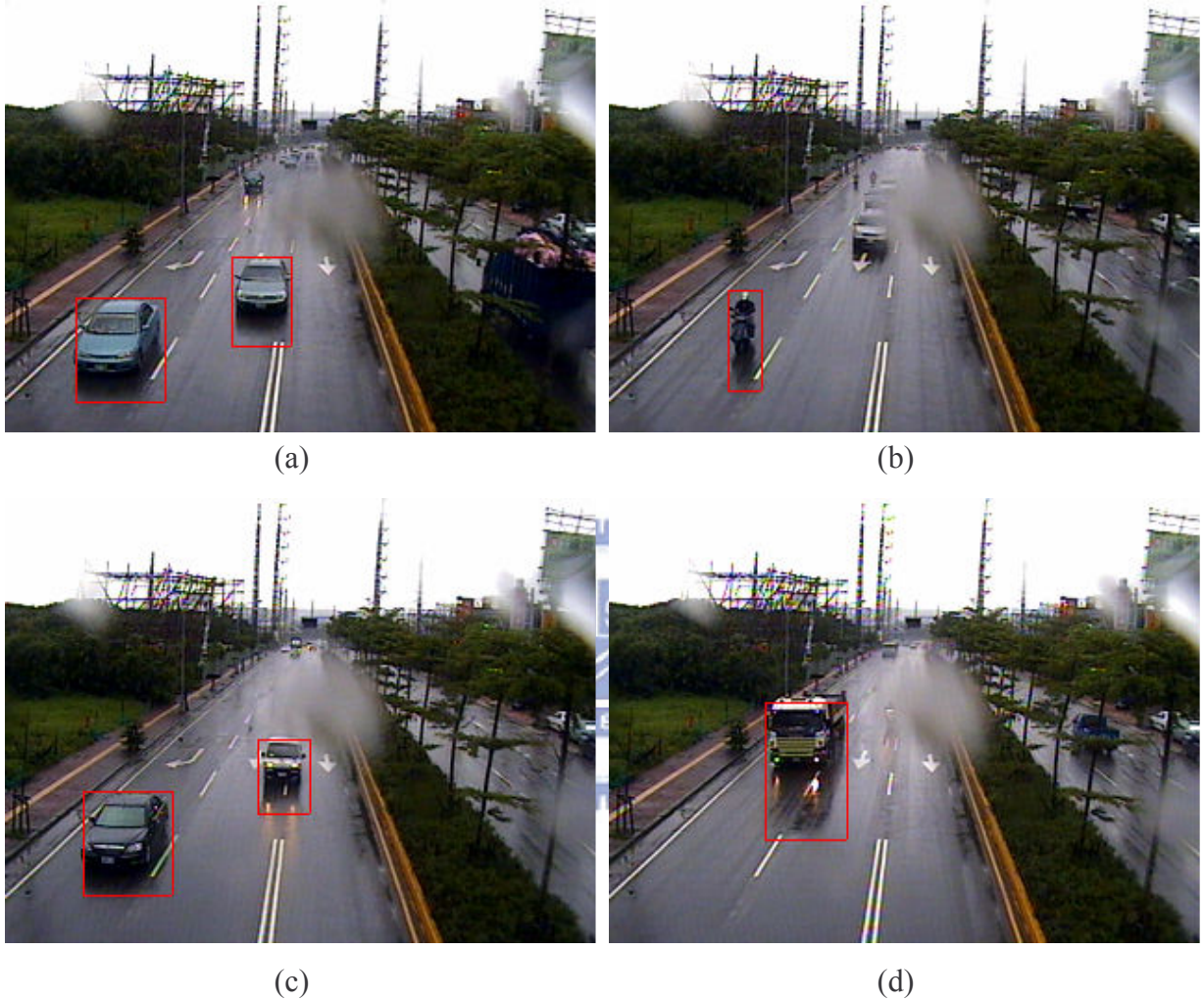


Fig. 5-4 Results of vehicle detection in the urban with vehicle headlights and rain drops.

Also, the proposed vehicle detection system can adapt to the expressway with 8 lanes and 2 different recognized directions. If there are lots of vehicle in a frame, the proposed system can also be performed in real-time because the proposed tracking reasoning method are low computation complication. Because vehicles are small and its characteristics are obscure in the image, it is tougher to detect and classify vehicle accurately. Furthermore, due to the lane information, the vehicles in the left side of Fig. 5-5(a) are detected correctly which

are occluded originally. However, small vehicles and large vehicles are detected well by proposed vehicle detection system.

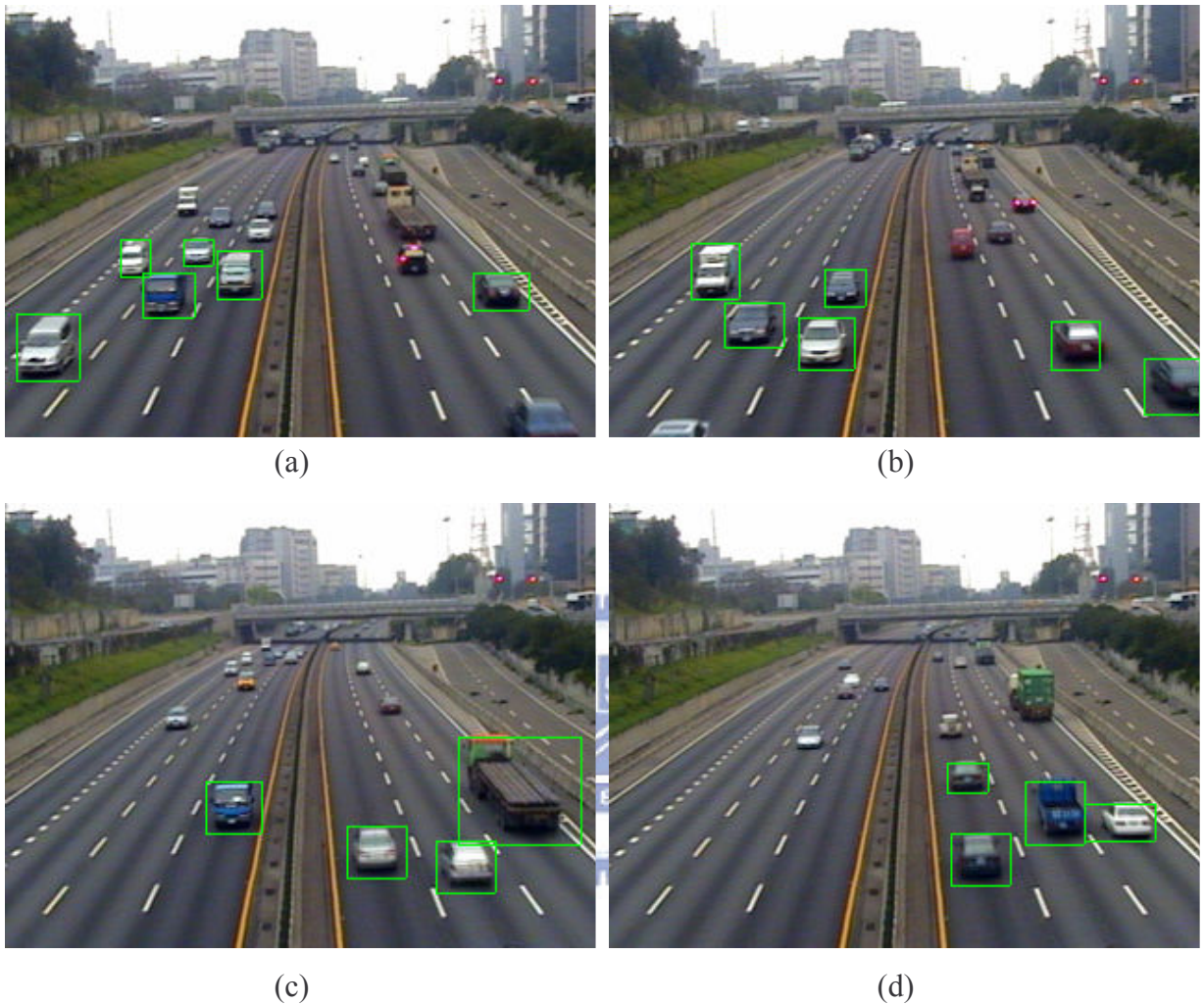


Fig. 5-5 Results of vehicle detection in the expressway.

Finally, straight lanes and a curve lane presented in the same road are shown in Fig. 5-6. Fig. 5-6(a)~(c) show that no matter a small vehicle or a large vehicle moves on a curve lane, it will be detected and tracked effectively. Of course, vehicles on the straight lanes are also detected at the same time.



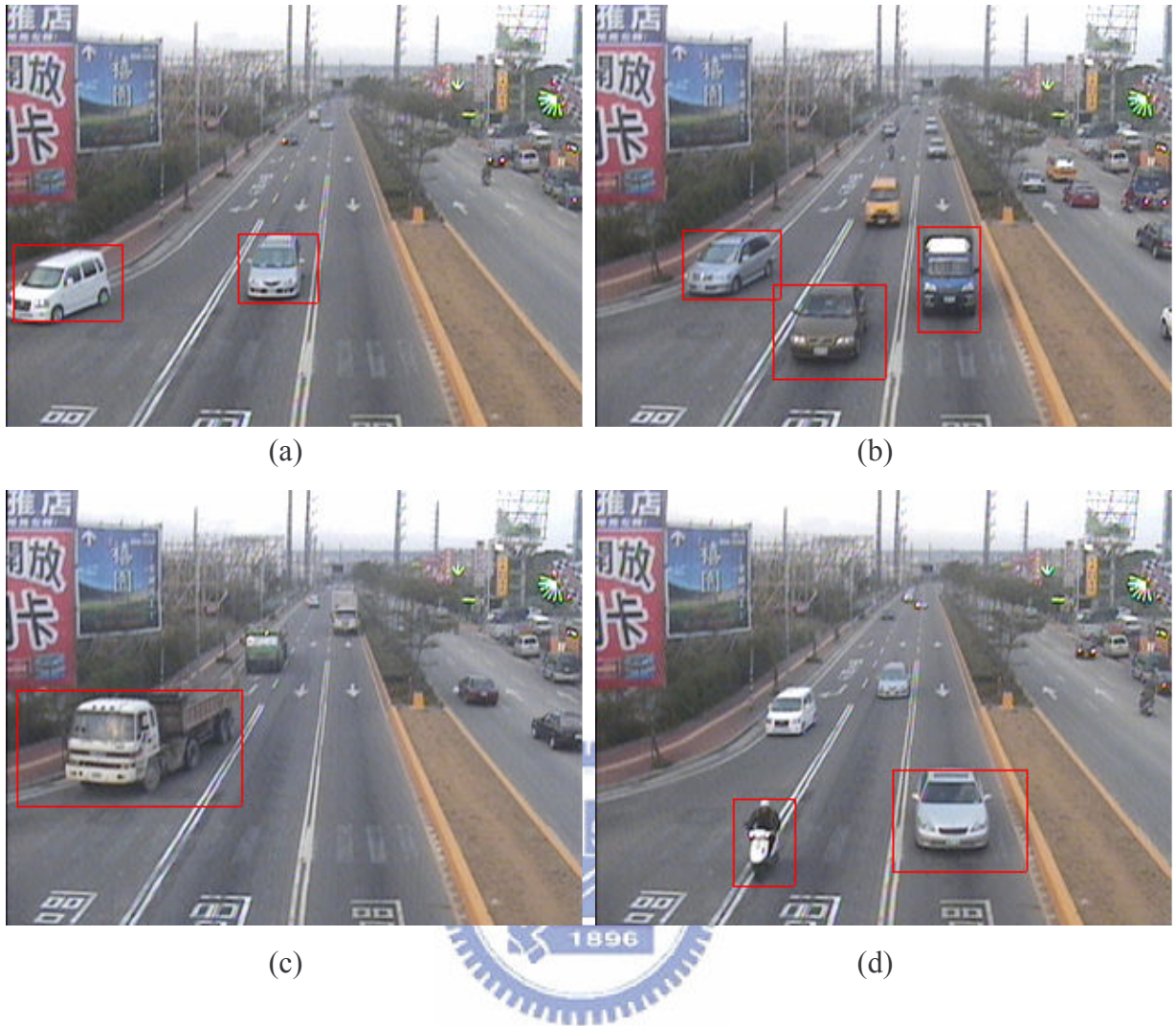


Fig. 5-6 Results of vehicle detection in the urban with curve and straight lanes.

## 5.2 Results of Traffic Parameter Calculation

Traffic parameter calculation is based on the tracking processing which provides the information of moving vehicles such as trajectories, motivation length, size, and motivation time. The formulae of traffic parameter are presented in Section 3.6. Fig. 5-7 and Fig. 5-8 show the result of several traffic parameters: the amount of both small and large vehicles, speed which is more important for providing traffic information, calculating by the proposed vehicle detection system in two different kinds of scenes: urban and expressway.



Fig. 5-7 Results of vehicle detection and traffic parameter calculation in the expressway.



Fig. 5-8 Results of vehicle detection and traffic parameter calculation in the urban.

Therefore, one long-time testing result of accuracy is shown in Table 5-1 which contains the quantity counted by both the proposed vehicle detection system and human and the detected and tracking rate. In the proposed system, the one of the classification rate of different vehicles is shown in Table 5-2. The average classification rate in that case is 89.7% which is lower than the detected rate in that there may be some misunderstanding of difference from large vehicles and small vehicles between human and the proposed vehicle detection system. Table 5-3 shows the accuracy rate of speed in three diverse lanes. The overall evaluation of proposed system is shown in Table 5-4 with 6 different test conditions. However, the evaluation of each test condition contains 3 time segments that a time segment is 5 minutes. Eventually, there are 6 parameters for a comparison to other techniques as shown in Table 5-5

Table 5-1 Average detection and tracking rate of total quantities of vehicles

Quantity	1 <sup>st</sup> lane	2 <sup>nd</sup> lane	3 <sup>rd</sup> land	Total
<b>The proposed VD System</b>	179	209	91	479
<b>Human</b>	183	205	86	474
<b>Error</b>	4	4	5	13
<b>Detected and tracking rate</b>	97.8%	98.0%	94.2%	97.3%

Table 5-2 Average classification rate of small or large vehicles

Quantity	1 <sup>st</sup> lane	2 <sup>nd</sup> lane	3 <sup>rd</sup> land	Total
The proposed VD System	134	151	75	360
Human	159	170	80	409
Error	25	19	5	49
classification rate	86.3%	90.7%	93.8%	88%

Table 5-3 Average speed detection rate of vehicles

	1 <sup>st</sup> lane	2 <sup>nd</sup> lane	3 <sup>rd</sup> land	Total
Speed accurate rate	96.9%	94.0%	95.8%	95.5%

Table 5-4 Overall evaluation of proposed system at 6 different test conditions

	Accuracy (%)
Detected and tracking rate (per 5 minutes)	94.34~100
classification rate (per 5 minutes)	89.4~100
Speed accurate rate	84.6~99.26

Table 5-5 Comparison to other techniques

	The [7]' s technique	The [18]' s technique	The autoscope system[21]	The proposed technique
<b>Average detection and tracking rate</b>	<b>96.9%</b>	<b>90%</b>	<b>95.39%</b>	<b>97.3%</b>
<b>Average classification rate</b>	<b>None</b>	<b>70%</b>	<b>None</b>	<b>89.7%</b>
<b>Average speed detection rate</b>	<b>None</b>	<b>4</b>	<b>97%</b>	<b>95.5%</b>
<b>Number of traffic parameters</b>	<b>1 (quantity)</b>	<b>1 (quantity)</b>	<b>5 (speed, quantity, headway, volume, occupancy)</b>	<b>5 (speed, quantity, headway, volume, occupancy)</b>
<b>Motorcycle detection</b>	<b>None</b>	<b>None</b>	<b>None</b>	<b>Yes</b>
<b>Queue resolution</b>	<b>None</b>	<b>None</b>	<b>None</b>	<b>Yes</b>

### 5.3 Parameter Adaptation



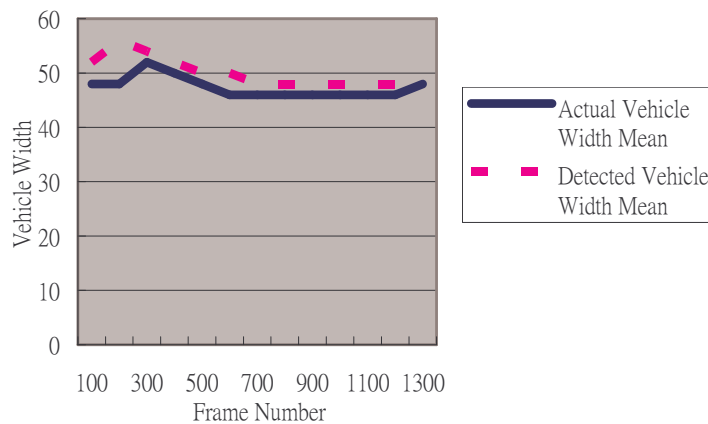
Parameter adaptation is used to adapt to different captured conditions. The strategy is to utilize an average vehicle height and width as the “parent” parameter of all other “children” parameters for tracking processing. Here, we illustrate two adaptive parameter examples in distinct captured conditions to reveal the importance of the processing.

After gathering the statistics of the vehicle width and height by both programs and human, the mean of them are calculated. Basically, the value of statistics is more precise if the statistic time is longer; however, if the statistics processing takes too much time, the later processing will be delayed. Therefore, we try to find out the best converged frame number which the detected vehicle height and width mean have the least error of actual vehicle height and width mean.

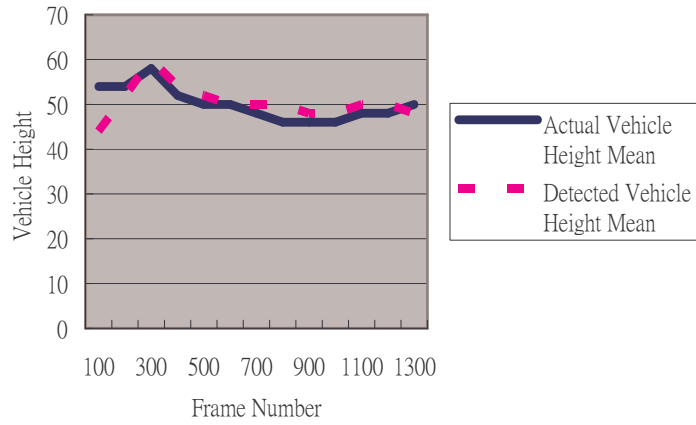
In other respects, the vehicle number is directly related to the result of vehicle width and height mean rather than the statistic time. Hence, the criterion of vehicle number is added to the statistics. Nevertheless, if there are few vehicles when we start up the system, the adaptive parameter processing might not be completed successfully. Either frame number or vehicle number is the terminative condition of adaptive parameter processing depending on which one is first reached.

Fig. 5-9 shows the mean of vehicle width and height of one of samples that the converged frame number is 700 and the converged vehicle number is 6 as shown in Fig. 5-10. Finally, we choose the frame number 700 or the vehicle number 6 as termination of this processing.

According to above experiments, the vehicle width is selected to be the main parameter of tracking processing which is more stable than the vehicle height.

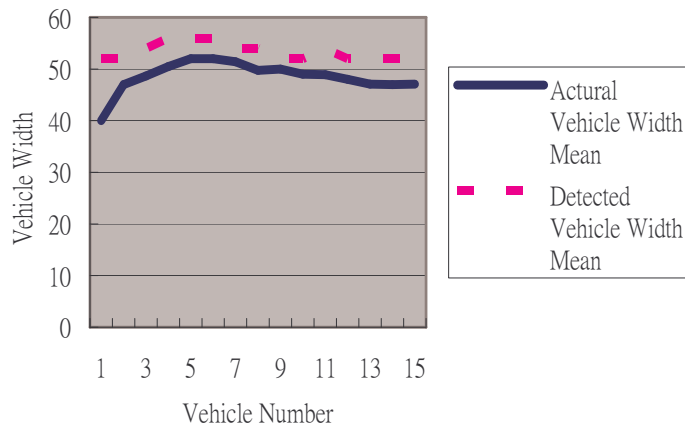


(a)

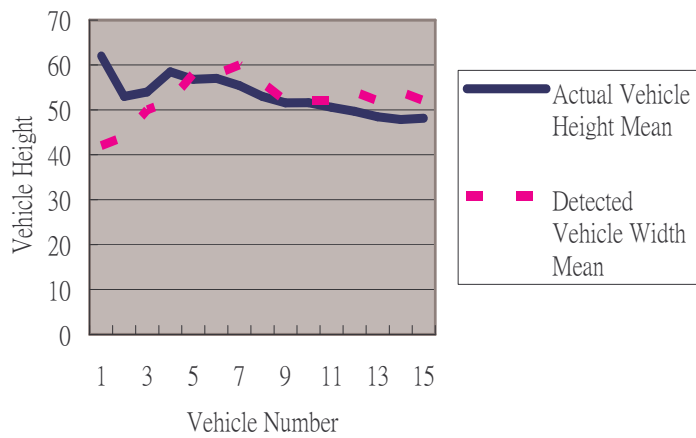


(b)

Fig. 5-9 The histogram of vehicle (a) width mean, and (b) height mean with respect to the frame number



(a)



(b)

Fig. 5-10 The histogram of vehicle (a) width mean, (b) and height mean with respect to the vehicle number

## 5.4 Results of Queue Resolution

During the tracking processing, vehicles may wait at the traffic lights and at traffic jam so that the stopped vehicles will interrupt the tracking processing. Consequently, the proposed queue detection and resolution are posed to detect and split occluded vehicle. There are some examples shown in Fig. 5-11. In Fig. 5-11, the first vehicle in the first or second lane is split successfully but deleted by tracking reasoning in that it connected to the boundary. Therefore, although the proposed queue detection and resolution techniques applied to stopped-vehicles are based on the edge property of vehicle which is influenced easily by illumination, the high adaptation of authors' system will decrease the effect of the weather condition.







Fig. 5-11 Stopped vehicles are detected and split by the proposed methods of queue detection and resolution.

## 5.5 On-line Demonstration

In our experiments, the size of each image is  $320 \times 240$  and the frame rate of the sequence is 30 fps. And the proposed system is developed on Windows XP platform with a Pentium-4 2.8 GHz CPU, 512M RAM. Therefore, the average processing time is 18 ms per frame. The processing time is faster than capture time so that we can safely arrive that the proposed system can process real-time.

In order to easy setup and maintain, a client and server architecture is utilized for the proposed system to retrieve remote MPEG-4 compressed video captured from a CCD camera. As the bandwidth of Internet is limited, a test of how bitrate of a compressed video affects the proposed system should be performed.

Fig. 5-12 shows that the accuracy of detection doesn't decrease a lot even the birate of video is small. The sample shown in Fig. 5-12(b) and (c) display that the quality of images is more important if the image sequence is with low frame rate. However, according to Fig. 5-12(a), the accuracy at bitrate 128Kbps and 256Kbps is larger than it at 256Kbps and

512Kbps. Because the detected amount will highly affected by noise object, some vehicles are not detected while the quality of images is well, but when the quality of images is bad, lots of noise would appear to be detected and considered as vehicles possibly. Conceivably, there are also some noises in the image of high quality but will be filtered out when the quality of image is not well. Our system can detect vehicles in high accuracy with the bitrate higher than 128 Kbps. Therefore, although the frame rate of input video is 15 fps and 10 fps, the accuracy of vehicle detection is precise as well.

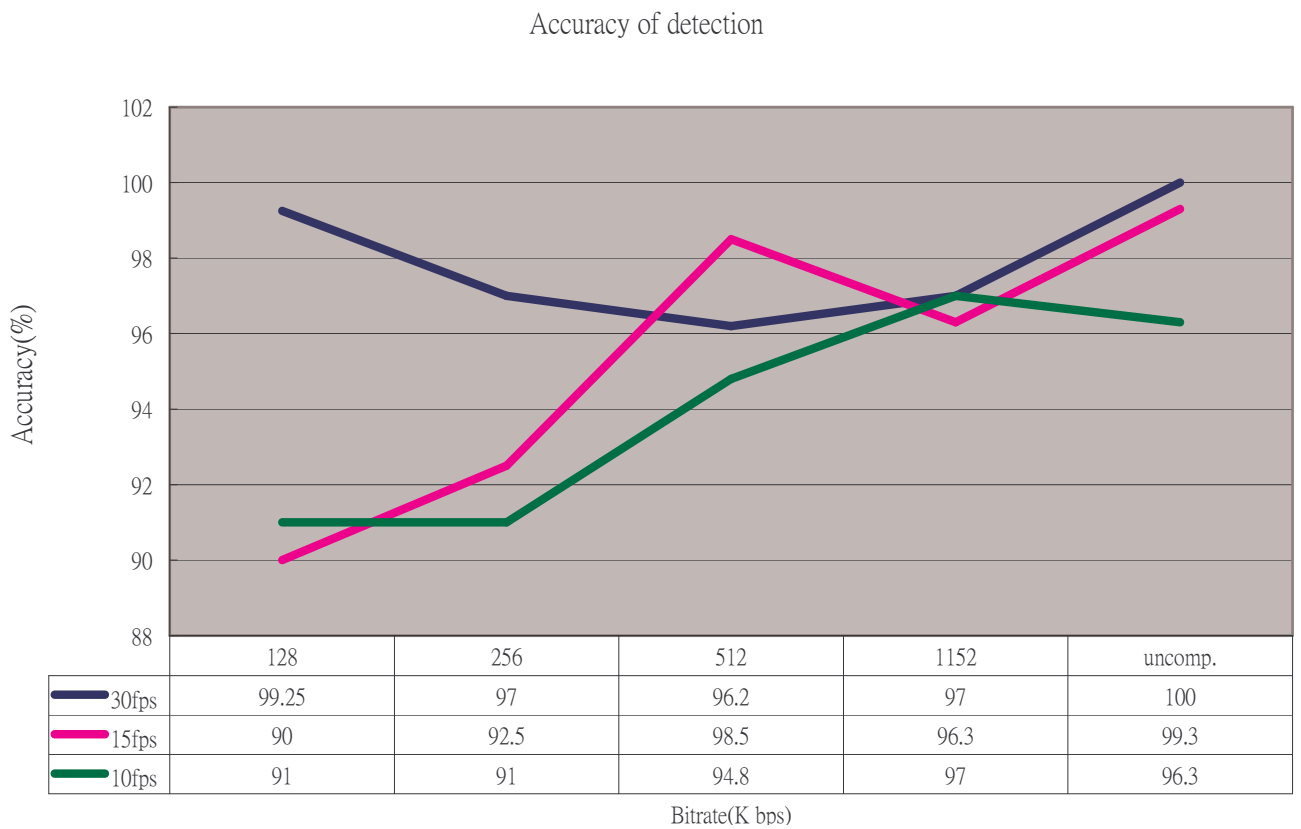


Fig. 5-12 The accuracy of detection of compressed images with bitrate 128Kbps to 1152Kbps with frame rate 30 fps, 15 fps, and 10 fps.

# Chapter 6 Conclusions and Future

## Work

### 6.1 Conclusion

This study presents an MVDT system with dynamic segmentation, adaptive parametric evaluation, vehicle detection, prior splitting based on lane information, vehicle tracking, queue detection and resolution, and comprehensive traffic parameter calculation. Initially, spatiotemporal statistics-based color background extraction approach with luminance adaptation and incorrect convergence compensation is utilized to segment moving objects robustly. Next, prior splitting based on lane information is exploited using automatic straight lane detection technique to resolve occluded vehicles have just entered the detection zone. Therefore, the stopped vehicles caused by the traffic light or the traffic jam would be connected to the front vehicle so that they can't be tracked successfully by proposed tracking processing. Moreover, in order to keeping the tracking reasoning at queue time, the queue detection and resolution are applied to split the stopped vehicles. Finally, traffic parameters based on tracked trajectories are calculated to improve traffic monitoring.

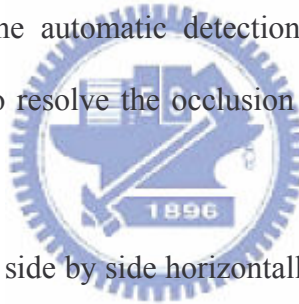
Experimental results indicate that developed system can operate in real-time with high accuracy. The precision and reliability of the proposed system is better because a lane mask is utilized to help to resolve vehicle occlusion, under real-time conditions. Furthermore, the accuracy of tracking at queue time increases due to the queue resolution. According to the traffic situation in Taiwan, motorcycles are also detected and tracked by the proposed MVDT system accurately. Due to the parameter adaptation, the developed system can be setup without the need for any information about the environment in advance, except the lane mask.

## 6.2 Future Work

Recently, an increasing number of MVDT applications have been applied to traffic intersection for monitoring traffic due to its convenience for set-up and low cost. This MVDT system built on PC-based can be set up on embedded system with portable size and lower cost. But embedded systems contain low computing power and fewer memories so that the PC-based application can't be ported without reducing code size and computing complication.

In this thesis, the proposed background extraction is based on the probability of appearance so that it will be failed due to traffic jam. Therefore, the other algorithm special for background extraction of traffic jam is necessary to increase the robustness of the system.

Furthermore, the straight lane automatic detection is used to resolve the occlusion. However, this processing can also resolve the occlusion on the curve lane if the curve lane automatic detection is applied.



Otherwise, vehicles occluded side by side horizontally in the top-bottom direction can be split by lane information; however, occluded vehicles in the left-right direction can't be separated using lane information in that the vehicles always across the lane visually in the image. So, the proposed system can be extended by occlusion detection and resolution for the left-right direction.

## References:

- [1] T. Bücher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W. V. Seelen, "Image processing and behavior planning for intelligent vehicles," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 62-75, Feb. 2003.
- [2] S. -C. Chen, M. -L. Shyu, S. Peeta, and C. Zhang, "Learning-based spatio-temporal vehicle tracking and indexing for transportation multimedia database systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 154-167, vol. 4, no. 3, Sep. 2003.
- [3] C. -J. Chen, C. -C. Chiu, B. -F. Wu, S. -P. Lin and C. -D. Huang, "The Moving Object Segmentation Approach to Vehicle Extraction," in *Proc. IEEE International Conference on Networking, Sensing and Control, ICNSC2004*, pp.19~23, March 21-23, 2004.
- [4] B. Coifman, D. Beymer, P. McLauchlan, J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Elsevier, Transportation Research Part C*, vol. 6, no. 2, pp. 271-288.
- [5] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 119-130, Jun. 2000.
- [6] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37-47, Mar. 2002.

- [7] D. M. Ha, J. M. Lee, Y. D. Kim, "Neural-edge-based vehicle detection and traffic parameter extraction," *Elsevier, Image and Vision Computing*, vol. 22, no. 11, pp. 889-907.
- [8] J. Kato, T. Watanabe, S. Joga, Y. Liu, and H. Hase, "An HMM/MRF-based stochastic framework for robust vehicle tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 3, pp. 142-154, Sep. 2004.
- [9] S. Kamijo, Y. Matsushita, K. Ikeuchi, M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108-118, Jun. 2000.
- [10] V. Khanna, P. Gupta, and C. J. Hwang, "Finding connected components in digital images", *International Conference on Information Technology*, pp. 652-656, Apr. 2001
- [11] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *Proc. Third European Conference on Computer Vision*, pp. 186-196, Springer-Verlag, 1997.
- [12] J. Lou, T. Tan and W. Hu, "Visual vehicle tracking algorithm," *IEEE Electronics Letters*, vol. 38, no. 18, pp. 1024-1025, Aug. 2002.
- [13] D. W. Lim, S. H. Choi, J. S. Jun, "Automated detection of all kinds of violations at a street intersection using real time individual vehicle tracking," in *Proc. 5<sup>th</sup> IEEE International Conference on Image Analysis and Interpretation*, pp. 126-129, Apr. 2002.
- [14] A. H. S. Lai and N. H. C. Yung, "Vehicle-type identification through automated virtual loop assignment and block-based direction-biased motion estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 86-97, Jun. 2000.



- [15]B. Li, R. Chellappa, "A generic approach to simultaneous tracking and verification in video," *IEEE Transactions on Image Processing*, vol. 11, no. 5, pp. 530-544, May 2002.
- [16]F. M., S. M. Y., "Real-time image processing approach to measure traffic queue parameters" in *Proc. Vision, Image and Signal Processing, IEE* vol. 142, pp. 297-303, Oct. 1995
- [17]Z. M., M. S.; M. C. M, "An efficient vehicle queue detection system based on image processing" in *Proc. 12th International Conference on Image Analysis and Processing*, pp.232 – 237, 17-19 Sept. 2003
- [18]C. E. Smith, S. A. Brandt, and N. P. Papanikolopoulos, "Visual tracking for intelligent vehicle-highway systems," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 4, pp. 744-759, Nov. 1996.
- [19]J. Suzuki, K. Horiba, and N. Sugie, "Fast connected-component labeling based on sequential local operations in the course of forward raster scan followed by backward raster scan," *Pattern Recognition*, vol. 2, pp. 434-437, Sep. 2000.
- [20]H. Tao, H. S. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 75-89, Jan. 2002.
- [21]H. Veeraraghavan, O. Masoud, and N. P. PaPanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Transactions on Intelligent Transportation Systems*, pp. 78-89 vol. 4, no. 2, Jun. 2003.
- [22]B. –F. Wu, S. –P. Lin, Y. –H. Chen, "A Real-Time Multiple-Vehicle Detection and Tracking System with Prior Occlusion Detection and Resolution" in *Proc. 5<sup>th</sup> IEEE International Conference on Signal Processing and Information Technology*, Dec. 2005, pp. 311-316.