



國立勤益科技大學  
電機工程系電機研究所碩士班

碩士論文

以視覺為基礎之全日嵌入式車輛偵測系統

**Vision-Based All-Day Vehicle Detection**

**Using Embedded System**

研究生：陳璿文

指導教授：郭英哲 博士

中華民國九十九年七月

以視覺為基礎之全日嵌入式車輛偵測系統

Vision-Based All-Day Vehicle Detection Using Embedded System

研究生：陳璿文  
指導教授：郭英哲 博士



國立勤益科技大學

電機系碩士班

碩士論文

A Thesis

Submitted to

Institute of Industrial Design

National Chin-Yi University of Technology

in Partial Fulfillment of the Requirements

for the Degree of

Master of Design

In

Electrical Engineering

July 2010

Taiping, Taichung, Taiwan, Republic of China

中華民國九十九年七月

國立勤益科技大學  
研究所碩士班  
論文口試委員會審定書

本校電機工程系碩士班 陳璿文 君

所提論文 以視覺為基礎之全日嵌入式車輛偵測系統

合於碩士資格水準，業經本委員會評審認可。

論文口試委員會：

召集人：郭吳哲

委員：林俊奇

郭吳哲

指導教授：郭吳哲

主

任：郭吳哲

中華民國九十九年七月十六號

## 摘要

根據研究顯示，90%以上的交通事故涉及人為失誤，且有 75%的交通事故，人為失誤是唯一的肇事原因。因此，為了減少人為失誤帶來的社會成本，許多國家已開始發展先進駕駛者輔助系統(Advanced Driver Assistance Systems, ADAS)來減少交通事故的發生。其中，以電腦視覺為基礎的獨立式駕駛輔助功能已被視為一個重點的發展方向。

本文所提出的全日車輛影像偵測系統主要應用在駕駛輔助功能中的前方車輛追撞預警(Forward Collision Warning, FCW)，使用一個具 USB 介面之 CMOS 感測器，並以軟體實現影像處理演算法。其中，考量了車內的空間、攜帶性以及商品化的程度，本文捨棄了系統的運算能力及即時性佳的個人電腦，改以嵌入式平台實作此系統。因此，在有限的硬體資源情況下，從動態的背景中找尋到前方車輛是本論文研究的主要工作。

在硬體及軟體部份，本文使用 Intel XScale PXA270 SOC-Based 嵌入式開發平台。所使用到的硬體周邊設備包含(1) USB 網路攝影機：影像擷取設備；(2)觸控式螢幕：使用者操作與顯示之介面；(3)音源輸出用喇叭：語音警告之輸出設備。本論文所使用之軟體包含 Embedded Linux 2.6.15.3 作業系統、網路攝影機的讀取與控制用之 Video for Linux Two(V4L2)函式庫、觸控式螢幕使用者圖型介面設計用之 MiniGUI 的函式庫、語音播放用之 madplay 應用程式。

全日車輛影像偵測演算法則是由 C 語言撰寫，其中包含以可拓理論為基礎的日間與夜間車輛偵測演算法的使用選擇、日間與夜間的前方車輛偵測、追蹤、距離估測、車道線偵測、危險情況時的語音警示與影像資訊記錄。其中，影像資訊記錄裡除了影像之外，也使用文字記載了當時的時間、日期與車輛之間的距離，並存入於可讀、寫的儲存空間中以利不幸發生事故時的調查與責任歸屬的釐清。

本論文所提出的系統主要實際測試於高速公路上，前方與旁側車道之車輛皆能成功的偵測、標定車輛在影像中的位置以及測得車輛之間的相對距離。其中，測距的範圍為五至三十公尺。由實測結果，本系統能成功的在日間與夜間中偵測到車輛，同車道車輛辨識率達 96% 以上，旁側車道車輛則達 92% 以上。日間情況下能排除地面標示文字、橋樑接縫、高架橋陰影與行道樹陰影等干擾。夜間情況中，能排除路燈、反光號誌、交通燈的干擾及明暗度不均等情況，並在各種照度下正常的偵測、追蹤車輛。



**關鍵字：**電腦視覺、先進駕駛者輔助系統、車輛偵測、車道偵測、嵌入式系統。

## Abstract

According to researches, over 90% of accidents are involved by human error, 75% of accidents are merely caused by human error. Therefore, in order to reduce the social cost induced by human error, many countries are developing Advanced Driver Assistance Systems (ADAS) to reduce the traffic accidents. Consequently, the vision-based independent driver assistance system is an important research topic.

The proposed All-day vehicle detection system is used in Forward Collision Warning (FCW) of ADAS, it is achieved by image processing algorithm based on computer vision technologies. The space of car, portability, and commercialization are decisive considerations when implementing the system. So, we use the embedded hardware platform instead of the personal computer (PC) though it has good abilities of computing and immediacy. The major function of our research is to find the preceding vehicles in the dynamic background under the limited hardware resources,.

In hardware, we use Intel XScale PXA270 SOC-based platform. The peripheral devices include: (1) USB Web Camera: the image capture device, (2) touch screen TFT-LCD display: graphic user interfaces and image display, (3) speaker: the speech warning output device. The software of the system contain Embedded Linux 2.6.15.3 operating system, Video for Linux Two open source code for driving Web Camera, the MiniGUI open source code that is utilized to design GUI in touch screen, and madplay open source code for processing voice signal.

All-day vehicle detection algorithm is implemented in C language. The functions of the vehicle detection algorithm include lane detection, vehicle detection, vehicle tracking, distance estimation, audio warning, and image information record. Audio alter

and image record will be acted when the distance between preceding vehicle and the test car is inside the safe range. Moreover, the record of image also includes text information of time, date and vehicle distance, and stored in non-volatile storage device for the investigation of accident.

In this thesis, the proposed system mainly tested on the highway, the preceding and passing vehicle are able to be detected correctly and the effective distance estimation range between vehicles is from 5 meters to 30 meters. The experimental results show that the proportion of correct identifications of preceding vehicles is above 96%, and which of passing vehicle is above 92%. Experimental results also indicate that the system correctly identifies vehicles in real time. In the daytime, the system can eliminate many the interferences, such as the text on the ground, the seam of bridge, the shadow of the overpass and shadows of trees. In the nighttime, the interferences of the street lamps, reflective signs, and traffic lights are also can be eliminated by the system.

**Keywords:** Computer Vision, Advanced Driver Assistance Systems, Vehicle Detection, Lane Detection, Embedded System.

## 誌謝

我必需先說完成這篇論文就像做夢一樣。能夠在碩士求學生涯中，完成本篇論文首先感謝指導教授郭英哲老師，在這兩年裡提供良好的研究環境給予我自由發揮與研究的空間，從老師身上學習到的不單是專業的知識，還有待人處事上所應該有的態度。在此，也同時感謝兩位口試委員暨南大學的郭耀文教授與本校電機所林俊成教授對本篇論文的指正與建議，使得本論文更接近於完善。

感謝林華峰學長帶領我進來嵌入式系統實驗室讓我在這良好的環境下成長，並給予我許多信心與勉勵。感謝李彥鋒學長耐心的教導、傳授我實作與理論上的相關技巧與知識，以及做人做事的態度，並且也常與我們分享許多有趣又特別的經驗讓大家開心。感謝蔡政育學長總是在無助與徬徨時鼓勵我們，給予大家繼續向前衝刺的動力。感謝同實驗室的大毛規劃的武嶺之旅讓我了解合歡山之美，而且一點都不累。感謝電機碩班熱愛打籃球的朋友們，讓我們一起拿到了冠軍。感謝電機碩班熱愛喝酒的朋友們，讓我了解到我的酒量還有很多進步的空間，也謝謝你們幫我清腸胃。感謝實驗室碩一學弟顏志鴻、張為彥有你們在實驗室總是充滿了快樂的氣氛。感謝實驗室學弟楊子儀一同陪我征戰各地，並隨時幫我注意許多細節給予我許多意見。感謝實驗室學弟林韋辰準備的蛋糕與耐操的機車讓我留下難忘的回憶。感謝實驗室學弟紹平總是熱心的為我們拍下許多照片。

最後，感謝進登、阿傑、冠宇、佑琳、正平、阿低、沅鴻、小方、大姊在我碩士生涯最低潮無助的時候鼓勵與幫助我。感謝佳珊這些年來的陪伴讓我在挫折時仍有動力的完成研究。特別感謝家人們不求回報的付出，讓我能無後顧之憂的完成學位。謹以此論文獻給所有我關心與關心我的人，謝謝你們大家，以上。

陳璿文 謹誌 中華民國九十九年七月二十七號



# 章節目錄

摘要 .....	I
Abstract .....	III
誌謝 .....	V
章節目錄 .....	VI
圖目錄 .....	VII
表目錄 .....	IX
第一章 緒論 .....	1
1.1 研究背景 .....	1
1.2 研究動機與目的 .....	6
1.3 系統功能 .....	7
1.4 系統架構與事前工作 .....	7
1.4.1 硬體架構 .....	7
1.4.2 軟體架構 .....	9
1.4.3 影像串流的處理 .....	10
1.4.4 系統開發與測試方式 .....	13
1.5 全日車輛偵測演算法架構 .....	15
1.6 論文貢獻 .....	16
1.7 論文架構 .....	16
第二章 相關研究 .....	18
2.1 日間車輛偵測之文獻探討 .....	18
2.2 夜間車輛偵測文獻探討 .....	23
2.3 車道線偵測文獻探討 .....	29
第三章 日間車輛偵測 .....	34
3.1 縮小搜尋區域與前處理 .....	35
3.2 前景擷取 .....	36
3.3 路面雜訊濾除 .....	38
3.4 標示車輛腳印 .....	41
3.4.1. 車道偵測與車道遮罩 .....	41
3.4.2. 車道判斷 .....	47
3.4.3. 車輛腳印標示 .....	47
3.5 車輛腳印驗證 .....	50
3.6 車輛追蹤程序 .....	52
第四章 夜間車輛偵測 .....	53

4.1	感興趣區域的定義與色彩空間之轉換.....	54
4.2	亮點物件的擷取.....	56
4.3	移除雜訊與強化特徵.....	60
4.4	亮點物件資訊處理.....	67
4.4.1	影像分割.....	67
4.4.2	資訊處理.....	71
4.5	框選車輛車尾燈.....	73
4.5.1	區分不同車道之亮點物件.....	74
4.5.2	車尾燈配對程序.....	76
4.6	驗證車輛影像.....	81
4.7	車輛追蹤程序.....	82
第五章	距離及碰撞時間估測.....	84
5.1	前方車輛距離估測.....	84
5.2	碰撞時間估測.....	86
第六章	基於可拓理論的日夜間整合.....	88
6.1	可拓理論簡介.....	88
6.1.1	物元基本理論.....	88
6.1.2	可拓集合理論.....	90
6.1.3	關聯函數.....	91
6.2	可拓日夜間辨識方法.....	94
第七章	實驗結果與分析.....	99
7.1	實驗結果.....	99
7.1.1	日間車輛偵測結果.....	99
7.1.2	夜間車輛偵測結果.....	102
7.1.3	車輛測距與碰撞時間估測結果.....	105
7.2	實驗結果分析.....	112
第八章	結論.....	115
8.1	結論.....	115
8.2	現存問題分析與未來研究方向.....	116
參考文獻	.....	118

## 圖目錄

圖 1.1	CMOS Camera 的應用層面與未來發展.....	5
圖 1.2	系統功能說明圖.....	7
圖 1.3	Create Xscale PXA270 嵌入式平台展示圖.....	8
圖 1.4	USB 網路攝影機(CMOS sensor) .....	9
圖 1.5	系統軟體架構圖.....	10
圖 1.6	影像擷取流程圖.....	11
圖 1.7	實驗室模擬測試系統架設圖.....	14
圖 1.8	實際道路測試之系統架設.....	15
圖 1.9	全日車輛偵測演算法架構.....	15
圖 2.1	車輛偵測之兩個步驟.....	18
圖 2.2	以 MC-MQDF 方法實現前方車輛偵測與識別之概念圖 .....	19
圖 2.3	車寬與道路寬度之比例範圍.....	20
圖 2.4	使用投影量分析找出正確的車輛左右邊界.....	21
圖 2.5	使用光流法偵測前方車輛影像.....	22
圖 2.6	運用車尾燈橫跨中線特性區分不同車道之車輛.....	23
圖 2.7	(a)夜間道路原始影像 (b)峰值分析之結果 (c)經由區隔車輛處理之結果 .....	24
圖 2.8	(a)虛擬地平線與明亮物件分割的結果 (b)成對車尾燈形狀與距離比例之 規則示意 .....	25
圖 2.9	(a)RGB 影像轉換至車尾燈特徵較為明顯之色彩空間 (b)車尾燈影像的擷取 .....	26
圖 2.10	車尾燈轉換共變異矩陣進行面積大小比對 (a)同一車輛之車尾燈 (b)不 同車輛之車尾燈 .....	27
圖 2.11	車尾燈光暈與實體示意圖.....	27
圖 2.12	連續影像中配對組合之車尾燈垂直移動距離.....	28
圖 2.13	車尾燈的範圍與重心標示.....	29
圖 2.14	車道線偵測前的影像前處理程序 (a)原始道路影像 (b)高斯模糊與影像 切割 (c)區域二值化.....	30
圖 2.15	車道線的峰與谷的例子.....	30
圖 2.16	水平掃描的分析圖例.....	31
圖 2.17	(a)3 X 9 的群組化用遮罩 (b)線段合併串連的示意圖 .....	32
圖 2.18	影像之區域劃分.....	33
圖 2.19	四十五度邊緣運算子 (a)左半部區域所用 (b)用於右半部區域.....	33

圖 3.1	日間車輛偵測演算法執行之程序.....	34
圖 3.2	ROI 範圍定義與 ROI 中 RGB 影像轉換至灰階影像.....	36
圖 3.3	Sobel 的水平與垂直濾波遮罩權重分配.....	37
圖 3.4	將 ROI 中灰階影像經由 Sobel 水平與垂直邊緣偵測後之影像.....	37
圖 3.5	圖 3.4ROI 中影像二值化後之影像.....	38
圖 3.6	ROI 影像中高亮度區域包含了路面與車道標示線.....	39
圖 3.7	對高亮度區域進行膨脹運算之結果.....	40
圖 3.8	用相減運算濾除路面雜訊.....	41
圖 3.9	Al與Ar區域之區分及邊緣偵測運算子掃瞄方向.....	42
圖 3.10	四十五度邊緣運算子(a) Al區域所用 (b) Ar區域用.....	42
圖 3.11	四十五度基礎之邊緣影像與偵測邊緣交會點方法.....	43
圖 3.12	利用正負邊緣交會之特性所產生的邊緣點影像.....	43
圖 3.13	偏移門檻值 $\Delta x$ 與 $\Delta y$ .....	44
圖 3.14	相同線段使用相同顏色表示.....	44
圖 3.15	目前行駛車道遮罩影像.....	45
圖 3.16	三個預設車道遮罩影像.....	46
圖 3.17	利用中央車道遮罩擷取車輛腳印.....	46
圖 3.18	中央車道車輛搜尋之範圍.....	48
圖 3.19	投影量統計示意圖 (a)統計 X 軸上投影量 (b)統計 Y 軸上投影量.....	48
圖 3.20	實際運作之方式.....	49
圖 3.21	(a)左右車道車輛搜尋方法 (b) 搜尋寬高比為 3:1 直角特徵.....	50
圖 3.22	中央與旁側車道的車輛腳印標示結果.....	50
圖 3.23	車輛影像對稱軸與邊界盒.....	51
圖 3.24	車輛追蹤與車輛偵測之關係.....	52
圖 4.1	夜間車輛偵測演算法執行之程序.....	53
圖 4.2	ROI 之定義範圍.....	55
圖 4.3	以 Y(亮度)為基礎之灰階影像.....	56
圖 4.4	以Cr(紅色色度)為基礎之灰階影像.....	56
圖 4.5	運用 k 值分出兩群組的灰階分佈示意圖.....	58
圖 4.6	經過二值化後影像灰階分佈情況.....	59
圖 4.7	對 Y 空間影像取門檻值二值化之結果.....	59
圖 4.8	對 Y 空間與Cr空間影像取門檻值二值化之結果.....	60
圖 4.9	正方形結構元素對原始影像進行膨脹運算的結果.....	61
圖 4.10	矩型結構元素對原始影像進行膨脹運算的結果.....	62
圖 4.11	正方形結構元素對原始影像進行侵蝕運算的結果.....	63

圖 4.12	矩型結構元素對原始影像進行侵蝕運算的結果.....	63
圖 4.13	幾何解釋斷開運算 (a)結構元素 B 在 A 裡移動, 並且不超出 A 的邊界 (b)粗線代表斷開的外邊界 (c)完整之斷開.....	64
圖 4.14	斷開運算的圖示說明 (a)集合 A 與結構元素 B (b)集合 A 被侵蝕之過程 (c)侵蝕運算完結果 (d)對已侵蝕完集合膨脹過程 (e)斷開運算之結果.....	66
圖 4.15	把圖 4.8 進行侵蝕運算之結果.....	66
圖 4.16	把圖 4.8 進行斷開運算之結果.....	67
圖 4.17	像素之間相鄰性之定義.....	68
圖 4.18	由左至右, 由上至下對影像掃瞄.....	68
圖 4.19	標記後對該標記點進行相鄰元素遞迴偵測.....	69
圖 4.20	各高亮度區塊標記結果.....	69
圖 4.21	(a)為第一次編號結果之示意圖 (b)等價表格.....	70
圖 4.22	亮點物件分割之結果.....	71
圖 4.23	投影分析找尋中心點之示意圖.....	72
圖 4.24	各個亮點物件中心點標示.....	73
圖 4.25	各個亮點物件的框選結果.....	73
圖 4.26	車道線偵測與中線區分法之關係.....	74
圖 4.27	中線區分法之示意圖.....	75
圖 4.28	實際影像中線區分之顯示.....	75
圖 4.29	AND 運算處理之結果.....	76
圖 4.30	亮點物件之資訊.....	77
圖 4.31	水平方向亮點物件之間關係圖.....	78
圖 4.32	垂直方向亮點物件之間關係圖.....	78
圖 4.33	兩亮點物件的高度比例關係圖.....	79
圖 4.34	亮點物件與車輛寬度之關係圖.....	80
圖 4.35	矩形邊界盒之寬高比例關係圖.....	80
圖 4.36	車輛車尾燈配對之結果.....	81
圖 4.37	(a)原始輸入影像 (b)影像中亮點物件之強度與大小相似 (c)誤判情況 (d)經由前一張影像修正後結果.....	82
圖 4.38	車輛追蹤之結果.....	83
圖 5.1	二維影像座標轉換至攝影機座標之示意圖.....	85
圖 5.2	攝影機座標轉車輛座標之示意圖.....	86
圖 6.1	點 x 在區間 X0 之外時距的關係.....	91
圖 6.2	點 x 在區間 X0 之內時距的關係.....	92
圖 6.3	點 x 存在於 b 與 d 區間的位值.....	92

圖 6.4	點 $x$ 存在於 $c$ 與 $a$ 區間的位值.....	93
圖 6.5	可拓關聯函數.....	93
圖 6.6	天空與道路亮度灰階值的取樣區域 (a)夜間 (b)日間.....	94
圖 6.7	各一萬筆日間與夜間天空灰階值的統計結果.....	95
圖 6.8	各一萬筆日間與夜間道路灰階值的統計結果.....	95
圖 6.9	可拓日夜間識別流程圖.....	98
圖 7.1	中央車道車輛偵測結果 (a)只有一台車輛行駛於同車道 (b)三線車道皆有車輛 (c)右側車道無車輛 (d)三線車道皆有車輛 (e)只有右側車道有車輛 (f)右側車道無車輛.....	100
圖 7.2	行駛於內側車道車輛偵測結果 (a) (b) (c)兩線車道上皆有車道 (d)車輛行駛於顏色不均的道路.....	101
圖 7.3	行駛於外側車道車輛偵測結果 (a)(b)兩線道皆有車輛情況.....	102
圖 7.4	行駛於光源強度較強環境 (a)(b)(c)左側與同車道有車輛情況 (d)左側來車頭燈照射於前方車尾之情況.....	103
圖 7.5	行駛於光源強度中等環境 (a)(b)右側與同車道有車輛情況 (c)旁側無車輛狀況 (d)左側車道行駛兩輛車輛與同車道有車輛之情況.....	104
圖 7.6	行駛於光源強度較差環境 (a)兩側車道無車輛狀況 (b)左側無車輛狀況 (c)左側無車輛狀況(d)只有右側有車輛情況.....	105
圖 7.7	(a)至(c)為日間情況時前方車輛距離與碰撞時間的估測結果.....	106
圖 7.8	(a)至(c)為夜間情況時前方車輛距離與碰撞時間的估測結果.....	107
圖 7.9	靜態測距驗正與校正結果 (a)十公尺 (b)二十公尺 (c)三十公尺.....	108
圖 7.10	(a)~(f)為前方車輛危險超車序列影像.....	110
圖 7.11	(a)~(c)系統在危險情況下紀錄的影像與資訊.....	111

## 表目錄

表 1	獨立式駕駛輔助系統感測器的發展方向.....	4
表 2	Creator PXA270 嵌入式平台系統規格表.....	8
表 3	車輛偵測、追蹤與影像擷取之處理時間統計.....	112
表 4	車輛偵測正確率統計.....	113



# 第一章 緒論

## 1.1 研究背景

根據研究顯示[1]，90%以上的交通事故涉及人為失誤，且有 75%的交通事故，人為失誤是唯一的肇事原因；另外，以時速 50 公里行駛時，如能提早 0.5 秒煞車，撞擊能量會減少一半。以台灣高速公路的交通事故統計為例[2]，事故發生原因屬車輛駕駛人因素者共計 835 件(占 92.2%)，其中以「未保持行車安全距離」195 件(占 21.5%)最多。交通事故造成人員傷亡與交通壅塞，會造成極大的經濟損失。除此之外，交通壅塞也會導致能源浪費與廢氣排放增加，惡化地球的空氣品質與環境。因此，發展智慧型車輛(Intelligent Vehicle, IV)中的駕駛輔助系統(Driver Assistance Systems, DAS)是為了防範事故並降低事故對乘員與車外人員傷害的主要目的。

智慧型車輛配備的駕駛者輔助系統能夠減少人為失誤所帶來的社會成本，其潛在利益龐大，國內外許多機構持續資助智慧型車輛與駕駛者輔助系統的研發工作，促進學術界與各產業界業者進行研究。尤其地面運輸領域的研究都特別注重駕駛者輔助系統；且許多技術已經商業化，成為市售的中高階車種的配備。

文獻[1]提到一開始的智慧型車輛發展並不是如此的順遂，像是歐洲汽車產業很早就開始進行智慧型車輛的開發，其中，電子式車身穩定控制(Electronic Stability Program, ESP)問世十年在德國也只有 40%的滲透率，主要原因在於早期消費者對此的認知不足；另一原因是其外部效益無法內化到消費者的利益之中，使的商業化之路如此漫長。之後，歐盟使用政策力推動各家廠商與相關利益者加速發展智慧型車輛系統，其中 1987 至 1995 年由歐洲委員會資助進行的 EUREKA Prometheus Project 是一項無人駕駛自動車的實驗計畫，這項計畫執行之後的結果更加確定了發展先進駕駛者輔助系統(Advanced Driver Assistance Systems, ADAS)能減少交通



事故的可行性。

至今，智慧型車輛仍處於發展階段，尚未開發完成。目前的汽車與預期計畫的智慧型車輛仍有一段不小的距離，這不單是技術的差距，還包括了業者如何在制訂規格時考慮一致性或相容性才能達到最大利益、各國政府政策是否能協調相關產業利益關係與法律爭議及個人隱私等問題，最重要的仍是消費者能否接受智慧型車輛的使用模式及價格。目前，美、日及歐盟正積極發展此領域，並制定相關政策。如此，才能使智慧型車輛帶來更多的商機。

智慧型車輛的駕駛者輔助系統中，主要分為獨立式與互動式兩種類型。其中，獨立式駕駛輔助系統包含：

(1) 適應性巡航控制(Adaptive Cruise Control, ACC)

在車輛保持一定速度行駛時，此系統可以與前車保持駕駛人所設定的安全距離。並用裝載於車頭的感測器偵測前車的距離，隨著前車速度的變化，系統自動控制駕駛者車輛的油門與煞車以保持安全距離。

(2) 停車後再起步(Stop & Go)

顧名思義，此功能主要用於走走停停頻繁與短距離的情況，例如：塞車，在車頭安裝短距離感測器，偵測前保險桿四周的車輛和行人，再搭配 ACC 自動的控制油門與煞車。

(3) 車道偏離預警(Lane Departure Warning, LDW)

當駕駛者因外在因素分心，未注意前方路況，導致車輛非有意偏離車道。該系統利用車頭或車尾之感測器辨識車道線，並在非有意偏離車道時，系統會藉由聲光或震動方向盤等方式警示駕駛者。

(4) 碰撞警示(Collision Warning)

駕駛者無專心於前方路況，或是突然從旁側出現行人或障礙物而使駕駛人未及反應時。系統利用裝載於車輛上的前向感測器偵測短距離目標，

計算目標的距離與相對速度，並在危險情況時警示駕駛者前方可能的撞擊意外。

(5) 預警式碰撞(Pre-Crash)

當系統藉由車輛上裝載的前向感測器判斷前方的撞擊無法避免時，發出警示並將車輛上的保護設施調整到對駕駛最安全的狀態(例如：安全帶緊縮、煞車力道控制等)。

(6) 盲點偵測(Blind Spot Detection, BSD)

車輛欲準備變換車道前，透過兩側後視鏡裝載的後向式感測器，偵測到駕駛者視角盲點處有無車輛。並在有可能會產生碰撞的情況下發生警示。

(7) 瞌睡偵測警告(Drowsy Detection Warning)

裝載一面向駕駛者臉部的感測器，當系統判斷駕駛者有倦意(眨眼、頭部晃動的頻率)時發出聲光或震動方向盤等警示。

(8) 夜間視覺系統(Night Vision System, NVS)

由於夜間行駛時能見視野較差，因此使用遠近紅外線增強夜間視野，減少碰撞發生。

由於，互動式駕駛輔助系統需要車輛本身新技術整合外，也必須配合道路硬體建設，各家車輛生產公司之間的無線通訊協定等，牽涉的因素較為廣泛，必須要長期發展才有成果，因此實現難度較高，導致商業化程度不如獨立式駕駛者輔助系統。

獨立式駕駛輔助的感測器主要由攝影機、雷達、雷射、遠/近紅外線感測器組成，其主要應用於 ACC、LDW、BSD、CW、NVS。表 1[1]顯示獨立駕駛者輔助系統感測器的發展方向。當中也不難發現許多功能也透過感測器之間互相配合的概念使其更加強健。

表1 獨立式駕駛輔助系統感測器的發展方向[1]

獨立式駕駛輔助系統	感測器的要求	符合要求的感測器	感測器的發展方向
ACC	長距離追蹤	77GHz Radar	降低成本
		Lidar	降低天候干擾、縮小體積
Stop & Go	短距離追蹤、高解析度	24GHz Radar	24GHz 與其他產業共用，長期可開發其他頻率
BSD	短距離偵測、高解析度、體積小	24GHz Radar	同前述
		CMOS Camera	提高動態範圍、降低 CPU 耗電量、使用 Mono Camera 可降低成本
LDW	車道辨識	CMOS Camera	提高動態範圍、降低 CPU 耗電量、辨識交通號誌、物體分類
		IR Sensor	與 CMOS Camera 共用
CW	短距離追蹤、高解析度	24GHz Radar	同前述
		CMOS Camera	提高動態範圍、降低 CPU 耗電量
NVS	提供無照明、或夜間車外影像	Near IR Sensor	與 CMOS Camera 共用
		Far IR Sensor	

目前，在適應性巡航中，主要是使用 Radar 或 Lidar 作為感測器，偵測與前方目標的距離。Radar 多半使用 77GHz，測距的範圍在 2 至 120 公尺，解析度約為 1 公尺頻寬為 0.5GHz，適合用來追蹤車輛前方目標，但其解析度較低，針對短距離

追蹤目標效果較差，其單價偏高多半裝載於高單價位車種應用，普及的程度有限。Lidar 成本約只有 Radar 的一半，其功能與 Radar 大至相同，較 Radar 優點為成本低、掃描角度與解析度高；缺點為體積大、容易受水氣、濕度及目標表面髒汙影響。另外，除了上述兩種感測器，CMOS Camera 則是日漸崛起的一種，特性為輕巧、安裝便利、價格便宜，但容易受到光源的影響。發展 CMOS Camera 的關鍵主要在於軟體演算法的強健度。CMOS Camera 依製程可分為 CCD、CMOS，即便 CCD 的影像品質較高，但應用於車用電子領域對此性質較無嚴苛要求，反倒是 CMOS 的成本較低與體積較小適用於車輛，且未來延伸性也較 Radar 與 Lidar 高。拓璞產業研究所(TRI)在 2008 年十二月的產業專題報告[1]中也提到未來 CMOS Camera 的發展在車用電子領域中是不可或缺與忽視的，這也是本論文使用 CMOS Camera 作為感測器用於偵測車輛的主要原因。圖 1.1 顯示 CMOS Camera 的應用層面及遠景。

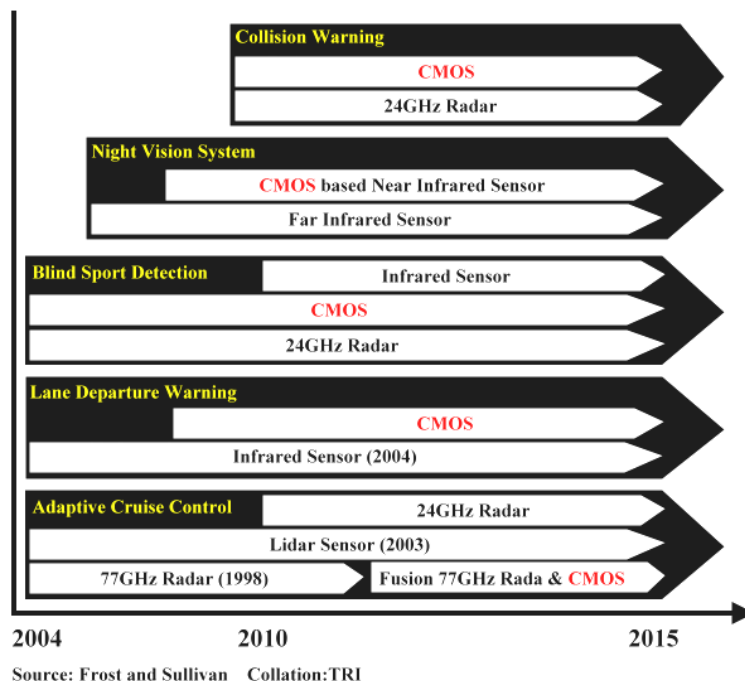


圖 1.1 CMOS Camera 的應用層面與未來發展 [1]

## 1.2 研究動機與目的

由於機器視覺的技術日漸成熟，因此以影像為基礎的偵測與識別的研究領域(人臉、指紋...等)也受到眾多學者開發、參與，其中車輛偵測領域在近幾年逐漸受到重視，主要原因在於：

- (1) 製程技術的改良使得光感測元件(例：攝影機)的功能性提高、價格降低、體積縮小，且車輛偵測所需影像品質要求不高，讓光感測元件在車輛偵測的應用上有足夠的應付能力。
- (2) 智慧型車輛的崛起，帶動了各種車用電子商品的成長。其中，以影像為基礎的車輛偵測的應用與延伸，更是智慧型車輛眾多功能之中不可缺少的單元。
- (3) 機器視覺領域中技術的累積，讓許多問題與瓶頸迎刃而解。
- (4) 隨著科技的進步，處理器的運算效能日益增加，以軟體演算法為主要因素的機器視覺領域有更大的發揮空間。

以影像為基礎的車輛偵測研究領域中，實現效能的好壞多半來自於軟體演算法，考慮到軟體演算法的執行速度、即時性與難易度，大多數的研究使用系統資源較充足的 PC-Based 架構實現。但安裝在空間有限的車輛中，PC 龐大的體積是會佔去車內許多空間，造成車內乘客的活動不便。因此，本文使用嵌入式系統實現此功能，其嵌入式系統定義為：『針對特定功能而設計的系統』，此點為嵌入式系統的主要優勢，它可為了特定的功能選擇系統的架構與資源，有效的減少不需要的週邊設備以降低成本，重點是能減少所佔用的空間與提升系統的可攜性。

本論文主要的研究目的，是在硬體資源有限的情況下，設計一個有效率與合適的影像處理演算法並能實現在嵌入式系統上，讓此系統具有準確、即時與輕便

的車輛偵測功能。

### 1.3 系統功能

本論文車輛偵測實現於嵌入式系統上，並可偵測在日間與夜間中行駛於高速公路的前方車輛。將一攝影機裝設於擋風玻璃內側，往駕駛者車輛車頭方向拍攝，擷取進來的影像經由一連串的影像處理程序找出前方車輛在影像中的位置，並藉由車輛底部的座標位置估算出攝影機與前方車輛的實際距離。當前方車輛行駛於危險距離中，系統會利用語音告警駕駛者保持適當距離，並且連續的紀錄危險情況時的影像、時間、日期與距離資訊於儲存空間中。日間與夜間情況下皆能顯示前方三線道車輛與同車道的前方車輛距離，在日間環境中系統能顯示車道的行駛資訊。圖 1.2 為系統功能的示意圖

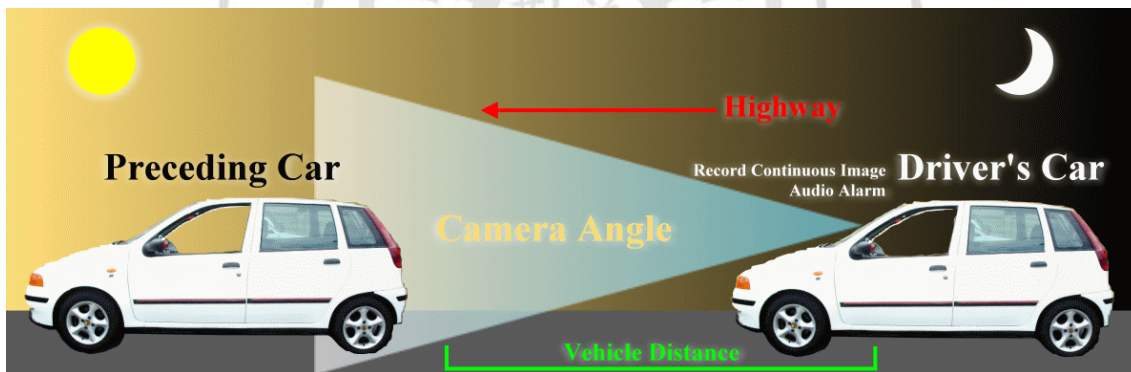


圖1.2 系統功能說明圖

### 1.4 系統架構與事前工作

#### 1.4.1 硬體架構

本論文以新華電腦公司的 Create XScale PXA270 作為硬體開發平台，如圖 1.3 所示，此為 ARM 系列 Intel XScale PXA270 型號之 SoC 處理器為設計基礎，其系

統規格表如表 2。外接裝置包含一 Logitech QuickCam E3500 (CMOS)網路攝影機、TFT-LCD 觸控式螢幕與一對外接式喇叭。QuickCam E3500 (圖 1.4)網路攝影機所能擷取的最高畫質為 VGA (640 x 480 像素)，在本系統中僅提供基本的影像擷取功能。觸控式螢幕提供圖形操作介面與即時影像的呈現，而外接式喇叭則模擬車用音響，用於語音警告的輸出功能。

表2 Creator PXA270 嵌入式平台系統規格表

<b>SOM Board</b>
CPU: Intel XScale PXA270 520MHZ Memory: Flash 32M Bytes, SDRAM 64M Bytes
<b>Based Board</b>
USB Host and Device Port Audio Interface(AC97) TFT LCD UART Port

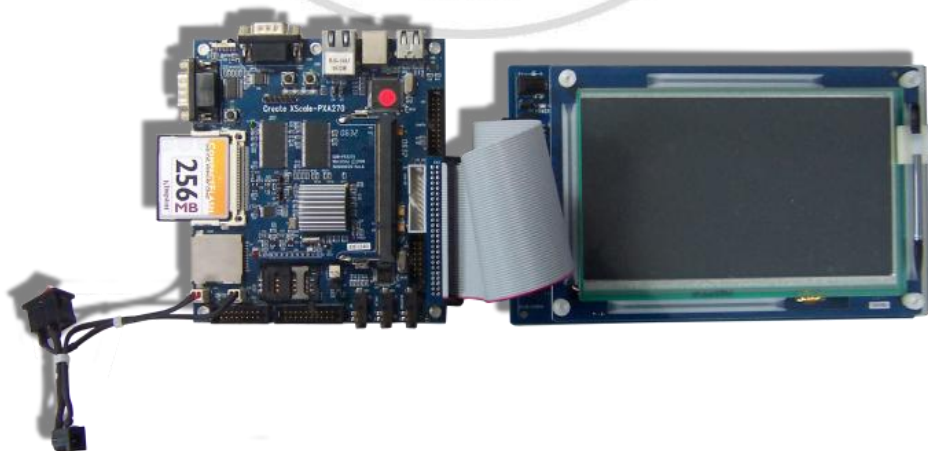


圖1.3 Create Xscale PXA270 嵌入式平台展示圖





圖1.4 USB 網路攝影機(CMOS sensor)

### 1.4.2 軟體架構

本論文軟體架構如圖 1.5 所示。系統藉由 USB Video Class Driver 驅動 Webcam，並使用 Video for Linux Two API 所提供的 I/O 控制函數擷取影像串流，並對影像串流進行處理並儲存於記憶體區塊中，其中 Video for Linux Two (V4L2)提供的 API 擷取與處理影像串流的細節將在下一個小節做詳細介紹。再利用 MiniGUI 所提供的圖型函數與使用者視窗元件，將 Webcam 所擷取的影像透過 Embedded Linux 中的驅動程式顯示在觸控式螢幕上。上層的 Frame Work (MiniGUI Application)是由 MiniGUI、Video for Linux Two API 所提供的函式所構成，其中包含了日、夜間的車輛偵測演算法、日夜間決策方法與依據本論文應用功能所建立的使用者介面。語音輸出的部份本文將 Madpaly 移植至嵌入式平台並由背景執行、播放，把 MP3 語音檔經過相關的函式庫(Z-lib, MAD-lib...)與 AC97 Driver 輸出至外接喇叭。本系統功能由標準的 C 語言撰寫，使用 ANSI C Library，它提供了基本的 I/O 函數。本平台作業系統使用 Embedded Linux 2.6.14.3，它是專為嵌入式平台所設計的精簡版 Linux。系統設計者可以在作業系統上多工執行多個視窗或應用程式，建立多元



化的服務系統。

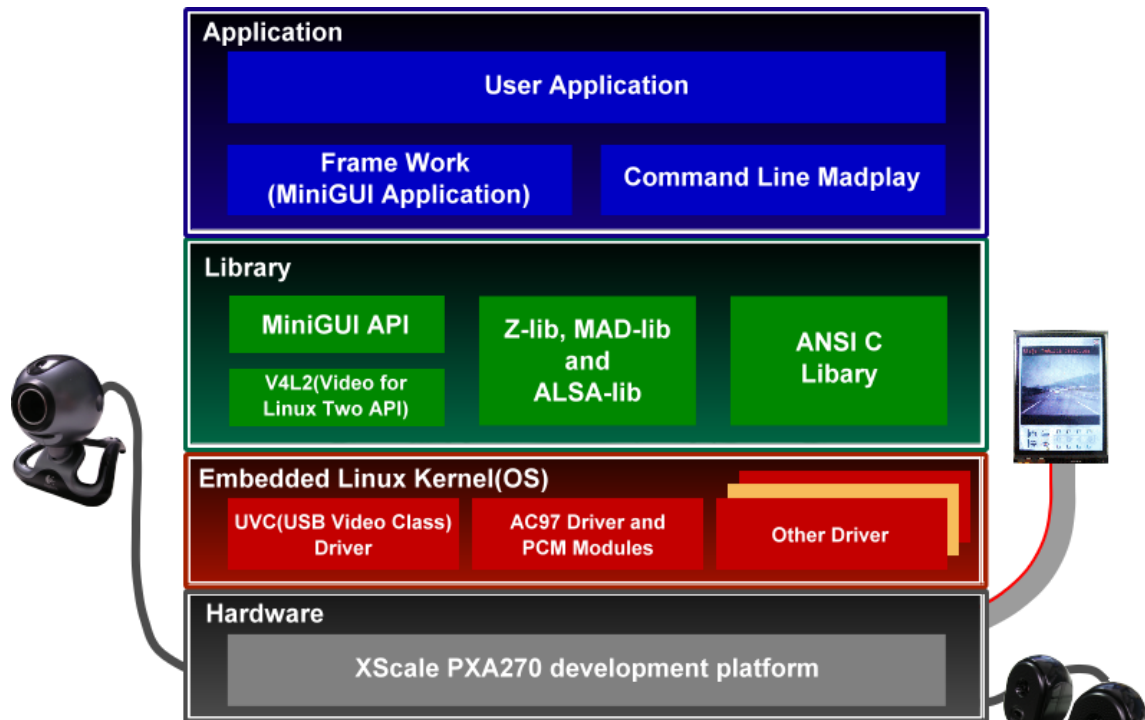


圖1.5 系統軟體架構圖

### 1.4.3 影像串流的處理

攝影機經由 USB Video Class 驅動之後，V4L2 提供 API 讓使用者能進行影像串流的擷取與攝影機的 I/O 控制，圖 1.6 則顯示了整個影像串流擷取的流程。以下分為數個步驟介紹從攝影機的初始設定與影像串流擷取的過程。

#### (1) 開啟影像擷取設備(Open Video Device)

在 Linux 作業系統中每一裝置都視為一個檔案，當核心啟動偵測到影像擷取設備，核心會把影像擷取設備代碼儲存，而該設備所對應檔案通常存在於/dev/video 中。本文使用 stat()這函式取得檔案的狀態，判斷是否

有對應其設備，再利用 `open()` 開啟影像擷取的設備。

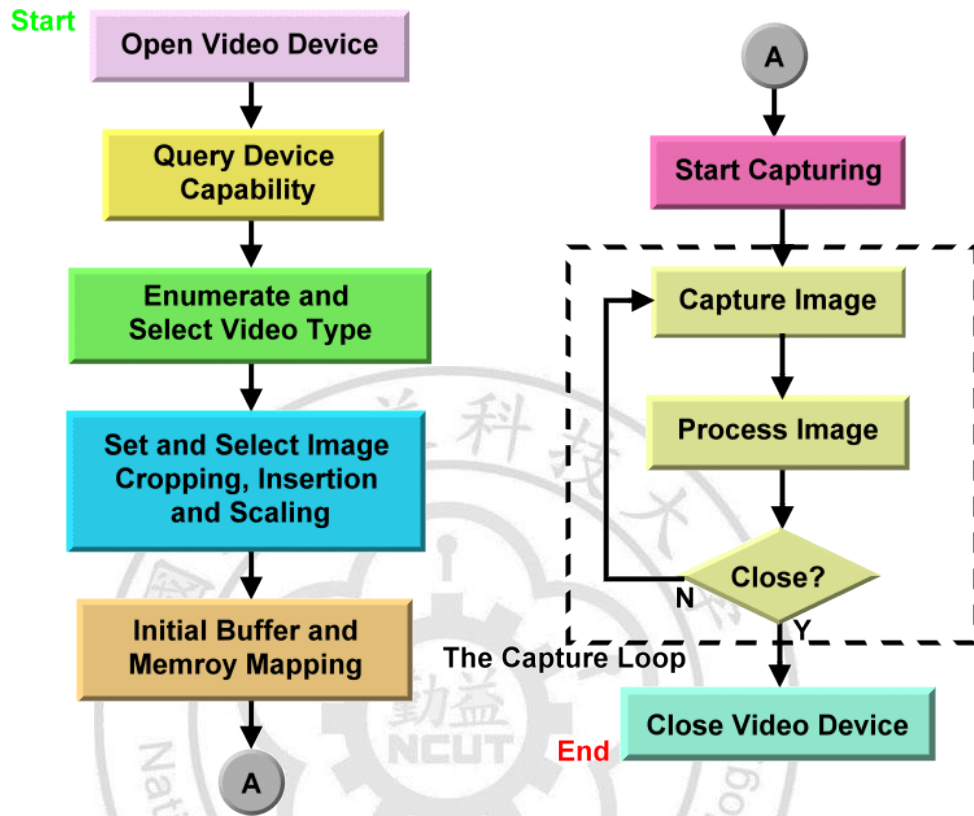


圖1.6 影像擷取流程圖

(2) 詢問設備能力(Query Device Capability)

使用 `VIDIOC_QUERYCAP` 這個 `ioctl()` 函式取得設備所具有的能力，並判斷該設備有無影像擷取與影像串流處理的功能。V4L2 中定義 `V4L2_CAP_VIDEO_CAPTURE` 為擷取影像功能，`V4L2_CAP_STREAMING` 為影像串流處理的能力。

(3) 選擇影像的型態(Enumerate and Select Video Type)

當設備的能力與功能都確認完之後，則將影像擷取與緩衝區的資料型態設定成影像串流用。V4L2 中定義

V4L2\_BUF\_TYPE\_VIDEO\_CAPTURE 為影像串流型態。

- (4) 設定與選擇影像擷取的大小、範圍與格式(Set and Select Image Cropping, Insertion, and Scaling)

接著選擇所要擷取影像的位置、大小與解析度，並選擇影像格式。其中，影像格式是使用 Motion JPEG。再利用 VIDIOC\_S\_FMT 這個 ioctl() 函式將上述的資訊進行設定。

- (5) 初始緩衝區與記憶體映射式(Initial Buffer and Memory Mapping)

V4L2 中提供兩種影像擷取的方法，分別是 Read/Write Method 與 Streaming I/O Method。Read/Write Method 是利用 read()和 write()這兩個函式重複的將驅動空間(Driver Space)的影像資料搬移至使用者空間(User Space)，如此動作將會浪費許多時間在影像資料的移動，因此效率較差。Streaming I/O Method 則是為了資料轉移的方便，先在驅動與使用者空間之間設置數個緩衝區，再用 mmap()函式將緩衝區映射至使用者空間，直接讀取影像的資料，這樣一來可以省去許多影像資料移動的時間，藉此提升影像擷取的效率。因此本文選用 Streaming I/O Method 作為本文取得影像資料的方法，使用 VIDIOC\_REQBUFS 這個 ioctl()函式要求緩衝區的數量，再透過 ioctl()函式中的 VIDIOC\_QUERYBUF 確認緩衝區是否為所需要的狀態。

- (6) 開始擷取影像(Start Capturing)

在所有設定都完成後，用 ioctl()函式中的 VIDIOC\_QBUF 與 VIDIOC\_DQBUF 交換緩衝區與驅動空間的影像資料，並用 VIDIOC\_STREAMON 這 ioctl()函式開啟影像串流。

- (7) 影像資料的處理(Process Image)

擷取進來之串流影像為 Motion JPEG 格式(簡稱 MJPEG)，由於

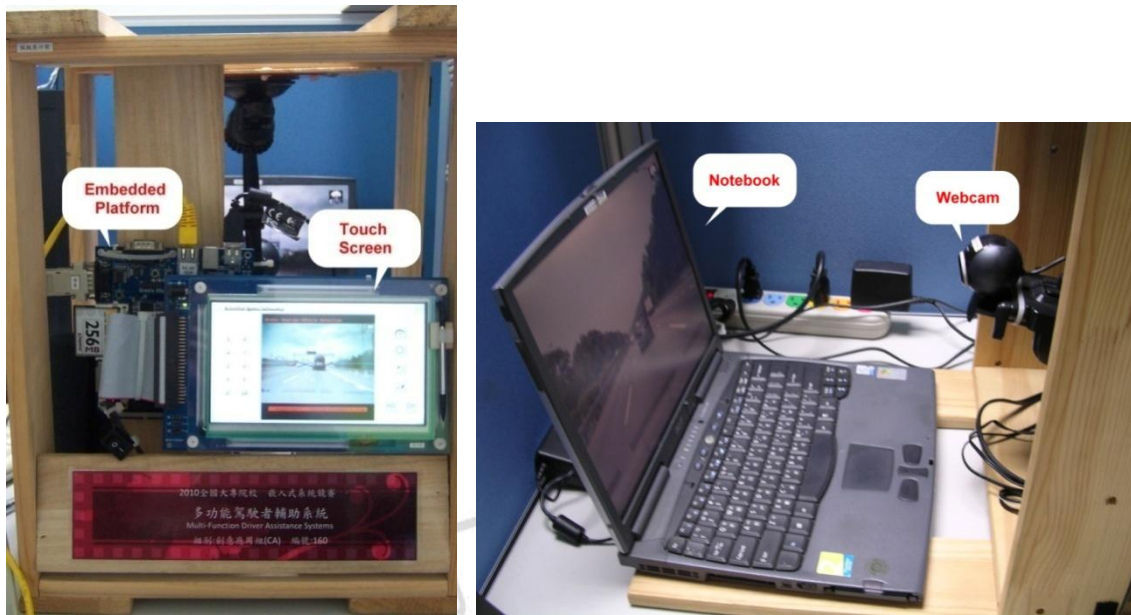
MiniGUI 提供之 API 無法對 MJPEG 格式之影像進行處理。而 MJPEG 與 JPEG 近似，因此本文將 MJPEG 轉換至 JPEG 供 MiniGUI API 使用。其中，本文將 MJPEG 中 AVI 檔頭換成 JPEG 的 JFIF 檔頭，並插入 MJPEG 沒有的霍夫曼編碼表(Define Huffman Table, DHT)，完成 MJPEG 轉換成 JPEG 的處理。

#### (8) 關閉影像擷取設備(Close Video Device)

關閉時，用 `ioctl()` 函式中的 `VIDIOC_STREAMOFF` 關閉影像串流，並透過 `munmap()` 解除記憶體映射。最後，呼叫 `close()` 關閉影像擷取設備。

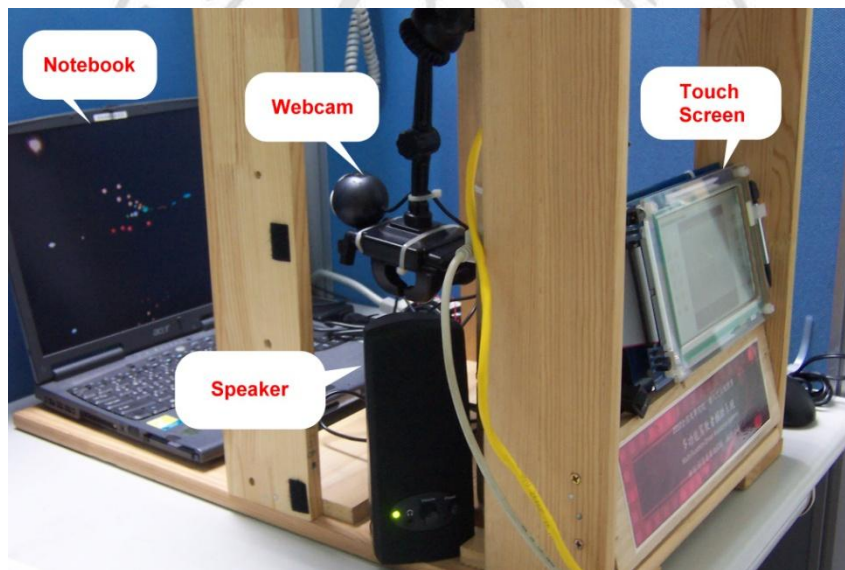
### 1.4.4 系統開發與測試方式

本論文對系統開發與測試方法分為兩階段(1)模擬開發階段(2)道路實測階段。為了減少上高速公路的次數與考量到開發的便利性，本文將 Webcam 裝設在車輛擋風玻璃內側朝車頭方向拍攝前方道路影像，並用筆記型電腦錄製行駛時的前方道路影像。在演算法的設計與程式的開發過程中，利用筆記型電腦播放預先錄製的道路影像並將 Webcam 朝向筆記型電腦螢幕拍攝，模擬在高速公路開車的道路情況。圖 1.7 為系統架設在實驗室的模擬狀況，圖 1.7(a)為模擬系統正視圖，圖 1.7(b)為模擬系統之攝影機擷取筆記型電腦螢幕中預先錄製的道路影片，圖 1.7(c)為模擬系統之側視圖。



(a)

(b)



(c)

圖1.7 實驗室模擬測試系統架設圖

完成模擬開發階段之後，本文將系統實際裝載於車內，首先將攝影機固定在擋風玻璃內側並朝車頭方向拍攝，並透過 USB 介面連接至嵌入式平台，行駛在道路上來驗證系統的穩定度及可靠度，實際道路測試之系統架設如圖 1.8 所示。





圖1.8 實際道路測試之系統架設

## 1.5 全日車輛偵測演算法架構

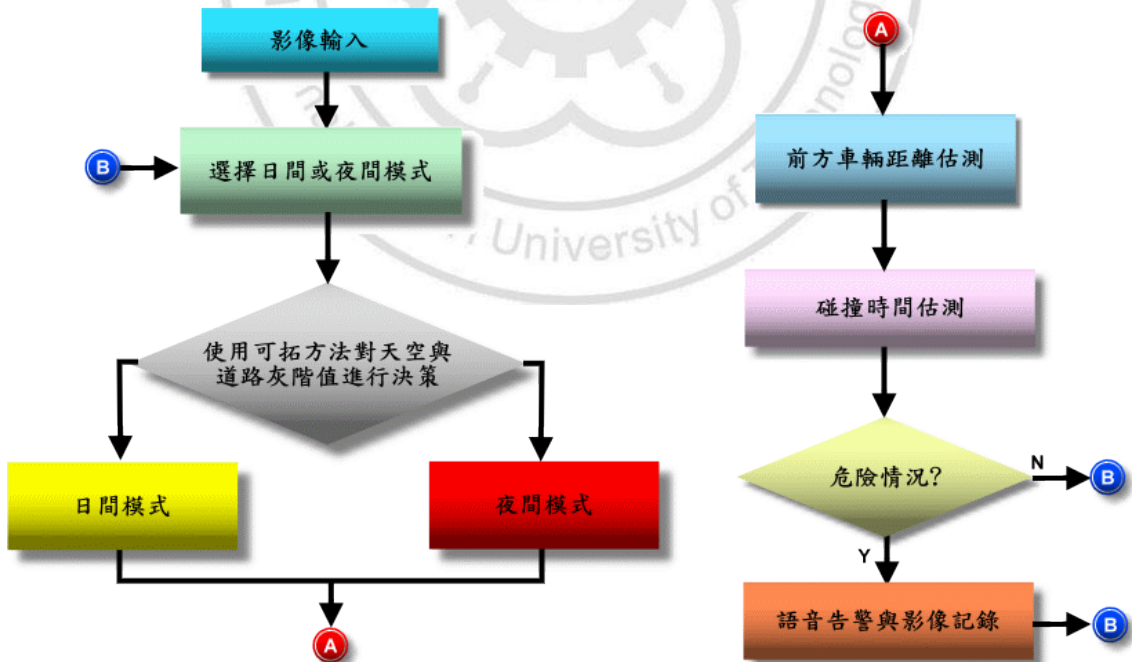


圖1.9 全日車輛偵測演算法架構

本論文所提出之全日車輛偵測演算法架構如圖 1.9。系統將影像輸入，由於日間與夜間之車輛偵測特徵不同，必須先選擇適用之模式，透過天空與道路灰階值並配合可拓理論[3]進行決策，選擇日間或夜間車輛偵測演算法進行車輛之偵測。當車輛偵測成功後會在影像中標示前方車輛之位置，並利用距離估測模型[4]計算出前方車輛距離，並算出與前方車輛的碰撞時間(Time to Collision, TTC)。之後，藉由駕駛者車輛與前方車輛之間的距離與碰撞時間這兩者因素判斷是否為危險之情況，系統在危險情況時會輸出語音的警示與紀錄當時之影像。

## 1.6 論文貢獻

預防交通事故的發生是本論文的主要貢獻，在高速公路環境下有許多的意外的發生耽誤駕駛者們保貴的時間與喪失人們珍貴的性命。其中在夜間情況下駕駛者行駛於高速公路上較日間情況中易感到疲倦，原因在於昏暗的燈光處於人們易眠的狀態，且經過了一日的勞累在夜間駕車發生車禍的比例遠高於日間。本文使用單一 CMOS 網路攝影機將駕駛輔助系統中的日間與夜間車輛偵測與追蹤實現在嵌入式系統中，並有車道偵測、車距判斷與語音警示等功能，當駕駛者車輛與前方車輛距離過近時，系統會提出語音警示提醒駕駛者；以避免意外的發生，並且記錄當時的影像、日期、時間與車距，以供不幸發生意外後的調查與責任釐清。使用低價位的 CMOS 實現系統的主要功能，是為了成本的考量，降低成本的所需並實作於嵌入式系統上，增加未來商品化的可能性與普及性，為預防交通事故的工作盡一份心力。

## 1.7 論文架構

本論文共分為八個章節，第一章為緒論，描述目前在視覺基礎之車輛偵測的環境與背景，並介紹本論文所使用的硬體與軟體架構、測試方法、論文之貢獻…等。第二章，針對車輛偵測之相關研究進行分析與討論。第三章與第四章為日間

與夜間車輛偵測的實現方法，是本論文之核心，詳細的介紹所有使用到之影像處理過程與方法。第五章為車輛測距與碰撞時間估測之方法。第六章為運用可拓理論實現日夜間決策之方法。第七章為本論文之實驗結果與分析。第八章為結論與現存問題之分析。最後是本論文的參考文獻。





## 第二章 相關研究

隨著製程技術的改良與處理器運算效能的提升，近十幾年來已經累積了許多以機器視覺為基礎的車輛偵測研究的文獻。其中，車輛偵測包含的範圍是很廣義的，從天色來區分就可分為日間與夜間兩大類，就天候角度有雨天、陰天、起霧與下雪等，而由車輛種類區分也可分為貨車、大客車、房車等，不勝枚舉。許多車輛偵測方法也結合了車道線偵測來加強車輛偵測的完整性與準確性。目前常見的文獻中多數是以日間與夜間的車輛偵測作為研究的方向，另外車道線偵測也是大多數車輛偵測文獻中有引用及介紹，在此本文針對日間、夜間車輛偵測與車道線偵測這三大方向進行文獻的探討。

### 2.1 日間車輛偵測之文獻探討

Sun 等人 [5]統整了許多日間車輛偵測的文獻，由他的觀察與分析後，提出了一個車輛偵測的主要概念與步驟，如圖 2.1。主要是：

- (1)假設的產生(Hypothesis Generation, HG)：在影像中找尋可能為車輛的影像區域。
- (2)假設的確認(Hypothesis Verification, HV)：藉由各種驗證的方法，確認假設的車輛影像，是否為車輛。

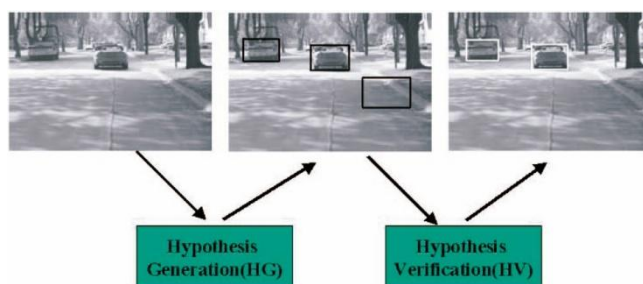


圖2.1 車輛偵測之兩個步驟 [5]

Kato 等人[6]主要運用多叢集改良型二次判別函式(Multiclustered Modified Quadratic Discriminant Function, MC-MQDF)進行車輛的偵測與辨識。主要處理步驟為(1)資料學習處理(2)車輛辨識處理。首先在離線(Off-line)狀態下進行資料的訓練，擷取車輛與非車輛的影像圖片與正規化，再經由 MC-MQDF 取出兩者的特徵並共同分佈在一個 N 維的特徵空間，分別對車輛與非車輛特徵做叢集取出車輛與非車輛的特徵值。實際進行車輛偵測與辨識，取出影像中可能為車輛的區域，擷取特徵值並與先前訓練的特徵值比對，判定是否為車輛。此方法因 MC-MQDF 具有多叢集功能，所以並不侷限在只分辨出車輛與非車輛，可以由設計者的設定與訓練資料的種類，加入車輛識別的能力，文獻中能分辨出機車、大型車與小客車種。此方能有效的偵測並辨識車輛與車種，但需耗費冗長資料訓練時間，以及作者將此方法實作於 PC 上，由此可知實際偵測時，擷取車輛影像、正規化與特徵擷取所花費的系統運算量過於龐大，不易實現於嵌入式系統中。

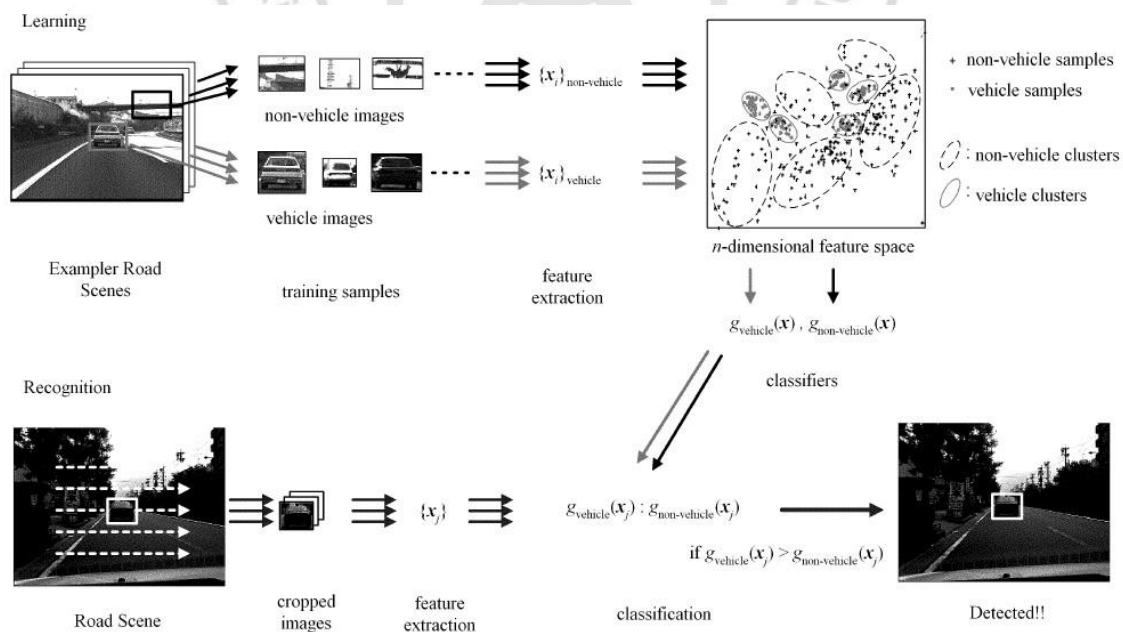


圖2.2 以 MC-MQDF 方法實現前方車輛偵測與識別之概念圖 [6]

Chern and Shry 等人 [7]利用日間情況下影像中車輛底部陰影的特徵進行車輛偵測。藉由車輛底部的陰影找出前方車輛，此方法主要由三個步驟組成(1)路面與陰影的判斷(2)前車陰影的偵測(3)前車輪廓找尋。一般的白天道路情況下，車底陰影的亮度是較道路表面還暗，因此計算像素之間的比例值找出陰影的區域。一般而言車輛底部陰影具有兩大特徵(1)橫向的長條狀(2)車輛底部陰影寬度與車道寬度持有一特定比例(如圖 2.3)，以中華民國高速工路之車道寬度為例，車輛寬度約為車道寬度的 0.45 至 0.75 倍。利用以上兩項特徵找出前方車輛的陰影。最後，根據車輛形體的長寬比例由車輛底部陰影寬度推算出車輛的所在區域，並對該區域進行垂直邊緣偵測找出車輛左右邊界，車輛上邊界則使用水平邊緣偵測找尋，如順利找到各個邊緣則完成車輛偵測。反之，該區域找尋不到左、右與上邊緣則判定此陰影不為車輛底部陰影。藉由車輛底部陰影找尋前方車輛，方法實用且實作難度不高，實現於嵌入式系統中所耗費的系統運算時間也是可以接受的範圍，但對天況的適應性較差，當遇到夕陽或早晨等太陽斜照的情況，容易受到樹木、旁車、高架橋的陰影干擾，影響車輛偵測的正確性。

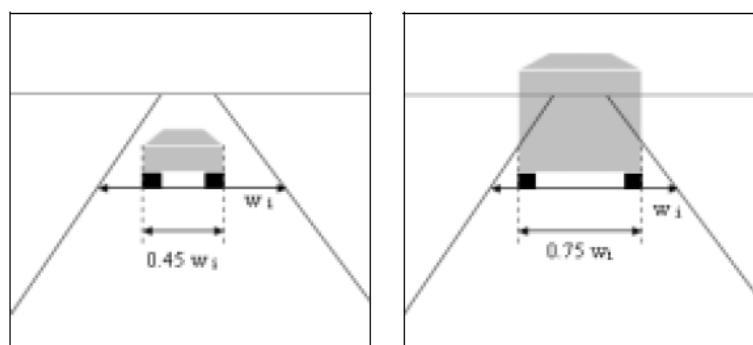


圖2.3 車寬與道路寬度之比例範圍 [7]

Chang 等人 [8]提出的方法主要是利用車輛的腳印(底部邊緣)偵測前方車輛。一開始為了避開不必要的處理，定義出一個車輛可能出現的感興趣區域(Region of Interest, ROI)，之後對該區域進行二值化分離前景與背景。其中，前景主要包括了可能為車輛的輪廓、腳印與道路標示線，再進行道路標示線消除的程序消除道路標示線等非車輛腳印雜訊。隨後，利用可能為車輛之腳印計算出車輛的區域並產生一邊界盒框選，再利用車輛左右對稱之特性驗證該車輛區域，如未滿足對稱性質則判定此車輛區域是無效的。接下來透過投影量分析方法找出前方車輛正確的上、下、左與右邊界。使用 Chang 所提出的車輛偵測方法能夠縮小車輛搜尋的區域減少系統的運算量與革除不必要的影像資訊，消除道路標示線能減少非車輛腳印的干擾，左右對稱性運算更能降低誤判的機會，而利用投影量分析的方式找到車輛的邊界能避免太陽斜照產生腳印拉長的影響(如圖 2.4)，其中 $X_{lb}$ 與 $X_{rb}$ 代表為車輛之左右邊界。但 Chang 在文獻中提出的方法是以 PC 實現，因此較無考量到硬體資源限制的問題。其中，對稱性運算方法使用 R、G、B 三灰階度共同計算，在嵌入式系統中將耗費許多運算的時間。

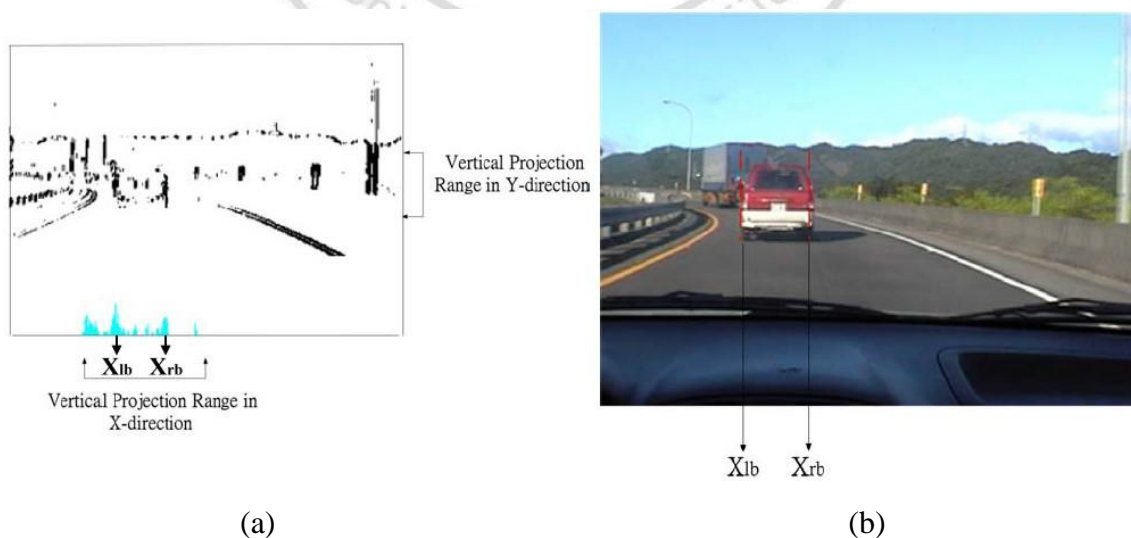


圖2.4 使用投影量分析找出正確的車輛左右邊界 [8]

文獻[9]中提出了以光流為基礎的方法偵測前方車輛。光流與影像流之定義極為相似。假設當有一連串的連續影像序列中，前一張與下一張影像之目標物件是靜止不動的狀態，此時在影像流的觀點來看目標物件的影像流值為零，但是以光流的觀點來解釋，由於光源在環境中是不斷的變化，因此即使目標物件並無動作，但仍是存在光流值。這便是影像流與光流兩者之差異。在文獻中提出了多種光流的計算方法，其中以誤差平方加總法(Sum of Squared Differences, SSD)最為常見，主要計算子影像區域中的值，並找出該區域中 SSD 的最小值，將兩張影像之間的位移使用向量表示，該向量即為兩張影像之光流。在動態影像中光流位移的大小，背景的變化小於車輛突然出現的變化程度，利用此特性分析光流之位移找出車輛的概略位置。圖 2.5(a)序列影像，圖 2.5(b)為序列影像之光流場，圖 2.5(c)為根據光流之分析所找出的車輛概略位置。運用光流法方式找出前方車輛能夠一同計算出車輛之間的相對速度，對駕駛者而言獲得此資訊，在避免事故的發生中無非是多增加了一項參考的指標。但要計算整張影像的光流場，在嵌入式系統中同樣的會陷入有限硬體資源的考驗且抗雜訊能力較差，在安全防護的系統中，應用光流法偵測車輛之準確度仍是有待改進之處。

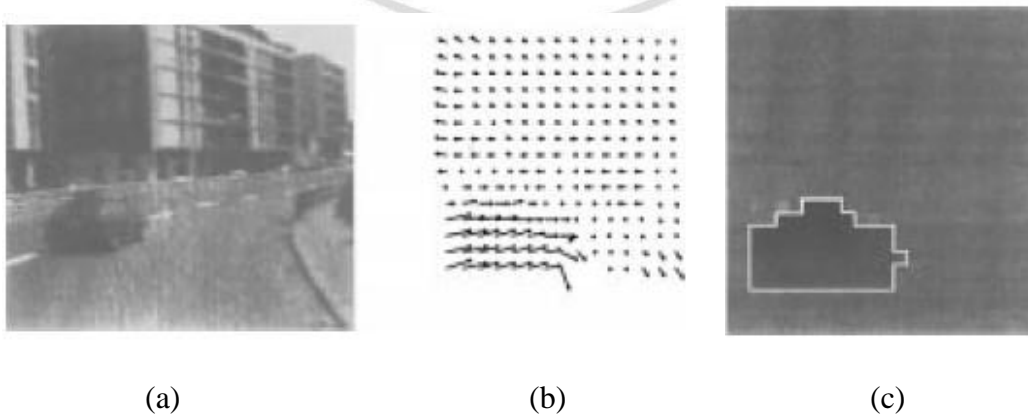


圖2.5 使用光流法偵測前方車輛影像 [9]

## 2.2 夜間車輛偵測文獻探討

當黑夜籠罩大地，影像中車輛的特徵就不同於日間情況下，多數的研究 [10]-[14][22][25] 大多使用車輛成對的車輛尾燈作為偵測車輛所用之特徵。另外，也有結合其他主動式感測器(雷達、雷射等)作為輔助的偵測車輛之方法 [15]。

Lu [10] 提出一夜間前方車輛距離警示的方法，利用影像中的車輛尾燈偵測前方車輛。首先，藉由前方車輛的成對尾燈橫跨影像中心線特性，區隔行駛於同車道與旁側車道的車輛尾燈，圖 2.6 中共存在著三台不同遠近距離之車尾燈，分別以  $c_1$ 、 $c_2$  與  $c_3$  表示車輛， $P_R$  與  $P_L$  為該車輛的右與左邊之車尾燈，並由左至右給予車尾燈編號  $N(i)$ ，利用式 (2.1) 與 (2.2) 計算出每兩個相鄰車尾燈是否有橫跨中線，找出同車道車輛車尾燈， $N_H(\max)$  為影像之寬度。接下來，利用峰值分析找出高亮度的座標位置即為車輛之尾燈進而偵測到車輛的位置。圖 2.7(a) 夜間道路環境之影像，圖 2.7(b) 為峰值分析的結果，圖 2.7(c) 為經由區隔車輛處理後找出的前方車輛。此方法相當的精簡並且快速，能夠執行在嵌入式系統中，並有不差的效果。不過只能偵測同行駛車道之車輛，無法偵測旁側車道車輛，且理論過於單薄對於環境的適應性仍有待考良之處。

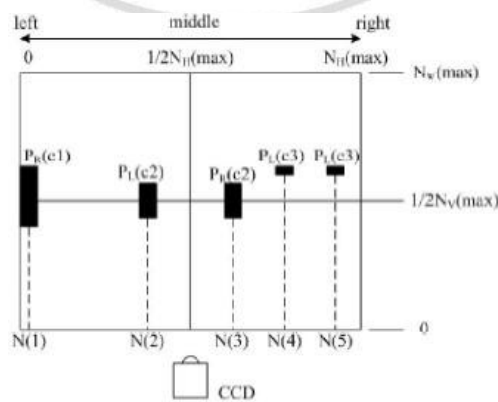


圖 2.6 運用車尾燈橫跨中線特性區分不同車道之車輛 [10]

$$\left[ \frac{N(i) + N(i + 1) - N_H(max)}{2} \right]^2 \leq \delta(i) \quad (2.1)$$

$$\delta(i) = \left[ \frac{N(i + 1) - N(i)}{2} \right]^2 \quad (2.2)$$

其中  $i = 1 \dots N$

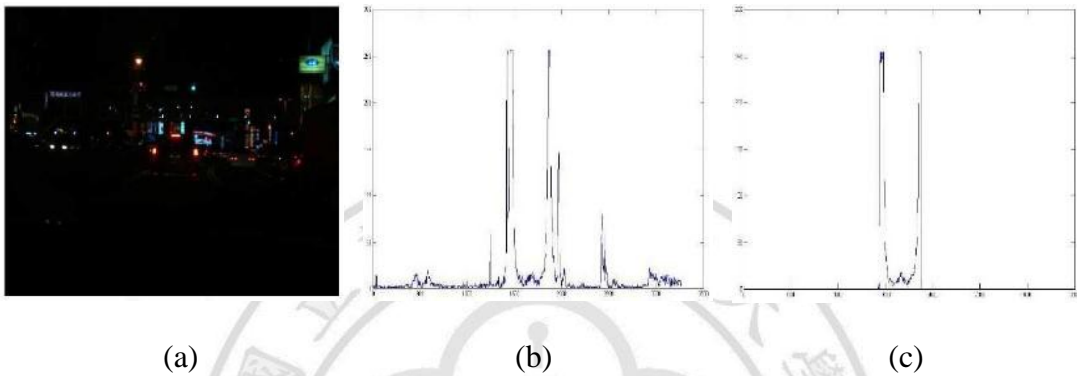


圖2.7 (a)夜間道路原始影像 (b)峰值分析之結果 (c)經由區隔車輛處理之結果 [10]

Chen and Wu 等人[11][12]的方法是先利用直方圖統計的方式找出動態影像中二值化的門檻值取出明亮物件並且定義出一條虛擬的地平線以隔離道路之外的明亮物件(交通號誌、廣告招牌等)。再使用相鄰元素標籤法(Connected-Component Labeling)分割明亮物件影像，之後判斷各個亮點物件之 R、G、B 三灰階色度的比值，區分來車的頭燈與前方車輛的車尾燈，其中頭燈之特徵為白、亮，尾燈則是由紅色與亮度所組成。同一台車輛之成對車尾燈存在著形狀與距離比例的規則，因此運用此規則比對各個明亮物件找出車輛的成對車尾燈，並將比對成功之結果予以框選。圖 2.8(a)顯示虛擬地平線與明亮物件分割的結果，圖 2.8(b)為成對車尾燈形狀與距離比例之規則示意圖。使用 Chen and Wu 所提出的方法主要透過配對車燈來偵測出前方的車輛與旁側車道之車輛，並且能藉由車燈的色度區分出來車



與行駛於前方之車輛。不難發現，成對車尾燈存在的規則中門檻值的設定是影響車輛偵測準確性的重要因素。但是車輛之成對車尾燈存在的規則是較偏向使用經驗值來定義，當車輛的種類或形體較為不同時，偵測車輛的效果就會容易受到影響。

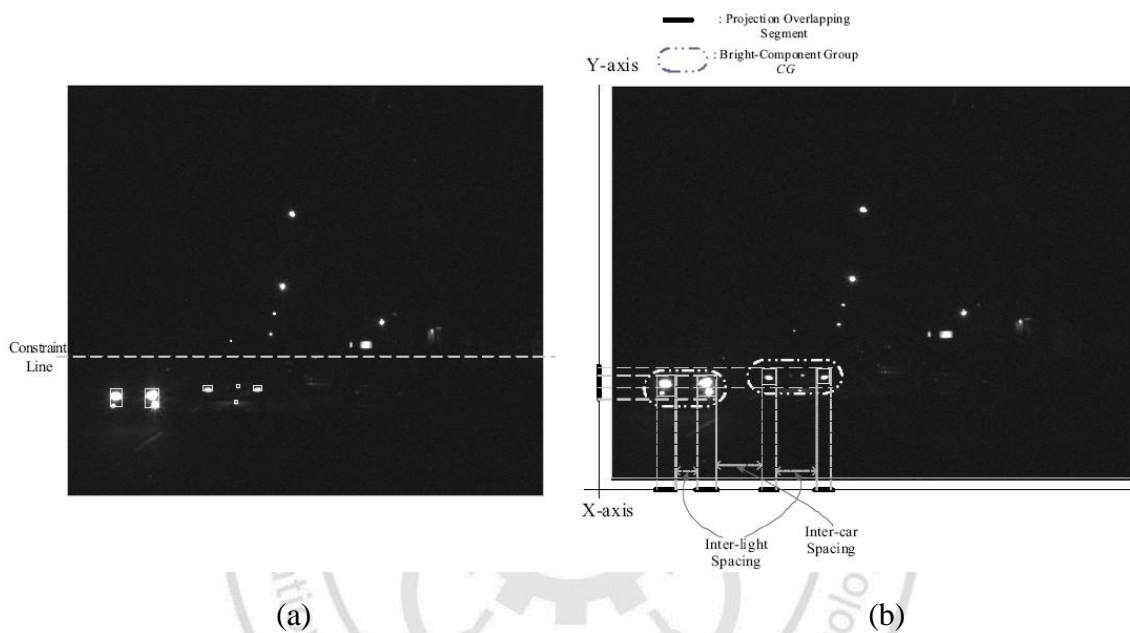


圖2.8 (a)虛擬地平線與明亮物件分割的結果 (b)成對車尾燈形狀與距離比例之規則示意 [11]

Wang and Huang 等人[13]與文獻[11][12]所提出方法略同，一樣的利用車輛尾燈特徵偵測前方車輛。主要分為五大步驟(1)色彩空間轉換(2) Gaussian Filter(3)適應性二值化(4)尾燈重心偵測(5)車尾燈配對。首先，將 RGB 影像轉換到讓車尾燈特徵更為明顯的色彩空間，此色彩空間主要利用該 Pixel 中的 R 與 B 相減的值所求得，其影像為 8bit，如圖 2.9(a)。再利用 Gaussian Filter 模糊影像去除雜訊，再進行適應性二值化取出可能為車尾燈的影像，如圖 2.9(b)。由於作者認為車尾燈影像的形體大小並無一致，因此找出各車尾燈影像之重心以利後續的車尾燈配對。根據車



輛外觀形狀與成對車尾燈之間的距離偵測車輛。Wang and Huang 的方法使用了數個影像前處理方法留下可能為車尾燈的影像，降低後續影像處理的負擔。並且使用車尾燈影像的重心進行配對，能減少前方車輛煞車時車尾燈膨脹變形帶給車尾燈配對的影響。不過此方法受限於一般小客車，並且與文獻[11][12]方法遇到相同的問題，就是使用經驗值當作車尾燈配對的門檻值，易受到車輛外型的種類影響。

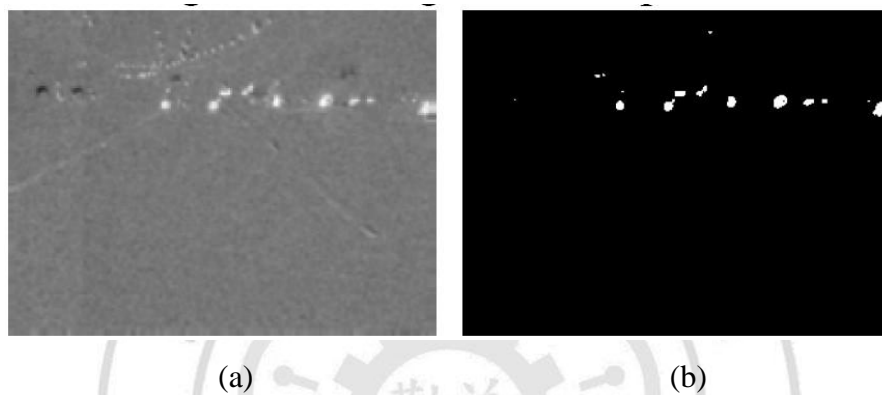


圖2.9 (a)RGB 影像轉換至車尾燈特徵較為明顯之色彩空間 (b)車尾燈影像的擷取 [13]

Gormer and Muller 等人[14]分析了車尾燈的色度組成，找出適合的門檻值擷取車尾燈影像。接下來對各個車尾燈進行配對的程序，首先找出兩個車尾燈影像之間的水平與垂直距離範圍合理的配對組合，再篩選出正確的配對組合，其篩選的步驟主要由以下四點構成：

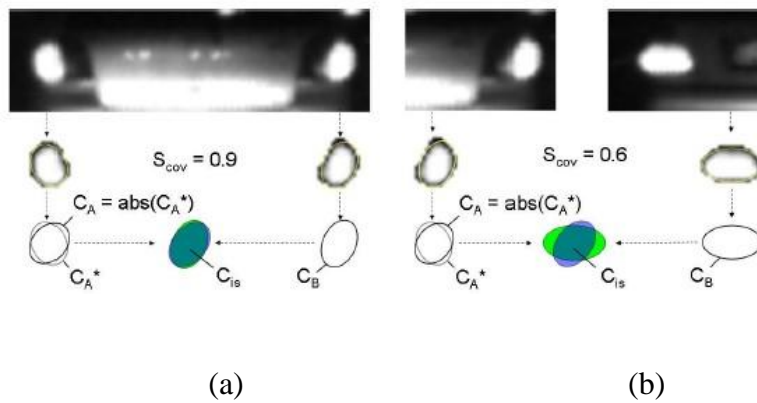


圖2.10 車尾燈轉換共變異矩陣進行面積大小比對 (a)同一車輛之車尾燈 (b)

不同車輛之車尾燈 [14]

- (1) 利用成對車尾燈大小面積相同的特性，將兩車尾燈影像轉換至共變異矩陣並進行正規化，再把兩矩陣進行相減以判斷大小面積是否相同，如圖 2.10 所示。
- (2) 比較配對組合中兩個車尾燈影像的光暈與本體之比例大小，如圖 2.11

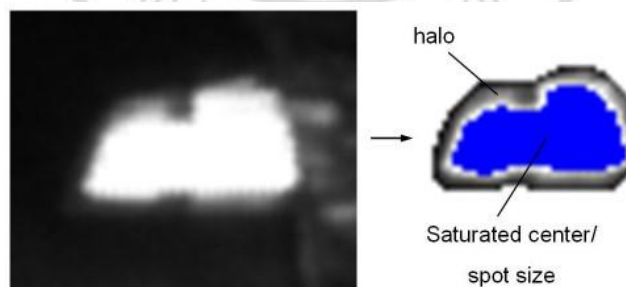


圖2.11 車尾燈光暈與實體示意圖 [14]

- (3) 利用連續影像的特性，判斷 N 張序列影像中同一配對組合之車尾燈垂直移動的距離是否相等，如圖 2.12。

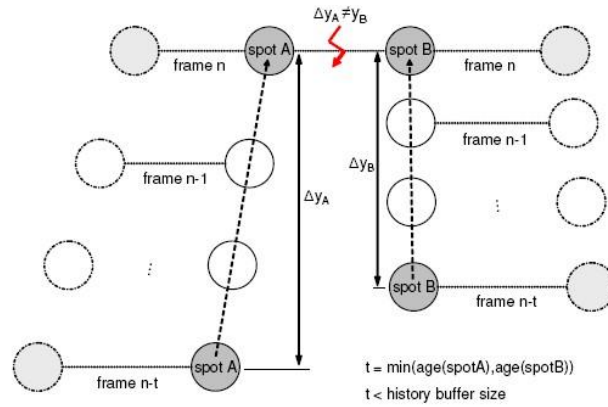


圖2.12 連續影像中配對組合之車尾燈垂直移動距離 [14]

- (4) 根據配對組合中，車尾燈影像之間的距離與大小對照實際的距離進行比對。

假如配對組合滿足上述四點，則判斷此一配對組合為一車輛之成對車尾燈。此文獻所提出的車尾燈配對方法雖較為複雜，但是有較好的偵測效果，與文獻[10]-[13]的車尾燈配對方法相比擁有較佳的適應性與正確性。但是由於方法較為複雜，有共變異矩陣的轉換與浮點數運算等處理，在硬體資源有限的嵌入式平台上執行有較大的難度。

Kim 等人[15]，在夜間車輛偵測中提出了結合視覺影像及聲納感測器的方法，同樣的也是利用車輛的車燈偵測車輛，其不同為加入了聲納感測方式，先利用聲納感測器偵測出前方物體的位置距離，並轉換置影像中的相對應位置，找出相對應位置的車燈，並判斷車燈大小及找尋車燈的重心，並求出車燈的信任度(Degree of Trust, DOT)，判定是否由無符合門檻值，來達到車輛偵測之結果。圖 2.13 為車尾燈的範圍與重心搜尋結果。



圖2.13 車尾燈的範圍與重心標示 [15]

## 2.3 車道線偵測文獻探討

在以視覺為基礎的車輛偵測中，車道線偵測是非常重要的環節。一般來說車道線偵測的應用功能主要包含(1)縮小車輛偵測搜尋的範圍(2)區分車輛行駛的車道位置(3)車道線偏移警示。以下將探討先前研究所提出的車道線偵測方法與其特點。

Hsiao and Yeh 等人[16][17]提出了車道線偵測方法，並應用於車道線偏移上。首先觀察影像中車道標示線的特性，整理歸納出幾點(1)透過水平方向掃描，可發現車道線區域為「黑-白-黑」(2)車道線形狀細長(3)車道線之線斷相鄰。在車道偵測前會先執行影像的前處理，將彩色影像灰階化以及將地平線以上的影像消除，並使用高斯濾波器將影像平滑化，消除胡椒粒般的雜訊，之後使用區域二值化方式[18]找出影像中物件的邊緣並二值化擷取車道線影像，以利後續的車道線偵測處理，影像前處理過程如圖 2.14。

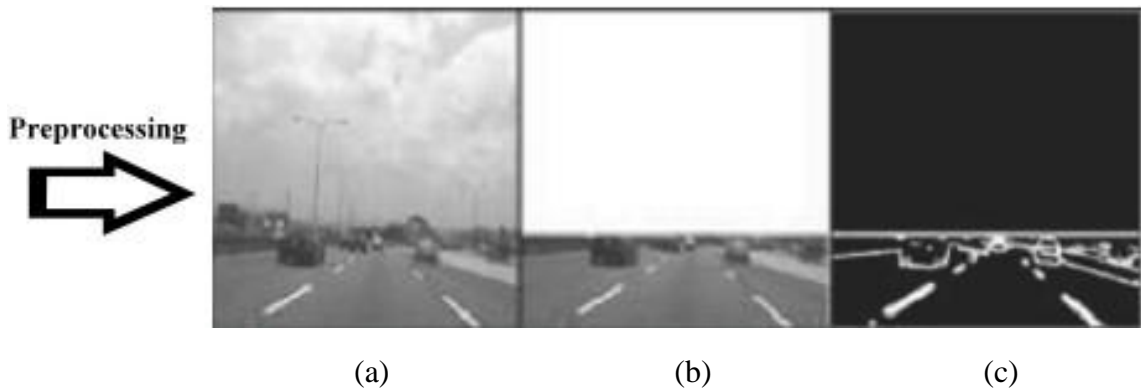


圖2.14 車道線偵測前的影像前處理程序 (a)原始道路影像 (b)高斯模糊與影像切割 (c)區域二值化 [16]

經過影像前處理程序之後，接著找尋車道線的特徵。將影像經由水平掃描並分析找出「黑-白-黑」之區域，地面為黑車道線為白。圖 2.16 與為水平掃描的分析圖例，圖 2.15 是一個峰與谷的關係。圖 2.15 中定義了六個變數  $P_s$ 、 $P_e$ 、 $P_p$ 、 $V_s$ 、 $V_e$ 、 $V_p$ ，各自代表為峰值第一個上升的點、峰值最後下降的點、峰值第一個開始下降的點、第一個上升點的灰階值、最後下降點的灰階值、峰值第一個下降點的灰階值。之後根據幾個判斷條件找出車道線的峰值點(Peak Point)位置，判斷條件為(1)左邊與右邊峰值高度是否大於預設門檻值(2)整個峰值的寬度是否大於預設門檻值，其寫成條件事如式(2.3)。

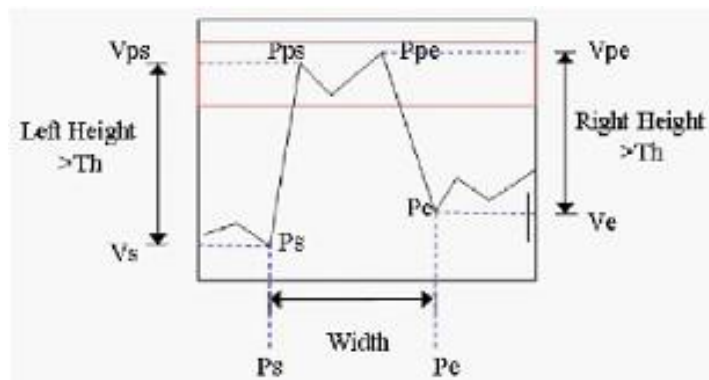


圖2.15 車道線的峰與谷的例子 [17]

$$\begin{aligned}
 & \text{If } (V_{ps} - V_s) > T_h \text{ and } (V_{pe} - V_e) > T_h \text{ and } (P_s - P_e) < T_w \\
 & \text{Then } V_p = (V_{ps} + V_{pe})/2 \text{ and } P_p = (P_{ps} + P_{pe})/2 \quad (2.3)
 \end{aligned}$$

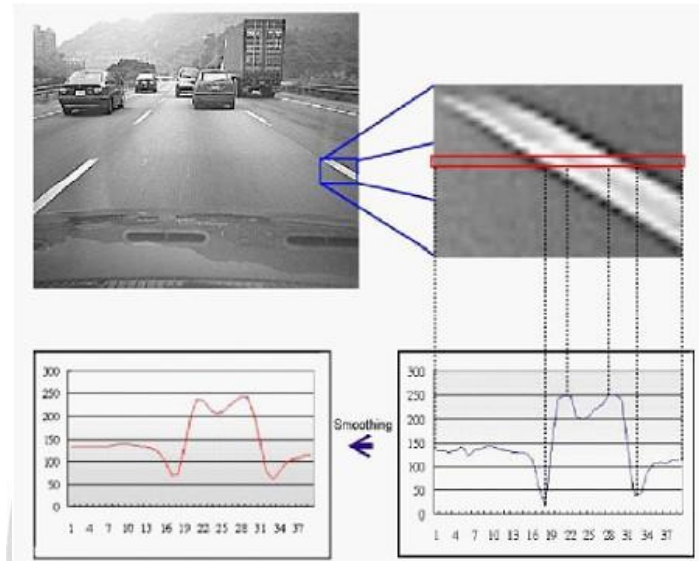


圖2.16 水平掃描的分析圖例 [17]

找出 Peak Point 之後，使用一  $3 \times 9$  的遮罩(如圖 2.17(a))由下往上對相鄰的 Peak Point 進行群組化的動作，當相鄰的 Peak Point 被群組後則以新的 Peak Point 為中心點繼續找尋相鄰的 Peak Point 形成車道線段。接著定義各個線段的起點、終點、斜率與截距，並計算同一方向相鄰線段的偏移值，當偏移值的大小在一預設門檻值之內，則將兩線段合併，將新線段的終點設定為被合併線段的終點，起點為原線段起點，斜率則為原線段之斜率，截距則再重新計算，如此便可將各線段合併串連，最後找出左右各一條最長的道路線段，標示出車道標示線。圖 2.17(b)為線段合併串連的示意圖。此車道偵測方法，能有效的偵測出車道標示線，並可執行在嵌入式系統中。不過只適用於筆直的道路線上，對於彎道的道路線將會失去一定的準確度。



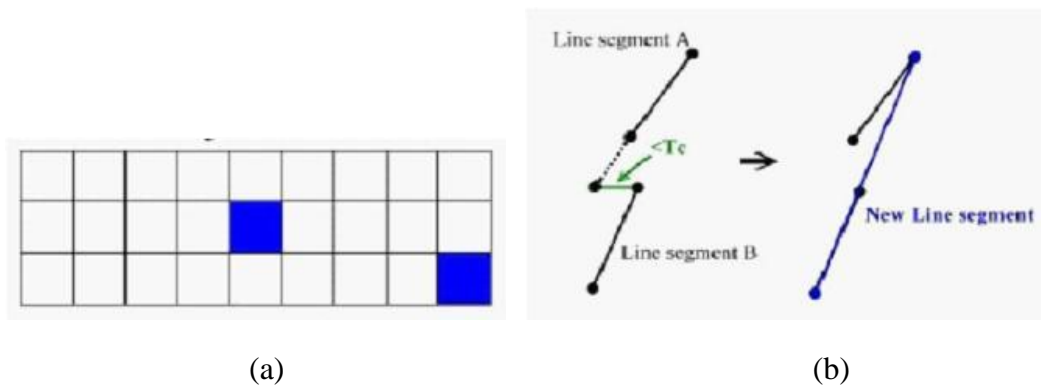


圖2.17 (a)3 X 9 的群組化用遮罩 (b)線段合併串連的示意圖 [17]

Chern and shyr 等人[7]提出了使用二次曲線近似的方法，偵測高速公路上彎道與直線道路的車道標示線。首先將影像劃分為三部分，為了少去多餘的處理將天空與道路影像分割，再把道路影像切成左與右區塊，如圖 2.18 所示。之後，使用四十五度邊緣運算子對道路影像的左與右區塊進行邊緣偵測，圖 2.19(a)的邊緣運算子為道路影像左區塊所使用，圖 2.19(b)則是用於道路右區塊。經由四十五度邊緣偵測後，再透過設定的門檻值產生一正、負邊緣影像，影像中正負邊緣的交會點即為車道線的峰值點，隨後進行群組化及線段連接的程序，產生車道線段的邊緣。最後使用二次曲線近似的方法標示車道線。使用二次曲線近似對彎道的車道線偵測有較佳的效果，在直線的車道線上也有頗佳的偵測能力。但當車道線邊緣點不足時，很可能會出現近似較差的情況，且運算量較高，能否於有限硬體資源中實現，仍是需考量的問題。

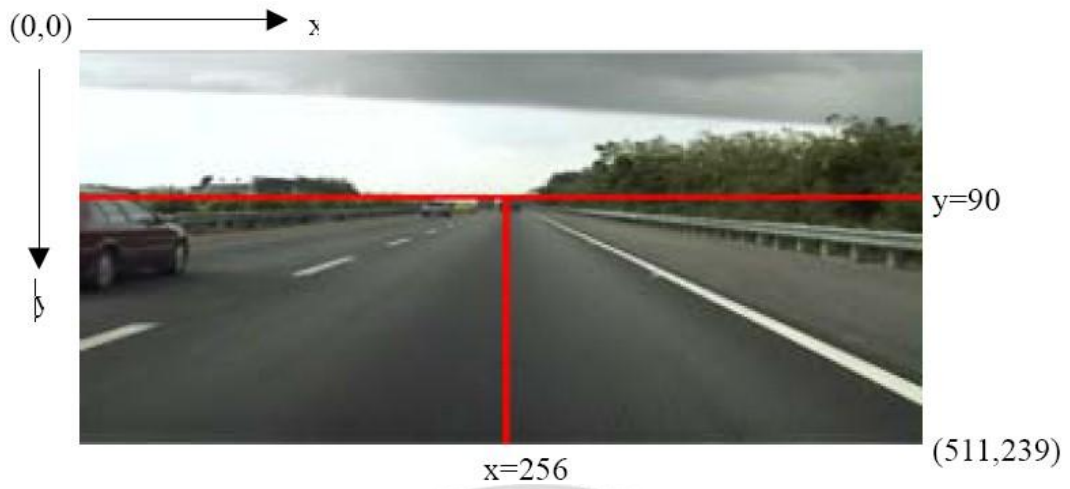


圖2.18 影像之區域劃分[7]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

(a)

(b)

圖2.19 四十五度邊緣運算子 (a)左半部區域所用 (b)用於右半部區域 [7]



### 第三章 日間車輛偵測

本論文在日間車輛偵測中，以台灣地區的高速公路為測試的地點，測試環境為光照充足的日間情況，主要的工作是在動態背景中找到前方的車輛，並在影像中標示車輛位置作為測距時影像深度的判斷。以下將簡單介紹日間車輛偵測演算法，此部份是由李彥鋒學長完成，更詳細的內容記載於文獻[19]。

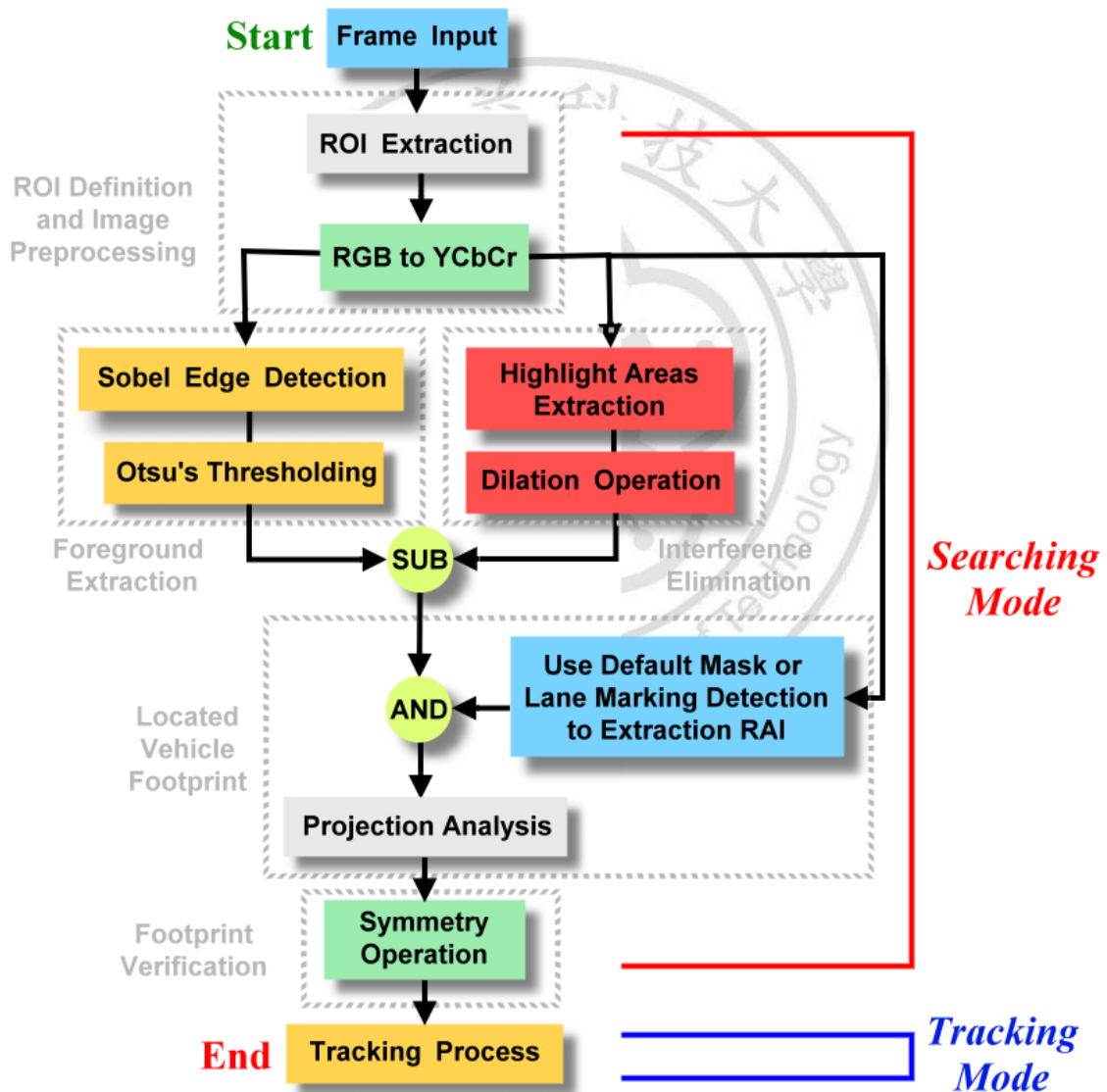


圖3.1 日間車輛偵測演算法執行之程序

圖 3.1 為日間的車輛偵測演算法執行之程序，由於變動的背景下無法使用影像相減的方式找到變動的區域(物件)，因此本文設計一個能在動態影像中偵測出前方車輛的演算法，主要分成搜尋、追蹤車輛兩個部份。在搜尋模式中，本文利用車輛底部邊緣的特徵(車輛腳印，有可能是車輛的底部陰影或保險桿)找尋車輛，本文將搜尋的程序分為(1)縮小搜尋區域與前處理，主要是為了讓演算法能在硬體資源有限的嵌入式系統中更有效率的執行。(2)前景擷取，分離前景與背景取出車輛輪廓、車道標示線等影像。(3)路面雜訊濾除，把前景中的不需要的雜訊移除(例：車道標示線)，留下車輛底部邊緣的影像。(4)標示車輛腳印，利用車道線偵測區分出前方車輛腳印所在的車道，並使用投影分析標示腳印位置。(5)車輛腳印驗證，利用車輛左右對稱的特性對車輛區域的影像進行驗證。如驗證成功，再使用一邊界盒將車輛影像框起。不難發現搜尋模式的多程序影像處理會耗費許多系統運算量，因此提出追蹤模式降低整體運算量。使用此類多層次的影像處理方法，必定會影響到系統運算的即時性，為了能在嵌入式平台上實現，慎選影像處理演算法讓處理效率更佳是重要的課題。接下來介紹影像處理之程序。

### **3.1 縮小搜尋區域與前處理 (ROI Definition and Image Preprocessing)**

為了減少系統運算量並提升車輛搜尋的效率，在搜尋車輛之前本文先行定義影像中的感興趣區域(Region of Interest, ROI)，縮小搜尋車輛的區域。本論文用輸入影像中的地平線作為 ROI 之上邊界，地平線以上區域為天空也是非道路之區域，車輛並不會出現於此處。下邊界則定義在前方約五公尺之處，除了塞車等情況，前方車輛在正常行駛狀況下並不會出現在此區域。

由於輸入影像為 RGB 色彩空間易受光源影響，因此本文將 ROI 裡 RGB 影像

轉換至 $Y C_r C_b$ 色彩空間[20]降低對光源的敏感度，並使用 Y(亮度)色系空間進行之後的影像處理程序，減少影像維度的計算，節省系統之運算量。圖 3.2 為 ROI 區域轉換至 Y 色彩空間影像顯示之結果(灰階影像)。



圖3.2 ROI 範圍定義與 ROI 中 RGB 影像轉換至灰階影像

### 3.2 前景擷取(Foreground Extraction)

本論文在這一步驟裡使用 Sobel [20]邊緣偵測強化灰階影像中物件的輪廓，並藉由 Otsu's Threshold Selection Method[21]將物件輪廓與背景分離，取出前景物件。一開始使用 Sobel 邊緣偵測強化水平與垂直紋理。式(3.1)中  $f$  為一個大小為  $M \times N$  的影像， $g(x,y)$  為邊緣運算的輸出影像， $w(s,t)$  是一大小  $m \times n$  之濾波器遮罩，其內容為像素權重值。遮罩(Mask)中心位置隨著  $x$  與  $y$  移動，遮罩內像素之索引位置則由  $s$  與  $t$  改變，其中  $a=(m-1)/2$  和  $b=(n-1)/2$ 。當中，為了產生一個完整的濾波影像，方程式必須在當  $x=0,1,2,\dots,M-1$  與  $y=0,1,2,\dots,N-1$  時使用，遮罩中  $m$  與  $n$  之大小需相同且為奇數，能接受的最小數值為 3。圖 3.3 為 Sobel 的水平與垂直濾波遮

罩權重分配，圖 3.4 是經由 Sobel 邊緣偵測後的影像輸出結果。

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (3.1)$$

-1	-2	-1
0	0	0
1	2	1

Horizontal

-1	0	1
-2	0	1
-1	0	1

Vertical

圖3.3 Sobel 的水平與垂直濾波遮罩權重分配



圖3.4 將 ROI 中灰階影像經由 Sobel 水平與垂直邊緣偵測後之影像

經過 Sobel 邊緣偵測後的影像，會加強影像中高頻的部份，並減弱影像中低頻的區段；高頻部份多半是物件輪廓之線條，低頻區段則為平坦的區域居多。觀察圖 3.4 可發現留下的影像大部份是車輛的輪廓與物件的線條，但仍存在些許的低頻雜訊。

本論文使用 Otsu's Threshold Selection Method(簡稱 Otsu's Method)進行影像的二值化，影像二值化在物件辨識中是很重要的步驟，主要功能是把影像中背景與前景影像分離。二值化之方法依照門檻值的設定可分為固定與適應性門檻值兩種，固定門檻二值化也就是依據使用者經驗所設置二值化之門檻值，門檻值設定方法簡單、無需繁複計算與統計，但是容易受到光照度與背景變化影響。適應性門檻值必需耗費時間計算、統計出適當的門檻值，不過可以依據光照度與影像背景的變化設定出適當之門檻值。本文使用 Otsu 所提出的最佳門檻值設定方法，利用直方圖統計分析影像中的灰階分佈並計算變異量，並由前景與背景兩群組之最小變異量加總找出最佳門檻值。在 4.2 小節中對 Otsu's Method 做詳細之介紹。圖 3.5 為圖 3.4 二值化後之影像。



圖3.5 圖 3.4ROI 中影像二值化後之影像

### 3.3 路面雜訊濾除(Interference Elimination)

圖 3.5 中，有許多非車輛輪廓之影像(例:道路標示線)。為了消除路面雜訊，本文找出 ROI 中高亮度的區域影像，處理過程主要是對道路取樣後，以平均灰階值作為門檻，再使用此門檻值對影像二值化，最後將該二值化影像進行膨脹運算

[20]。

首先對 ROI 道路區域中五個彼此相隔十個像素距離之像素取樣，為了避免取樣時像素灰階度不均勻情況(像是道路上的提示文字)，以致影響高亮度區域擷取的結果，本文計算出取樣像素的平均灰階值作為二值化之門檻值。並觀察日間道路情況時路面的灰階值範圍對門檻值進行修正。之後，使用膨脹運算，突顯高亮度區域。圖 3.6 為高亮度區域擷取的結果。



圖3.6 ROI 影像中高亮度區域包含了路面與車道標示線

本文所使用的膨脹運算是二值型態學中的基本運算，其最簡單的應用之一是在連接縫隙、修補破碎的影像，早期常使用在指紋辨識的前處理。本論文使用膨脹運算並非為了連接或修補破碎之影像，而是為了擴張高亮度區域。圖 3.7 為高亮度區域經膨脹運算後之結果。

最後，把經過 Otsu's Method 與膨脹運算後得到的兩張強化影像，也就是圖 3.5 與圖 3.7，使用有條件的相減運算進行處理。如此便能將路面上的雜訊濾除(道路標示線)，減少道路區域的干擾因素，方便搜尋車輛底部邊緣特徵(車輛腳印)。圖 3.8

為上述兩張影像進行有條件相減運算之結果，經過相減運算之後幾乎能消除路面  
上的雜訊，留下車輛的輪廓與底部特徵。式(3.2)為有條件的影像相減運算式， $E(x,y)$   
為經由 Otsu's Method 後之邊緣強化影像， $W(x,y)$ 為高亮度區域膨脹後影像，  
 $NLM(x,y)$ 為經過相減運算之雜訊移除後影像。



圖3.7 對高亮度區域進行膨脹運算之結果

```
if( $E(x,y) == 255 \ \&\& \ W(x,y) == 0$ )  
     $NLM(x,y) == 255$ 
```

```
else
```

```
     $NLM(x,y) == 0$  (3.2)
```



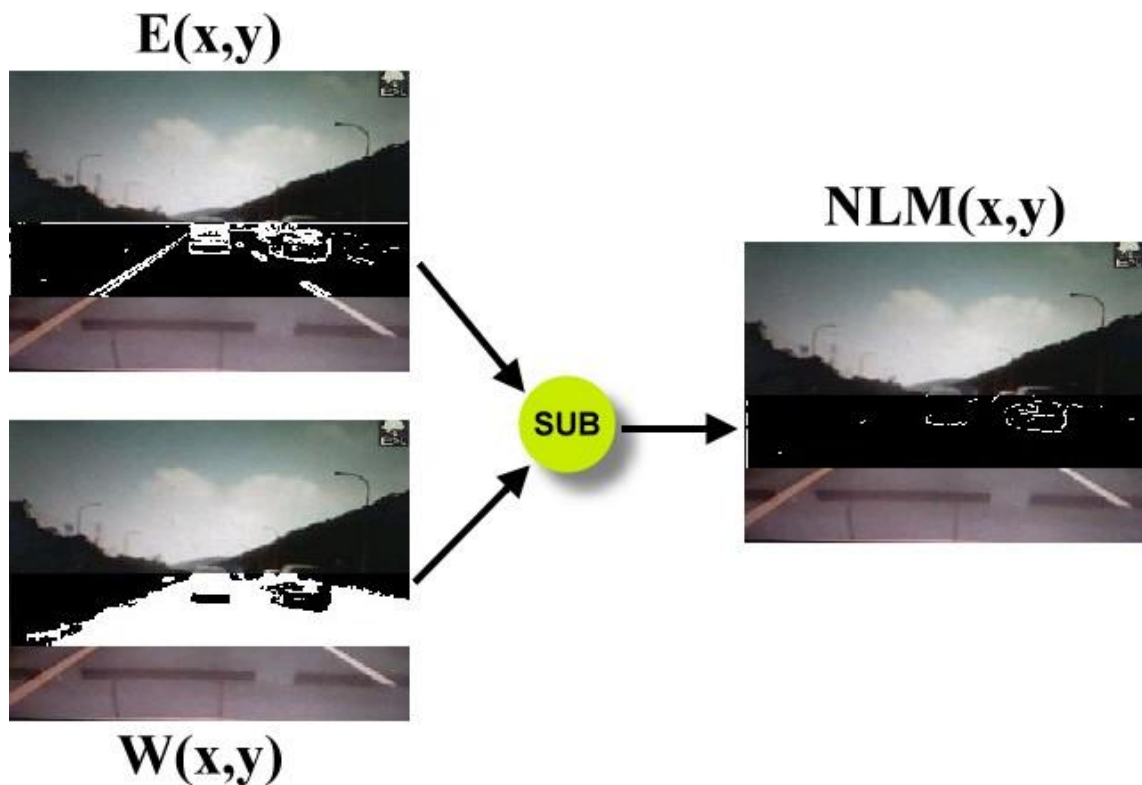


圖3.8 用相減運算濾除路面雜訊

### 3.4 標示車輛腳印(Located Vehicles Footprint)

在 3.3 節中，本論文使用影像相減的方式去除路面雜訊(車道標示線)，在這一章節本文利用車道線偵測，標示目前所行駛之車道區分前方與旁側車道車輛的腳印，其中在車道線偵測部份，亦透過車道標示線虛線與實線之特徵，判斷目前所在之車道位置，減少旁側車輛偵測錯誤。之後，透過投影分析標示前方車輛腳印。

#### 3.4.1. 車道偵測與車道遮罩

本論文使用 ROI 的灰階影像作為車道偵測之輸入處理，首先將 ROI 影像分為兩部份，分別是左部份 $A_l$ 與右部份 $A_r$ ，如圖 3.9 所示，接下來使用四十五度之邊緣子對 $A_l$ 與 $A_r$ 區域進行邊緣偵測，圖 3.10(a)為 $A_l$ 區域邊緣偵測所用四十五度之邊緣運算子，(b)則為 $A_r$ 所使用之四十五度之邊緣運算子。 $A_l$ 區域掃描方向為由下至上



由左往右， $A_l$ 區域掃描方向為由下往上由右至左，如圖 3.11 所示，經由邊緣偵測後我們可以得到以四十五度為基礎之邊緣強化影像。



圖3.9  $A_l$ 與 $A_r$ 區域之區分及邊緣偵測運算子掃描方向

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

(a) (b)

圖3.10 四十五度邊緣運算子(a)  $A_l$ 區域所用 (b)  $A_r$ 區域用

經過四十五度邊緣運算子處理之後，邊緣影像中存在著正負值，本文利用此特性設定門檻值找出正負邊緣，將正邊緣門檻值設定為正二十，負邊緣門檻值設定為負二十，並用綠色代表正邊緣，紅色為負邊緣。之後，由影像中心往左與右方向搜尋 $A_l$ 與 $A_r$ 區域，找到第一個正負邊緣之交會處，此處通常為車道線的位置，並且紀錄該位置之座標，其邊緣影像與搜尋方法如圖 3.11 所示。經由上述處理後

便可得到如圖 3.12 之邊緣點(Edge Point)影像。



圖3.11 四十五度基礎之邊緣影像與偵測邊緣交會點方法



圖3.12 利用正負邊緣交會之特性所產生的邊緣點影像

接下來本文利用同一車道線段邊緣點相鄰之特性，設定一個相鄰偏移門檻，將相同線段邊緣點群組找出各個車道標示線段，當相鄰像素之偏移值大於門檻值

則代表該相鄰像素屬於不同線段之邊緣點，如圖 3.13 所示，其中 $\Delta x$ 與 $\Delta y$ 代表 x 軸與 y 軸方向之偏移門檻。將每組串連之線段頭、尾位置用鏈結串列記錄，接著依照順序將所有串列依順序連接成左與右各一條的車道標示線，並且用不同顏色顯示不同線段，如圖 3.14 所示。

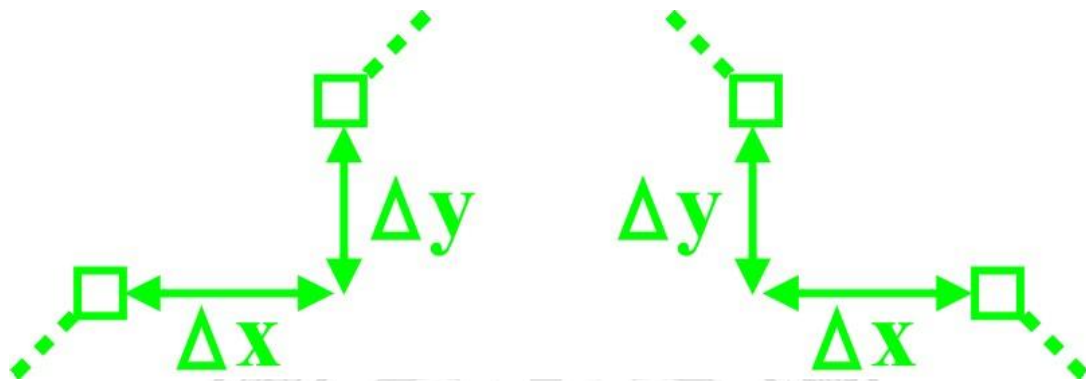


圖3.13 偏移門檻值 $\Delta x$ 與 $\Delta y$



圖3.14 相同線段使用相同顏色表示

找到每一條線段後，比較各個線段之 Y 軸截距，分別在左與右兩個串列中找尋最長的一條線段，並求出其斜率。利用式 3.3 求出兩線段無限延伸後之線段交會點(Vanishing Point)，又稱車道線段消失點。其中，a 為 Y 軸截距，b 為 X 軸截距，

c 為斜率，vy 為兩線段之交會點。

$$a = y_1 - y_2, b = x_2 - x_1, c = x_1y_2 - x_2y_1$$

$$vy = \frac{a_Lc_R - a_Rc_L}{a_Rb_L - a_Lb_R} \quad (3.3)$$

式(3.4)為直線方程式，主要使用在描繪車道標示線，當中 M 為斜率，I 為 Y 軸截距。由於使用式(3.4)不易繪製出車道區域遮罩，本文使用 Hough [20]轉換將座標空間轉換至參數空間，改寫成式(3.5)。車道遮罩建立方法是由影像中心向左右兩側搜尋，只需計算該像素 I 值是否與車道線的直線方程式經由 Hough 轉換之 I 值相同，即可知道是否達到車道線的邊緣位置。之後，在 ROI 中將未達到車道線邊緣之像素值填入 255，抵達車道線邊緣後，則將剩餘的像素值填入 0，圖 3.15 為車道遮罩產生結果。

$$y_i = Mx_i + I \quad (3.4)$$

$$I = -Mx_i + y_i \quad (3.5)$$



圖3.15 目前行駛車道遮罩影像

除了使用車道線偵測產生目前車道遮罩之外，另外有三個預設遮罩如圖 3.16 所示。當車道線偵測失敗，像是 Vanishing Point 超過地平線等狀況時，演算法就會使用預設的中央車道遮罩。左與右兩遮罩則直接使用在旁側兩車道上，取出旁側車道之車輛腳印。車輛腳印的擷取方法，是利用無車道線之影像與車道遮罩影像進行 AND 運算，取出該車道上可能為車輛腳印的影像，如圖 3.17 所示。



圖3.16 三個預設車道遮罩影像

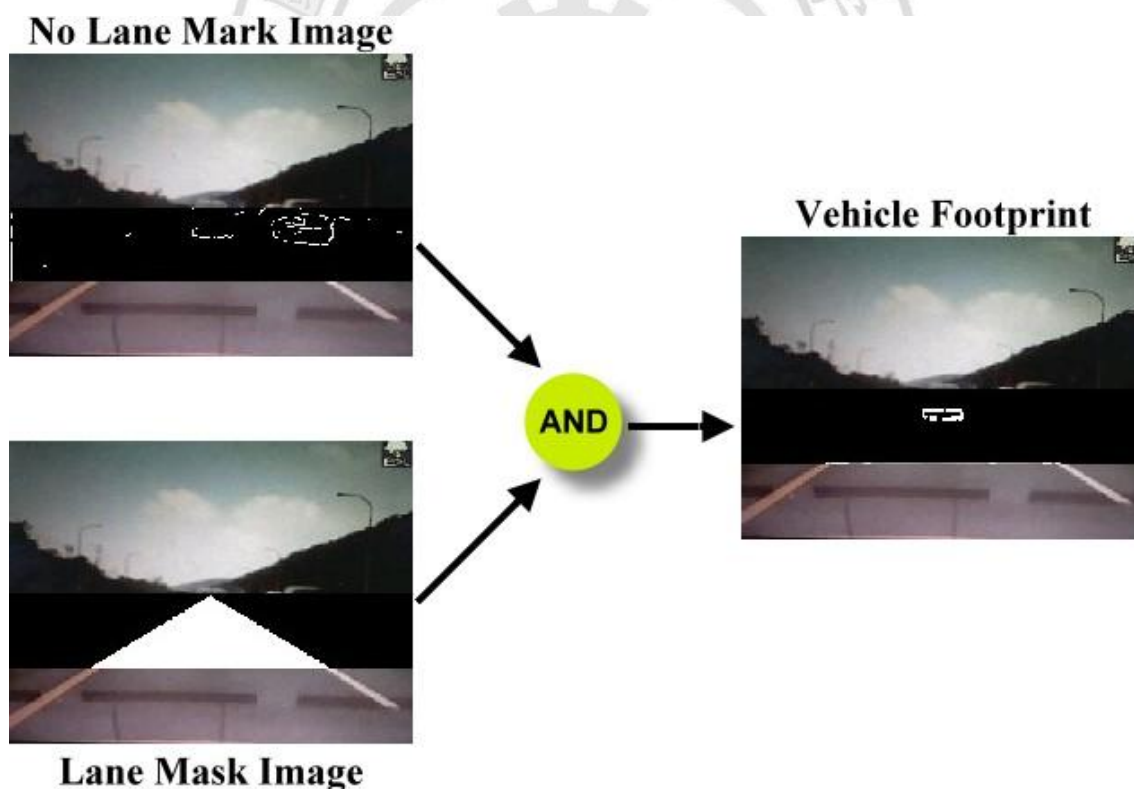


圖3.17 利用中央車道遮罩擷取車輛腳印

### 3.4.2. 車道判斷

台灣目前高速公路之車道數量不固定，本文以三線道為例，中央車道的左右車道標示線皆為虛線，而內側或外側車道則分別是左邊實線與右邊實線，因此可歸納出判斷目前車輛行駛於車道的準則。判斷車道線為虛線或實線方法有兩種，第一種是利用車道線線段的數量，但是由於本文處理的範圍在 ROI 中，虛線的車道線段的取樣較不充足，因此方法較不適用於本論文。第二種是判斷最長之車道線段長度，本論文判斷最長之車道線段是否大於二十二個像素，大於此則判斷此線段為實線，否則為虛線。判斷車道能幫助我們是否需要偵測左側或右側之車輛，省去偵測車輛與左右車道遮罩擷取旁側車輛腳印所耗費時間。

### 3.4.3. 車輛腳印標示

標示車輛的是利用二值化後的車輛底部資訊計算車輛在影像中的位置。本論文在中央車道(目前行駛車道)利用投影量分析方法搜尋標示出車輛的下邊界與左右邊界。由於中央車道的車輛是最主要的危險因素，因此使用較精確的方式進行偵測與標示。在旁側車道車輛的標示方法，本文使用較為精簡但不失正確性的方式偵測與標示車輛，降低系統的運算量。

中央車道車輛之腳印搜尋與標示方法，是先定義一個中央車道的搜尋範圍減少不必要的運算時間，搜尋範圍之定義如圖 3.18，其中 $Y_t$ 與 $Y_b$ 各自為 ROI 的上與下邊界， $X_l$ 與 $X_r$ 各自為 ROI 的左與右邊界，搜尋範圍的上、下邊界跟 ROI 相同，左與右邊界則定義在 ROI 左、右邊界各往內縮一百個像素值。車輛左與右邊界的搜尋，是將高亮度影像投影至 X 軸上，並找出投影量最高的兩個位置，即為車輛之左右邊界，圖 3.19(a)為統計 X 軸上投影量之示意圖。車輛底部邊界的搜尋，則是將高亮度影像投影至 Y 軸上，並找出投影量最高之位置，由於車底保留資訊較



車頂多，因此，此處即為車輛的底部邊界，圖 3.19(b)為統計 Y 軸上投影量之示意圖。圖 3.20 為實際上運作之方式。

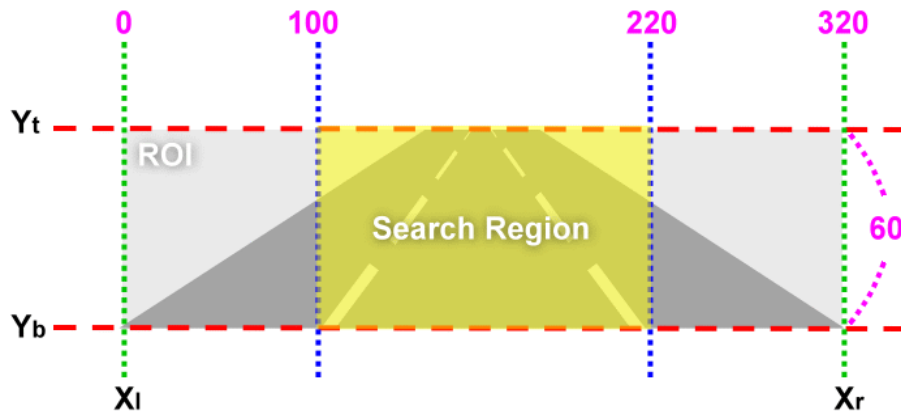


圖3.18 中央車道車輛搜尋之範圍

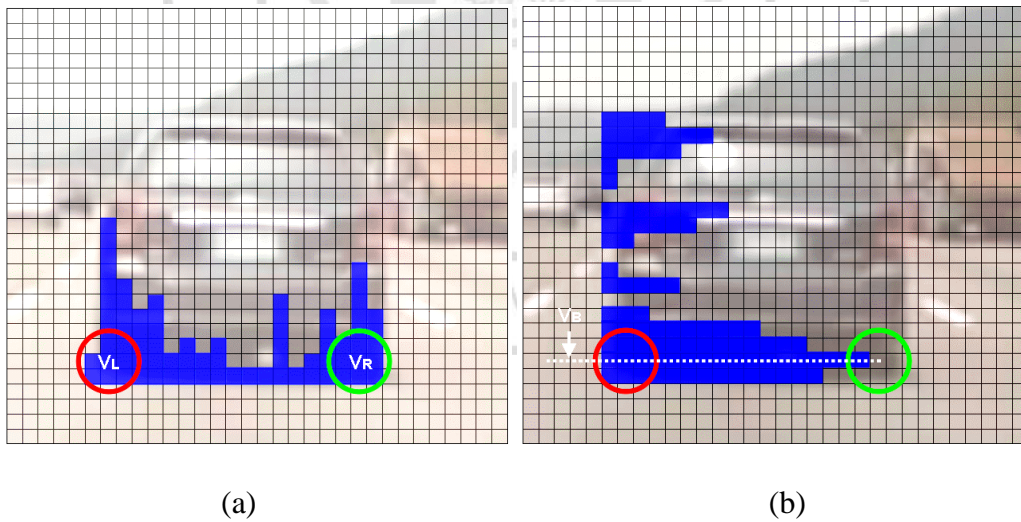


圖3.19 投影量統計示意圖 (a)統計 X 軸上投影量 (b)統計 Y 軸上投影量

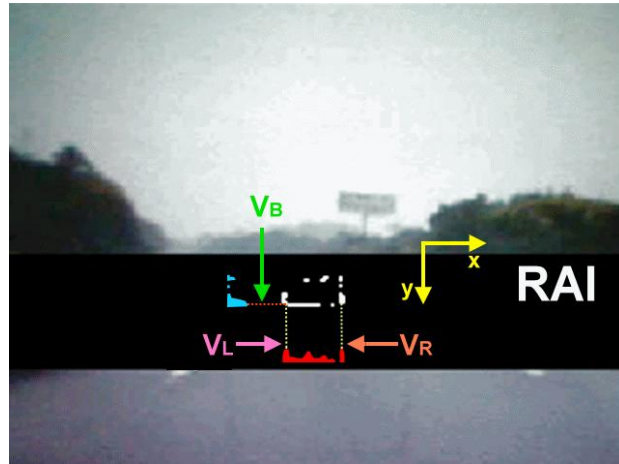


圖3.20 實際運作之方式

左、右側車道之車輛腳印標示，是找尋車輛內側的直角特徵，左側與右側車輛直角特徵各自為”└|”與”|└”。以左側車道為例，搜尋的方式是由影像的左下角為起始點，由左向右進行搜尋車輛的垂直特徵，直到預設終點即第一百二十個像素的位置。如果沒有找到車輛底部特徵，便將起始點向上移一個像素，繼續由左向右搜尋，如圖 3.21(a)所示。右側車道使用的方法與左側車道相同，不同的是搜尋方向與左側車道相反。搜尋中一旦找到第一個具有直角特徵的像素位置，系統即利用繪圖函數將該像素位置用紫色圓框標示出來。圖 3.21(b)為直角特徵之定義。

在本章節中，系統所處理之像素是已經過二值化處理的影像，而路面雜訊(車道標示線)也已濾除，並經過車道遮罩擷取車道內車輛底部的資訊與部份邊緣影像，因此使用投影分析方法可以得到不錯的車輛邊界標示結果。圖 3.22 為車輛腳印標示與車道偵測之結果，使用投影量分析(Projection Analysis)方法雖然較為耗時，但是十分適合使用在車輛腳印與邊界的搜尋。



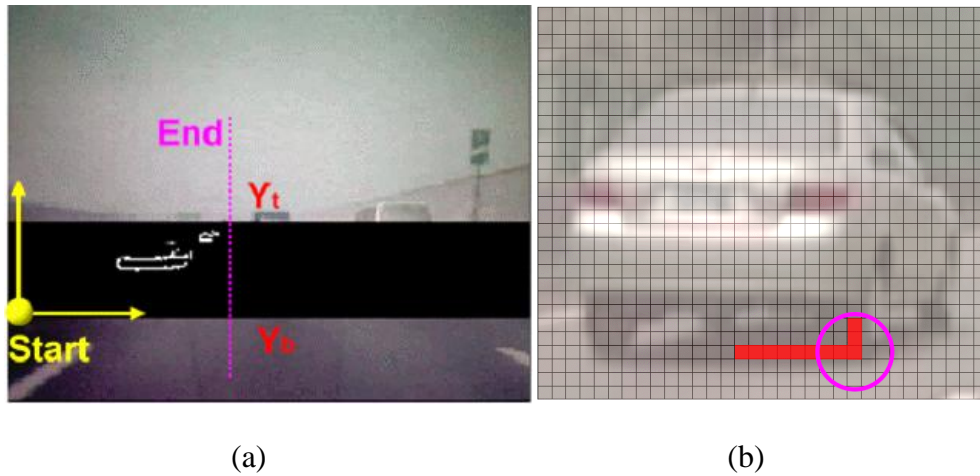


圖3.21 (a)左右車道車輛搜尋方法 (b) 搜尋寬高比為 3:1 直角特徵



圖3.22 中央與旁側車道的車輛腳印標示結果

### 3.5 車輛腳印驗證(Footprint Verification)

在此步驟中，本文利用車輛影像區域驗證標示之車輛腳印是否正確。首先，利用左右邊界差值求出車寬，並由底部邊界位置往上推算零點八倍車寬求得車輛上邊界，其中，一般小型車輛寬高比為 1：0.8，取得車輛影像區域之後，利用車輛左右對稱之特性作為驗證之方法。使用對稱性運算[8]檢驗車輛影像之對稱軸是否在已標示區域中心，如果對稱軸的偏移量在預設區域範圍內，則此區域影像則

屬車輛影像，否則取消該區域影像之邊界標示。本論文將文獻[8]的 RGB 對稱性運算式修改成式(3.6)，只計算 $Y_C, C_b$ 中的 Y，以降低系統之運算量。

$$S(j) = \sum_{i=V_B}^{V_T} \sum_{\Delta x=1}^{W/2} \sum_{j=V_L-\Delta k}^{V_R+\Delta k} |p(j + \Delta x, i) - p(j - \Delta x, i)| \quad (3.6)$$

$$J_{sym} = \min S(j) \text{ and } S(j) < S_{th}$$

其中， $V_T$ 為車輛上邊界，是由車底往上推算零點八倍所求得， $V_B$ 為車輛底邊界， $V_L$ 為車輛左邊界， $V_R$ 為車輛右邊界，這三個邊界在 3.43 節中就已得知。 $W$ 為車輛寬度，由左右邊界相減求得。 $\Delta x$ 為計算用偏移量，其計算偏移範圍在 1 至  $W/2$ 。 $j$ 與  $i$ 分別為 X 軸與 Y 軸之索引值， $\Delta k$ 為  $j$  往左右方向的擴展值。

圖 3.23 為經過對稱運算後顯示對稱中心的結果，在確認對稱軸位於車輛影像之中心區域後，用深藍色的邊界盒(Bounding Box)將車輛邊界標示。



圖3.23 車輛影像對稱軸與邊界盒

### 3.6 車輛追蹤程序(Tracking Process)

本論文在日間車輛偵測演算法中，使用光流法(Optical Flow)[9]與絕對值平均誤差法(Mean Absolute Error, MAE)實現車輛追蹤之方法。透過偵測小範圍影像區域之變化量，調整車輛邊界盒的寬與高，藉由此方法降低系統整體運算量。圖 3.24 為車輛追蹤程序之流程圖。

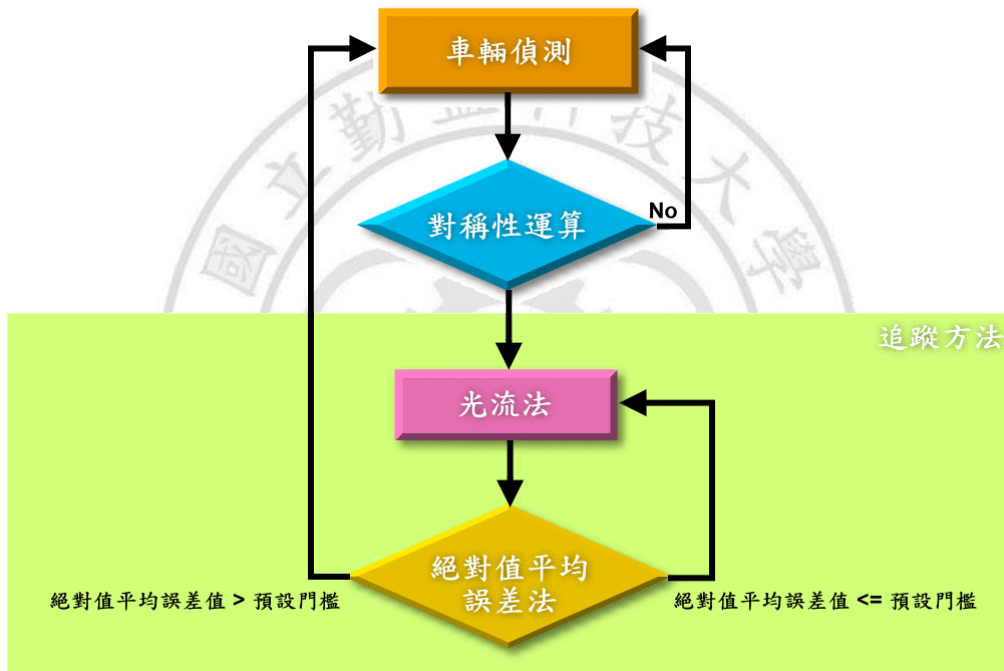


圖3.24 車輛追蹤與車輛偵測之關係

在執行車輛追蹤之前，車輛已透過一連串之影像處理程序給偵測與標示在影像中。本論文利用光流法動態微調車輛邊界盒，並在微調之後使用 MAE 計算出影像變化的差異值，若差異值過大則代表車輛之間的相對速度有所改變，超出了追蹤範圍，系統則必須重新偵測車輛找尋新的追蹤範圍，反之，則繼續進行追蹤，直到差異值過大或追蹤次數大於系統預設的追蹤次數(十五次)。

## 第四章 夜間車輛偵測

我們觀察在夜間情況下的影像中，發現許多在日間情況下可見的車輛特徵都因周圍的照度不足而消失，像是車輛陰影與直角特徵等，因此無法使用日間車輛偵測方法搜尋前方車輛。在夜間車輛偵測中，主要的工作於在昏暗的環境與動態影像中偵測出前方車輛。以下將詳細的介紹夜間環境中車輛偵測之程序。

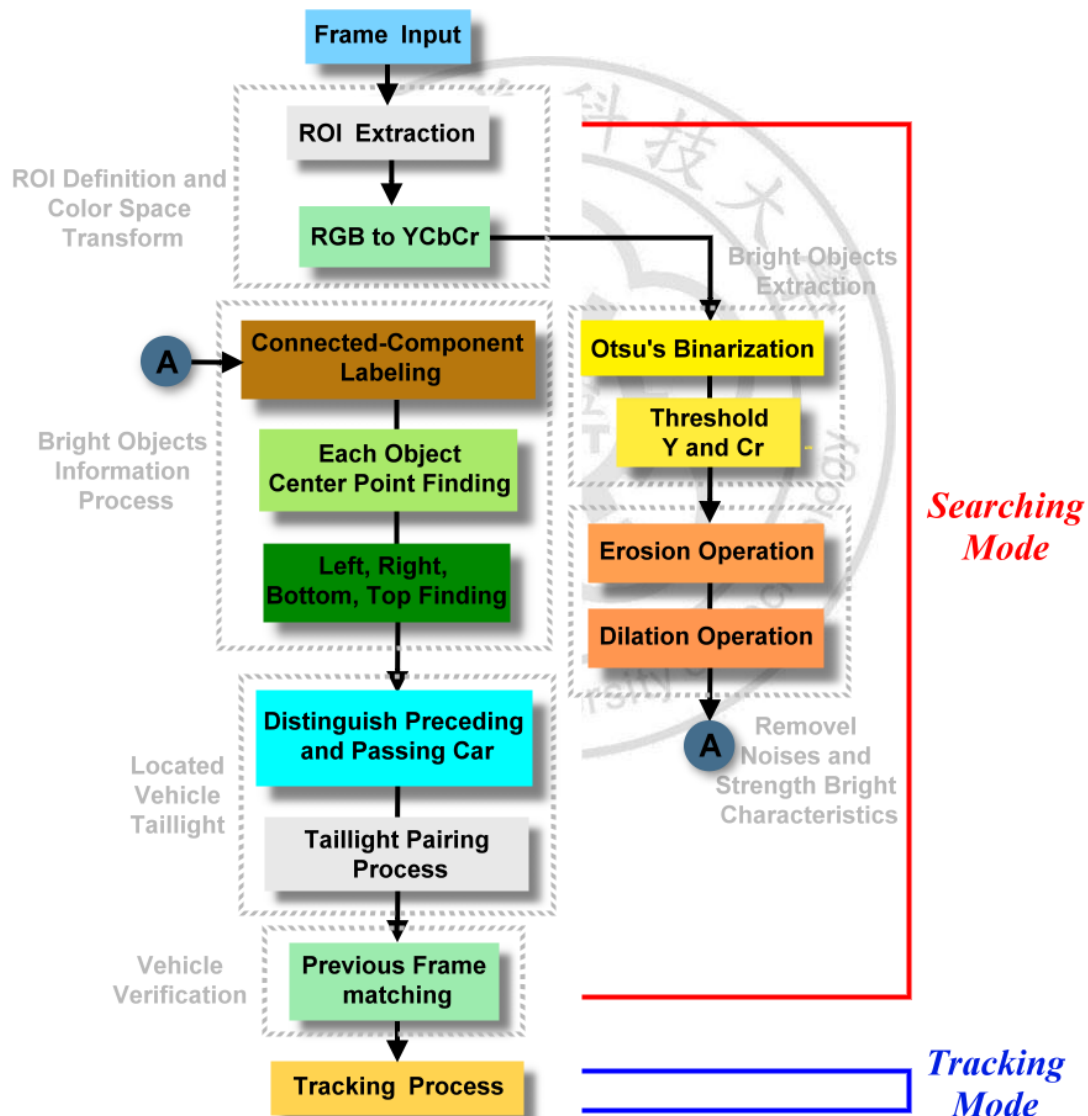


圖4.1 夜間車輛偵測演算法執行之程序

當黑夜籠罩大地時，影像中唯一可判別的車輛特徵為車輛後方的成對車尾燈。為了在動態影像裡擷取車輛車尾燈之特徵，本文設計一個夜間車輛偵測演算法，如圖 4.1 所示。使用多階層的影像處理方法將前方車輛偵測並標示在影像中，演算法中分為搜尋與追蹤模式，搜尋模式用於搜尋車輛之成對車尾燈，追蹤模式則是為了讓演算法能在硬體資源有限的嵌入式系統上執行所做的設計。搜尋模式裡共由六大步驟所組成，分別是：(1)感興趣區域的定義與色彩空間之轉換，主要是為了降低系統運算量與減少不必要的搜尋動作。(2)亮點物件的擷取，把可能為車尾燈特徵的影像物件擷取，分離前景與背景。(3)移除雜訊與強化特徵，把強度不足之亮點物件移除，並強化其餘的亮點物件，使其特徵更易於偵測。(4)亮點物件資訊處理，將各個亮點物件影像轉換成獨立影像物件，並計算每一個被分割亮點物件的相關資訊(如寬、高等)。(5)框選車輛車尾燈，先分出不同車道的車輛車尾燈(圖 4.26 將介紹其流程)，並將各個車道的車輛車尾燈進行配對，找出車輛成對的車尾燈。(6)驗證車輛影像，利用前一張影像與下一張影像配對的結果進行比對，如比對成功則將此一配對結果保留，完成驗證。透過一連串的影像處理、配對與驗證程序無非就是為了讓偵測結果達到最高的正確率，但如此將會花上較多的系統處理時間，為了讓系統達到即時性，因此本文另外提出追蹤模式來降低系統運算量，讓系統達到即時性與準確性兼顧的雙贏狀態。以下將會介紹各個演算法中的處理步驟。

#### **4.1 感興趣區域的定義與色彩空間之轉換(ROI Definition and Color Space Transformation)**

如同日間偵測演算法，在搜尋車輛之前，本文定義一個感興趣區域(Region of interest, ROI)。如此可以隔離許多干擾因素(如路燈、交通號誌等)，減少搜尋的範

圍，增加演算法執行的效率。本文定義一個虛擬地平線也就是 ROI 的上邊界，由於影像輸入之大小為 320 x 240，其上邊界定義在 Y 軸座標一百二十的位置，下邊界為上邊界往下推移六十個像素值，其位置與駕駛者車輛前方五公尺之處相對應，此處在正常駕駛情況下，不會有車輛出現，ROI 之定義如圖 4.2 所示。之後的演算法程序一律只處理 ROI 中的影像資訊。

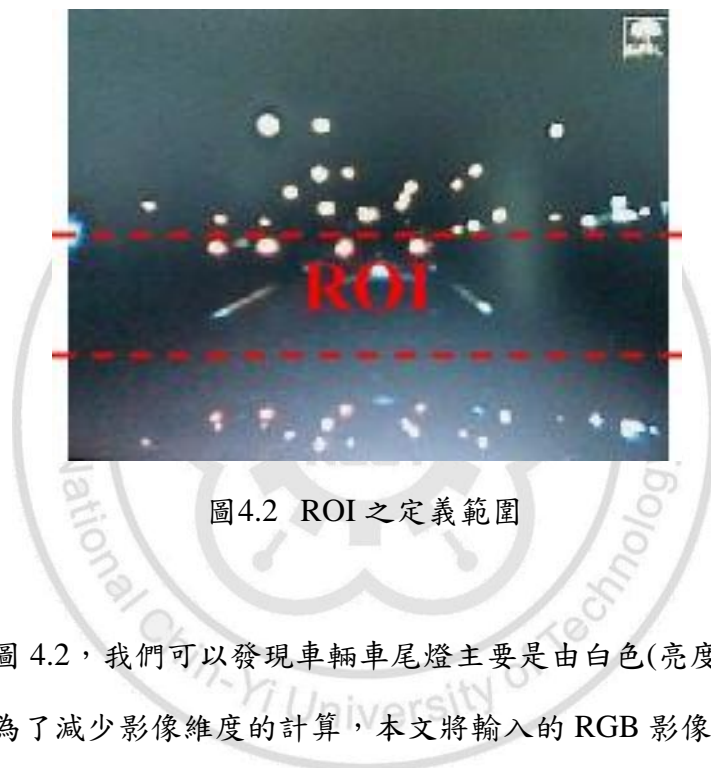


圖4.2 ROI之定義範圍

經由觀察圖 4.2，我們可以發現車輛車尾燈主要是由白色(亮度)與紅色色度所組成的，並且為了減少影像維度的計算，本文將輸入的 RGB 影像轉換成  $Y C_r C_b$  色彩空間[20]中的 Y(亮度)與  $C_r$ (紅色色度)，其轉換的計算如式(4.1)所示。圖 4.3 為以亮度為基礎的灰階影像，圖 4.4 則是以紅色色度為基礎之灰階影像，可以發現具有紅色特徵的車燈在此張影像中是屬於較高亮度之影像。

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & 0.0813 \\ -0.1687 & -0.3313 & 0.5 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (4.1)$$



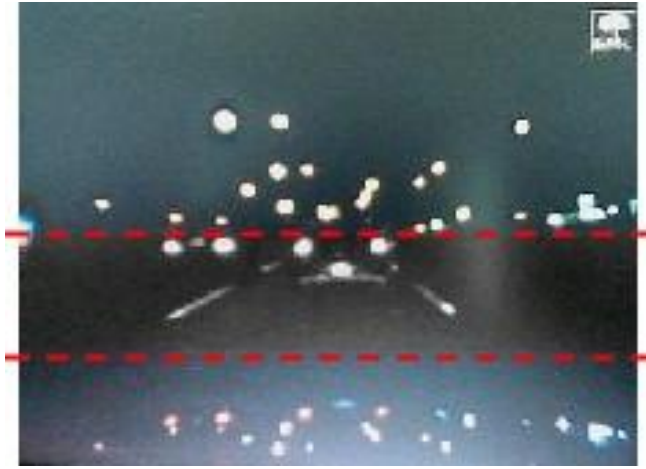


圖4.3 以 Y(亮度)為基礎之灰階影像

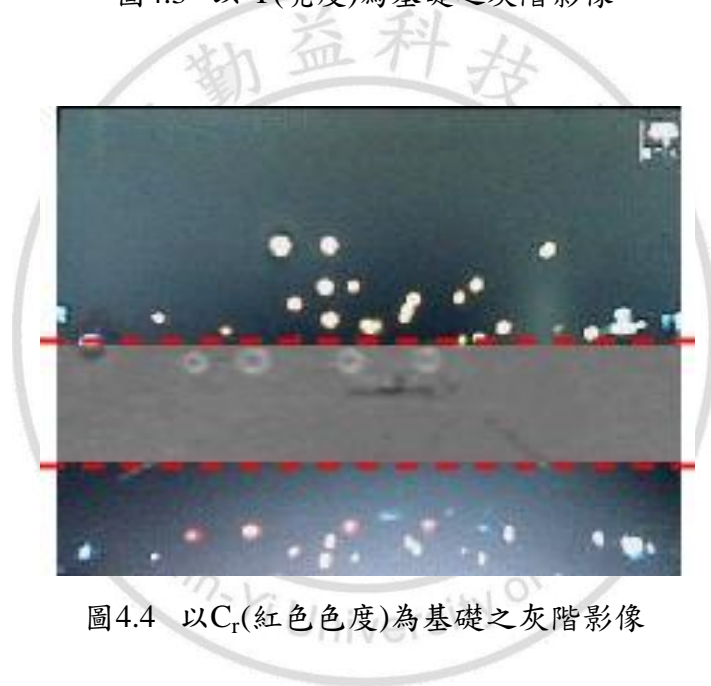


圖4.4 以  $C_r$ (紅色色度)為基礎之灰階影像

## 4.2 亮點物件的擷取(Bright Objects Extraction)

本文使用二值化的方式擷取影像中的亮點物件，也就是可能為車輛車尾燈之影像。不難發現，車尾燈的組成主要是亮度與紅色色度，利用此特性對 Y 與  $C_r$  設定門檻值，進行二值化。本論文使用 Otsu's Method[21]找尋背景不斷變動的影像中最合適的 Y 門檻值。Otsu 提出的門檻值設定方法，利用影像中灰階值出現的機率與分佈計算變異量，找出分隔前景與背景最佳門檻值，以下將介紹 Otsu's



Method 找出最佳門檻值之程序。

首先定義影像中灰階值範圍為 1 至 L 也就是  $[1, 2, \dots, L]$ ,  $i$  則為灰階值的索引,  $N$  為影像中像素的總數量,  $n_i$  為  $i$  灰階值之像素的數量,  $P_i$  為  $i$  灰階值的像素在影像中出現的機率, 如式(4.2)所示。以本論文灰階影像為例,  $L$  等於 256,  $N$  為  $320 \times 240$ 。

$$P_i = \frac{n_i}{N}, \quad P_i \geq, \quad 0 \sum_{i=1}^L P_i \quad (4.2)$$

接下來式(4.3)至(4.7)中, 設定一  $k$  值將灰階值範圍分為兩個群組, 這兩群組可視為我們所要分離的前景與背景, 令  $C_0$  代表灰階值 1 至  $k$  的群組,  $C_1$  則代表灰階值  $k+1$  至  $L$  的群組, 而  $\omega_0$  與  $\omega_1$  分別代表  $C_0$  與  $C_1$  的出現機率,  $\mu_0$  與  $\mu_1$  則各別代表  $C_0$  與  $C_1$  的平均值, 其中  $\mu_T$  為原影像之總平均值(Total Mean Level)。

$$\omega_0 = Pr(C_0) = \sum_{i=1}^k P_i = \omega(k) \quad (4.3)$$

$$\omega_1 = Pr(C_1) = \sum_{i=k+1}^L P_i = 1 - \omega(k) \quad (4.4)$$

$$\mu_0 = \sum_{i=1}^k iPr(i|C_0) = \sum_{i=1}^k \frac{P_i}{\omega_0} = \frac{\mu(k)}{\omega(k)} \quad (4.5)$$

$$\mu_1 = \sum_{i=k+1}^L iPr(i|C_1) = \sum_{i=k+1}^L \frac{iP_i}{\omega_1} = \frac{\mu_T - \mu(k)}{1 - \omega(k)} \quad (4.6)$$

$$\omega(k) = \sum_{i=1}^k P_i, \quad \mu(k) = \sum_{i=1}^k iP_i, \quad \mu_T = \mu(L) = \sum_{i=1}^L iP_i \quad (4.7)$$

再利用式(4.8)、(4.9)與(4.10)計算各群組變異量與變異量總和。

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 Pr(i|C_0) = \sum_{i=1}^k (i - \mu_0)^2 \frac{P_i}{\omega_0} \quad (4.8)$$

$$\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 Pr(i|C_1) = \sum_{i=k+1}^L (i - \mu_1)^2 \frac{P_i}{\omega_1} \quad (4.9)$$

$$\sigma_w^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 \quad (4.10)$$

最後，我們利用不同 k 值計算出  $\sigma_w^2$ ，並找出最小  $\sigma_w^2$  值所對應的 k 值，此 k 值即為經由 Otsu's Method 所求出的最佳適應性門檻值，圖 4.5 為用門檻值 k 所分出的兩個群組的灰階分佈示意圖，圖 4.6 為二值化後試示意圖。圖 4.7 則為影像經由找尋出來的最佳門檻值對 Y 空間影像二值化結果。

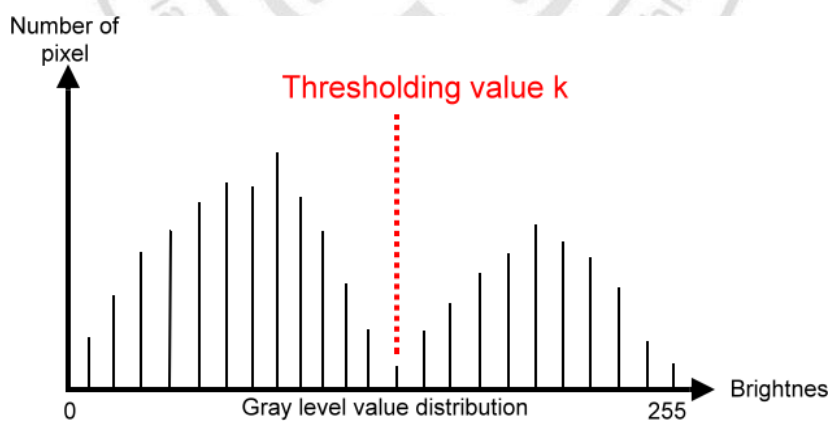


圖4.5 運用 k 值分出兩群組的灰階分佈示意圖

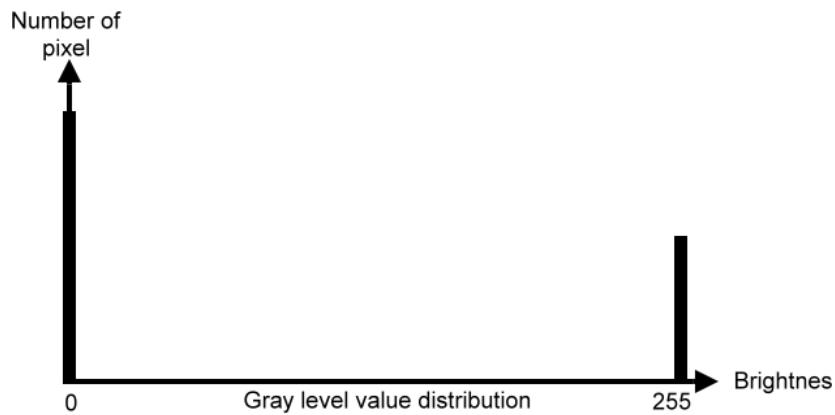


圖4.6 經過二值化後影像灰階分佈情況



圖4.7 對 Y 空間影像取門檻值二值化之結果

觀察圖 4.7 可以發現，車尾燈影像的亮點物件已被擷取出來，但有許多非車尾燈的影像物件(例如：車道線、車牌與車身反光)存在著。因此本文加入 $C_r$ 空間影像進行二值化動作，消除非車尾燈影像之物件，式(4.11)為加入 $C_r$ 參數後二值化的條件判斷式。其中 $Y_{th}$ 是由 Otsu's Method 所計算求得， $C_{rth}$ 則是經由我們的觀察與實驗所自行定義出來的經驗值為， $B(i,j)$ 為二值化影像之像素值，在本論文中將 $C_{rth}$ 值定義為 136，圖 4.8 為對 Y 與 $C_r$ 取門檻值二值化之結果。

$$B(i,j) = \begin{cases} 255, & \text{if } Y \geq Y_{th} \text{ and } C_r \geq C_{r_{th}} \\ 0, & \text{otherwise} \end{cases} \quad (4.11)$$



圖4.8 對 Y 空間與 $C_r$ 空間影像取門檻值二值化之結果

### 4.3 移除雜訊與強化特徵(Removal Noises and Strength

#### Bright Characteristics)

圖 4.8 中，幾乎只剩下車尾燈影像的亮點物件。但仔細觀察後，可以發現仍有些許雜訊殘留在影像中，且亮點物件影像有部份破碎，導致特徵強度不足。為解決此問題，本文使用影像處理中型態學的斷開運算(Opening Operation)[20]對二值化影像進行處理。

型態學中的斷開運算主要是由膨脹(Dilation)和侵蝕(Erosion)運算組成。膨脹運算主要是用於橋接縫隙，修補破碎影像。當 A 和 B 為 $Z^2$ 中的兩個集合，A 藉由 B 膨脹則定義如式(4.12)。

$$A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \varnothing \right\} \quad (4.12)$$

其中， $B$  對於原點的反射並對此反射平移  $z$  單位。 $z$  的所有移位之集合為  $A$  藉由  $B$  膨脹所產生，在膨脹以及其他形態學運算上的結構元素通常是定義為集合  $B$ ，根據前述解釋可將重新定義式(4.12)為式(4.13)。

$$A \oplus B = \left\{ z \mid [(\hat{B})_z \cap A] \subseteq A \right\} \quad (4.13)$$

如此定義式(4.13)能讓它更符合在二維空間中執行影像處理的概念。雖然膨脹是基於集合運算，但是當結構元素  $B$  視為迴旋積(Convolution)遮罩並建立於算術運算時，將  $B$  對其原點翻轉並依序位移滑過集合  $A$  上方，如此就如同將一個遮罩在影像中滑動計算。本文藉由圖示 4.9 與 4.10 顯示膨脹運算的處理程序及結果，圖 4.9 為正方形結構元素對原始影像進行膨脹運算的結果，圖 4.10 為矩型結構元素之膨脹運算，其中  $A$  為原始影像  $B$  為結構元素。

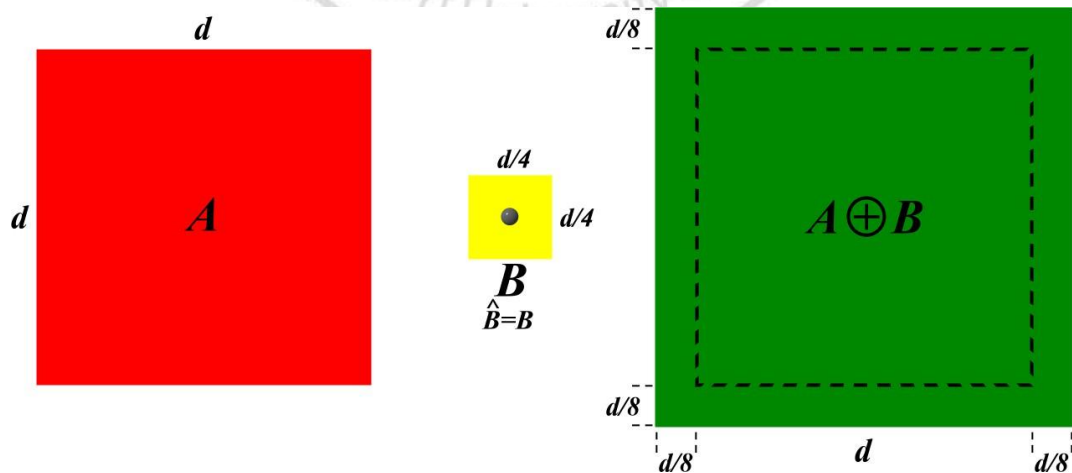


圖4.9 正方形結構元素對原始影像進行膨脹運算的結果

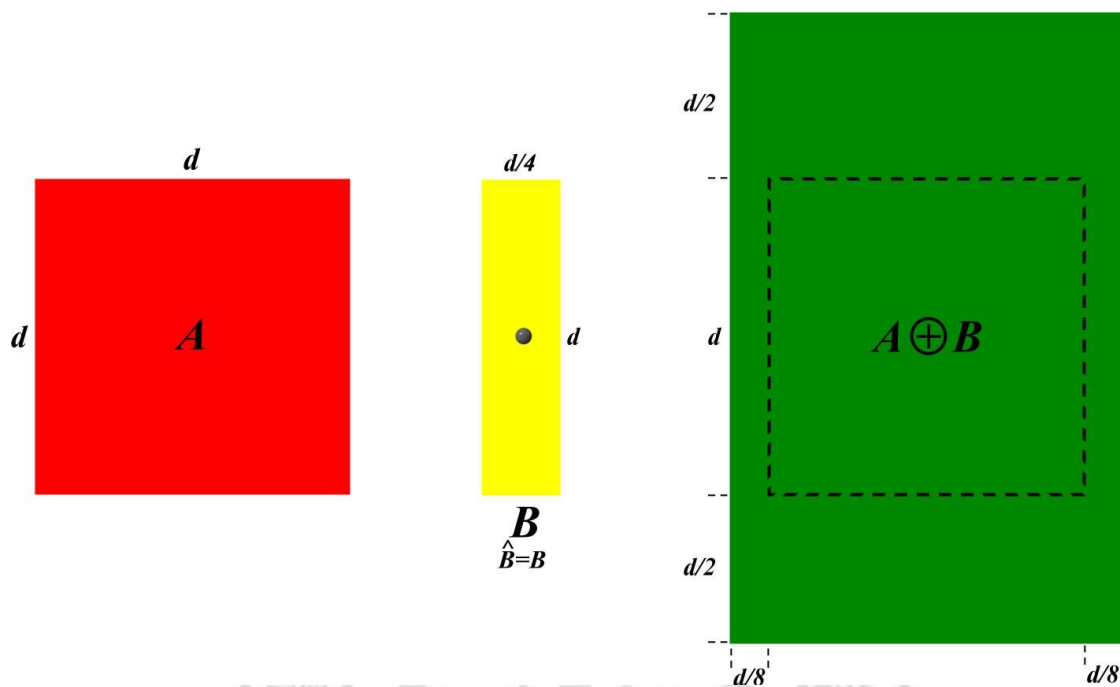


圖4.10 矩型結構元素對原始影像進行膨脹運算的結果

二值型態學中的侵蝕運算主要應用功能在於除去二值化影像裡不需要的雜訊，雜訊是依影像中物件的大小而定義。侵蝕運算定義集合  $A$  與  $B$  在  $Z^2$  中， $A$  被  $B$  侵蝕，則就記為  $A \ominus B$ ，如式(4.14)所示。此式也可以解釋為當  $A$  被  $B$  的結構元素侵蝕，使得  $B$  位移  $z$  後仍包含在  $A$  所有  $z$  點的集合，如此說法更貼近二維影像的處理。圖 4.11 與 4.12 為侵蝕運算的一個過程與結果，圖 4.11 中使用一個正方形結構元素  $B$  對集合  $A$  進行侵蝕，集合  $A$  被侵蝕部份，為虛線內白色之區域，虛線代表  $B$  的移動極限，如  $B$  的原點超出此虛線，這一集合就不再包含在集合  $A$  中，也就是違反了定義。圖 4.12 為使用拉長的結構元素對集合  $A$  侵蝕，值得注意的是原始的集合  $A$  被侵蝕成一條線。

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (4.14)$$

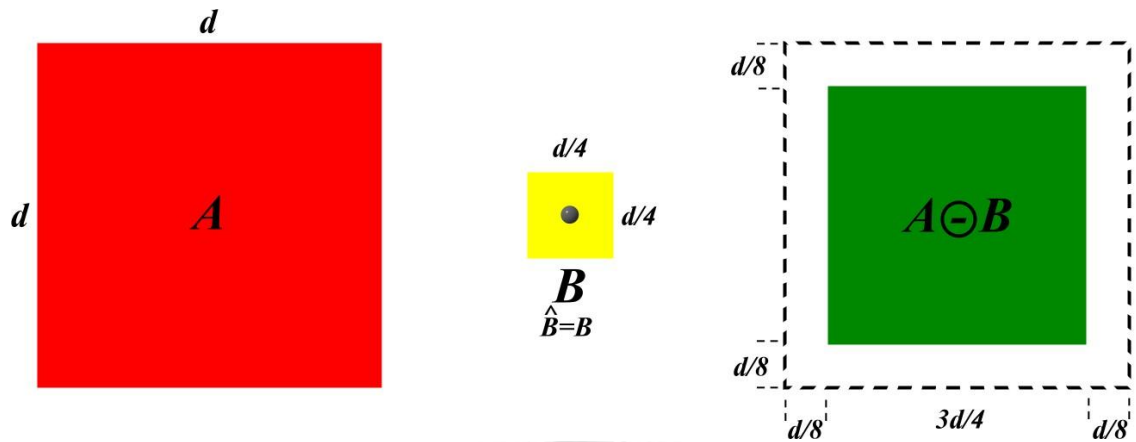


圖4.11 正方形結構元素對原始影像進行侵蝕運算的結果

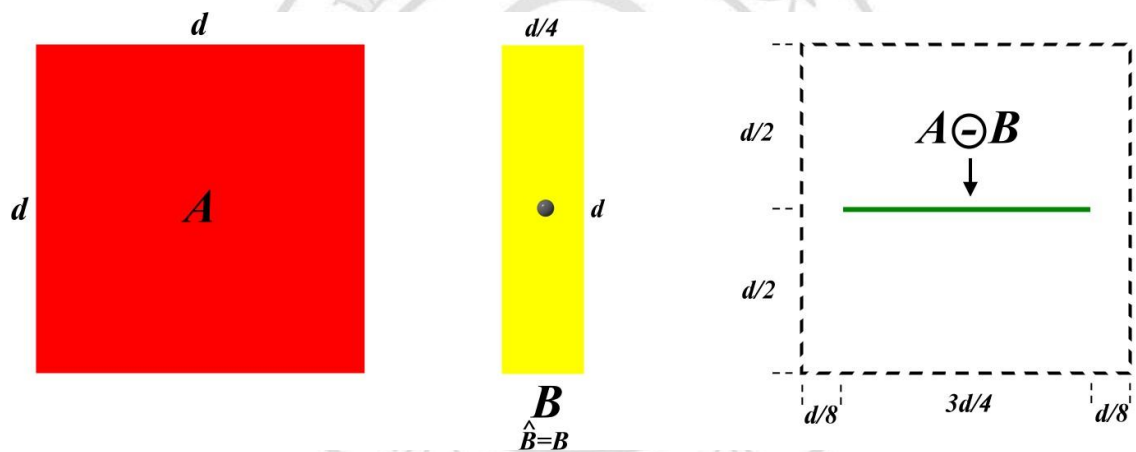


圖4.12 矩型結構元素對原始影像進行侵蝕運算的結果

如上述所介紹，膨脹運算是擴張一張影像，相反的，侵蝕運算則是讓影像收縮，把這膨脹與侵蝕型態運算搭配使用可以產生斷開(Opening)與閉合(Closing)這兩種重要的型態運算。斷開運算常用於讓物體輪廓平滑，消除窄細及微小的影像，閉合運算也是使輪廓部份平滑，不過跟斷開運算不同之處，在於它是把窄小之縫隙連接與修補影像空洞。本論文僅使用到斷開運算進行影像之處理，因此閉合運算的細節將不多加敘述，斷開運算可被定義為式(4.15)。



$$A \circ B = (A \ominus B) \oplus B \quad (4.15)$$

式(4.15)中，結構元素  $B$  將集合  $A$  斷開，記為  $A \circ B$ 。集合  $A$  被  $B$  斷開可以解釋為  $A$  被  $B$  侵蝕，再將結果使用  $B$  膨脹。為了清楚說明斷開運算，利用圖 4.13 簡單做一個幾何的解釋。把一個結構元素  $B$  作為一個在二維空間移動的球，而  $B$  圍繞在  $A \circ B$  的邊界內， $A \circ B$  邊界為  $B$  結構元素中的原點所達到最遠之  $A$  的邊界。透過上述幾何擬合(fitting)特性，要得到  $A$  和  $B$  的斷開，可以取與  $A$  擬合之  $B$  所有平移的集合。如此，可以產生一個以集合論為基礎的公式定義斷開運算，如式(4.16)。

$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\} \quad (4.16)$$

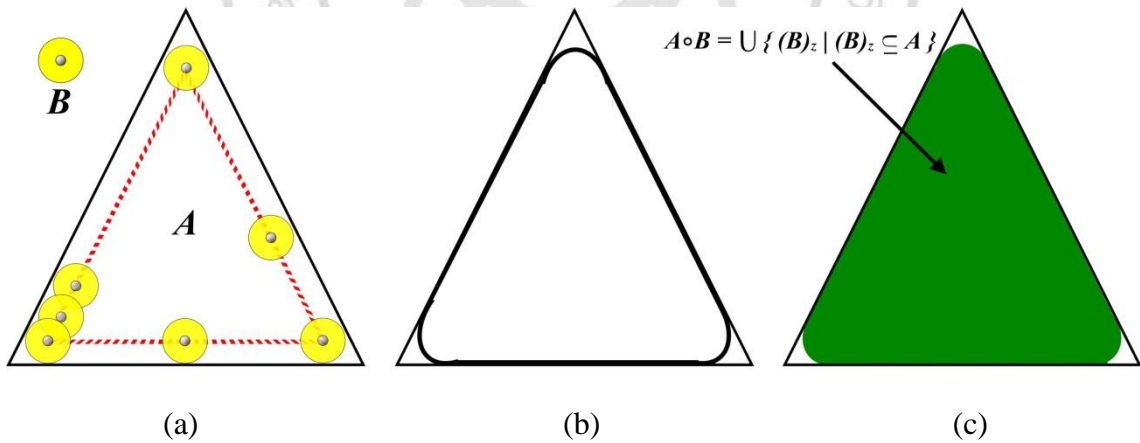
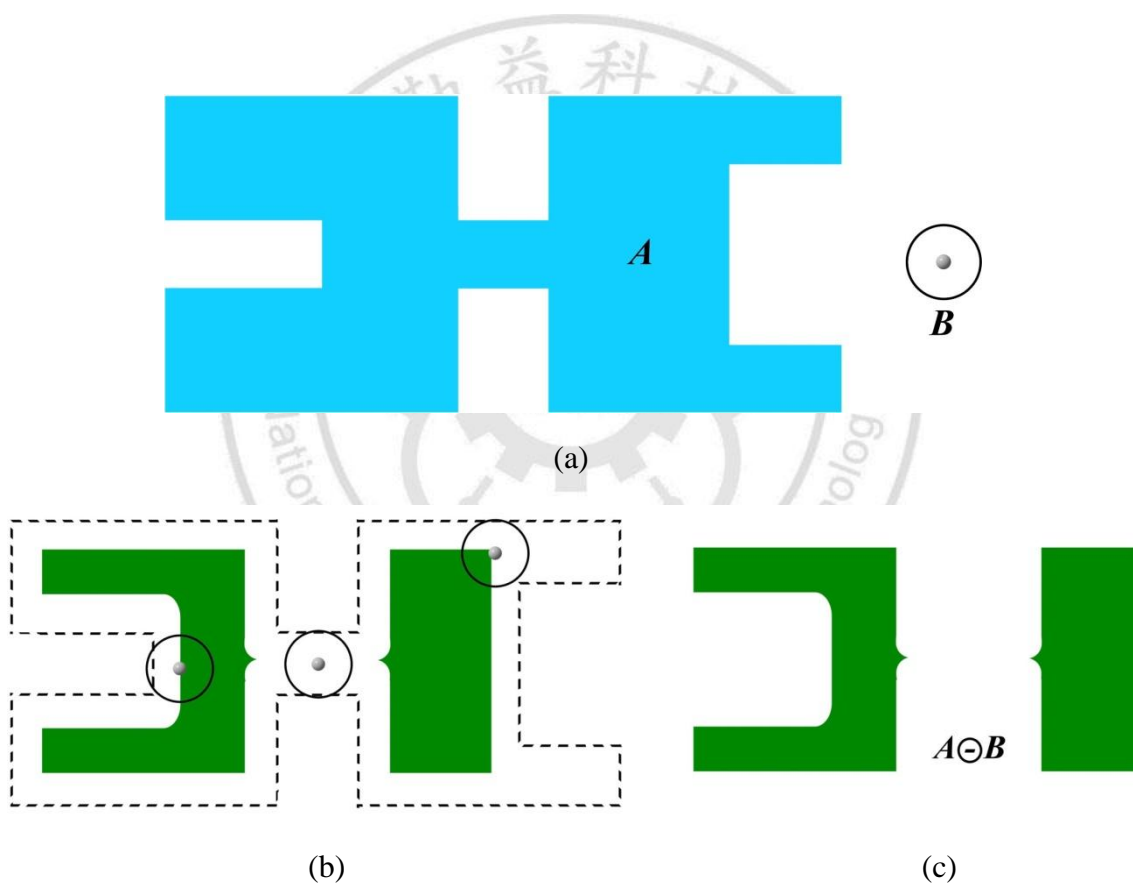


圖4.13 幾何解釋斷開運算 (a)結構元素  $B$  在  $A$  裡移動，並且不超出  $A$  的邊界 (b) 粗線代表斷開的外邊界 (c)完整之斷開

圖 4.14 則是以二維影像為例子，它進一步的說明斷開運算。圖 4.14(a)中為一

個集合  $A$  與結構元素  $B$ ，圖 4.14(b) 顯示圓形結構元素正侵蝕集合  $A$  的各處，圖 4.14(c) 為侵蝕運算執行完的結果，將原本相連的圖形分開。值得注意的是連接圖形的細長矩形寬度比結構元素的直徑小，代表結構元素無法完全在此部份集合中，如此便與式(4.14)條件不符。而右邊兩條凸出之矩形也是一樣。總之，與結構元素沒有擬合之元素均被消除。圖 4.14(d) 則是將已被侵蝕之集合進行膨脹運算的處理過程。圖 4.14(e) 則是斷開運算完之結果，可以發現影像物件輪廓部份平滑，而窄細的物件被消除。



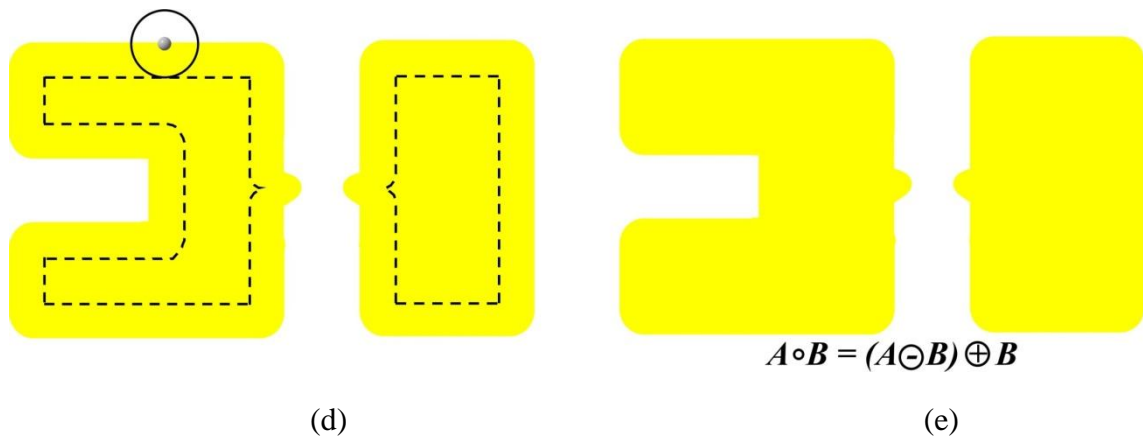


圖4.14 斷開運算的圖示說明 (a)集合 A 與結構元素 B (b)集合 A 被侵蝕之過程 (c)侵蝕運算完結果 (d)對已侵蝕完集合膨脹過程 (e)斷開運算之結果

本論文將圖 4.8 進行二值化型態學中的斷開運算，主要目的在消除微小雜訊與增強、修補車尾燈特徵的影像物件。圖 4.15 為將圖 4.8 進行侵蝕運算之結果，圖 4.16 則為圖 4.8 執行完斷開運算之結果。



圖4.15 把圖 4.8 進行侵蝕運算之結果



圖4.16 把圖 4.8 進行斷開運算之結果

## 4.4 亮點物件資訊處理(Bright Objects Information Processing)

接著，為了後續的車尾燈配對程序，必須把各個亮點物件影像分割，並計算其資訊。本文使用相鄰元素編號演算法(Connected-Component Labeling)[20]將影像中的亮點物件分割成單一物件，並予以編號，並求出各個被分割亮點物件之中心點與上、下、左、右的邊界位置。

### 4.4.1 影像分割

影像分割通常是將二值化影像，使用相鄰元素編號把各個高亮度像素進行編號，把相鄰的高亮度像素群組。然而由實作面的方向來看，可以分成遞迴與非遞迴兩種實現相鄰元素編號的方法。以下為兩種方法實現相鄰元素編號演算法的步驟。其中，根據像素之間的相鄰性定義為(1)4-鄰接(2)8-鄰接，如圖 4.17。本論文皆以 8-鄰接方式實現與說明。

Neighbors:

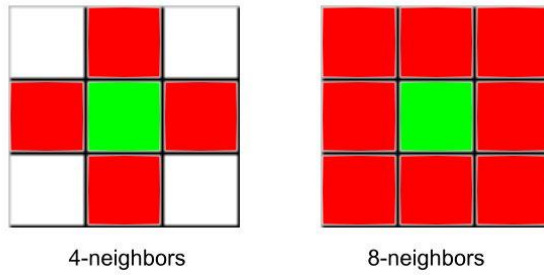


圖4.17 像素之間相鄰性之定義

## A 遞迴

**Step1:**將影像由左至右，上至下方式掃瞄，若此高亮度像素未被標記過，則將此高亮度像素標記新號碼，號碼值通常是由小至大。如圖 4.18 所示

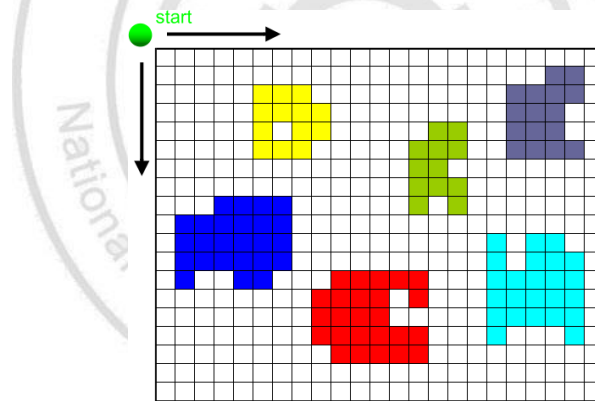


圖4.18 由左至右，由上至下對影像掃瞄

**Step2:**接著利用遞迴方式搜尋該高亮度像素是否有相鄰的高亮度像素，若有則標記相同號碼，否則跳出此遞迴。如圖 4.19 所示

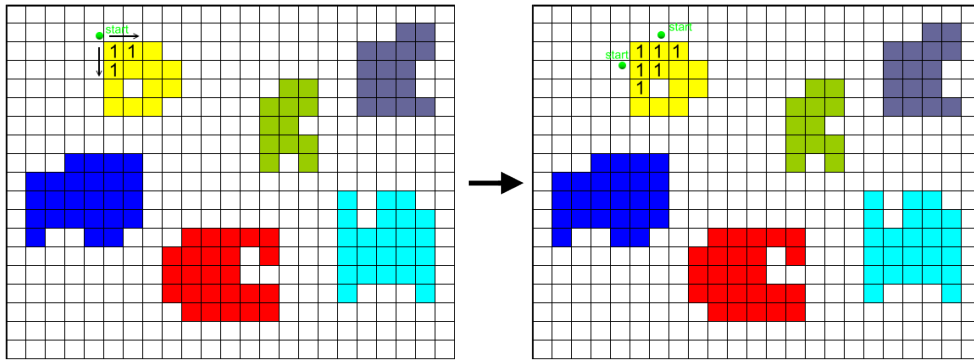


圖4.19 標記後對該標記點進行相鄰元素遞迴偵測

**Step3:**重複 Step1 及 Step2 直到影像掃描結束。圖 4.20 為執行完之結果。

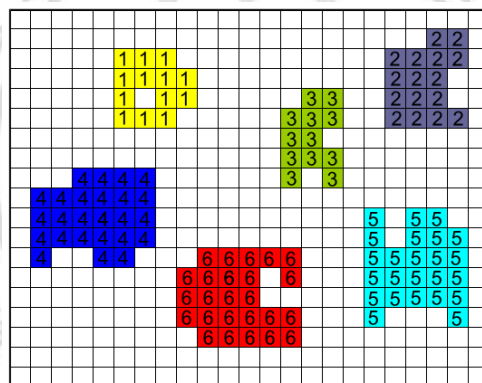


圖4.20 各高亮度區塊標記結果

## B 非遞迴

**Step1:**進行第一次掃描，由上至下，左至右的檢查影像中每一個像素。

**Step2:**找到未給予編號之高亮度像素時，會有三種情況判斷。(1)原像素周圍無相鄰像素或相鄰像素無編號，則給予原像素新編號。(2)原像素周圍存在一個或多個相鄰像素且相鄰像素皆為相同編號，則給予相鄰像素所屬之編號。(3)原像素周圍存在多個相鄰像素且相鄰像素中有數個不同編號，

則建立一個表格紀錄這些編號為等價，並暫時把相鄰像素中最小的編號給予原像素。其中，搜尋相鄰編號之順序為左上、上、右上、左...以此類推。

**Step3:**第一次掃瞄與編號標記完成，整理表格中等價編號，找出各等價編號之最小編號。圖 4.21(a)為第一次編號結果之示意圖，圖 4.21(b)為等價表格。

**Step4:**進行第二次掃瞄，重設影像中之編號，把整理出的最小編號取代等價編號，完成影像分割與標記程序。最後執行完之結果與圖 4.20 相同。

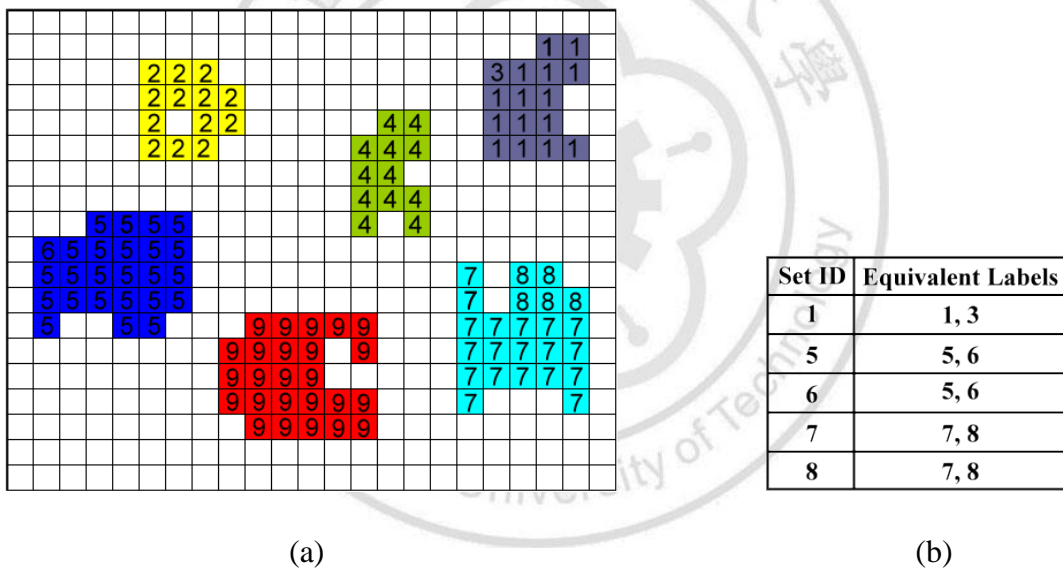


圖4.21 (a)為第一次編號結果之示意圖 (b)等價表格

比較上述兩種實現方法，使用遞迴實現相鄰元素編號法，會讓程式碼較為精簡且可讀性較高，實作難度較易，但是當影像物件像素較多時會造成記憶體堆疊的負擔，容易使程式執行發生問題。因此，本論文使用非遞迴方法實現此演算法，圖 4.22 為影像經由相鄰元素編號法分割成各個獨立之亮點物件，並且用不同顏色



代表各個亮點物件之編號。



圖4.22 亮點物件分割之結果

#### 4.4.2 資訊處理

本論文使用投影分析找出各個亮點物件的中心點。投影分析主要是計算二值化影像中每一個行(對 x 軸投影, 或稱垂直投影)與列(對 y 軸投影, 或稱水平投影)高亮度像素之數量, 圖 4.23 為將一個趨近圓形的影像物件進行投影分析之示意圖, 可以看出影像物件的中心點為 x 與 y 軸投影量最高之處。

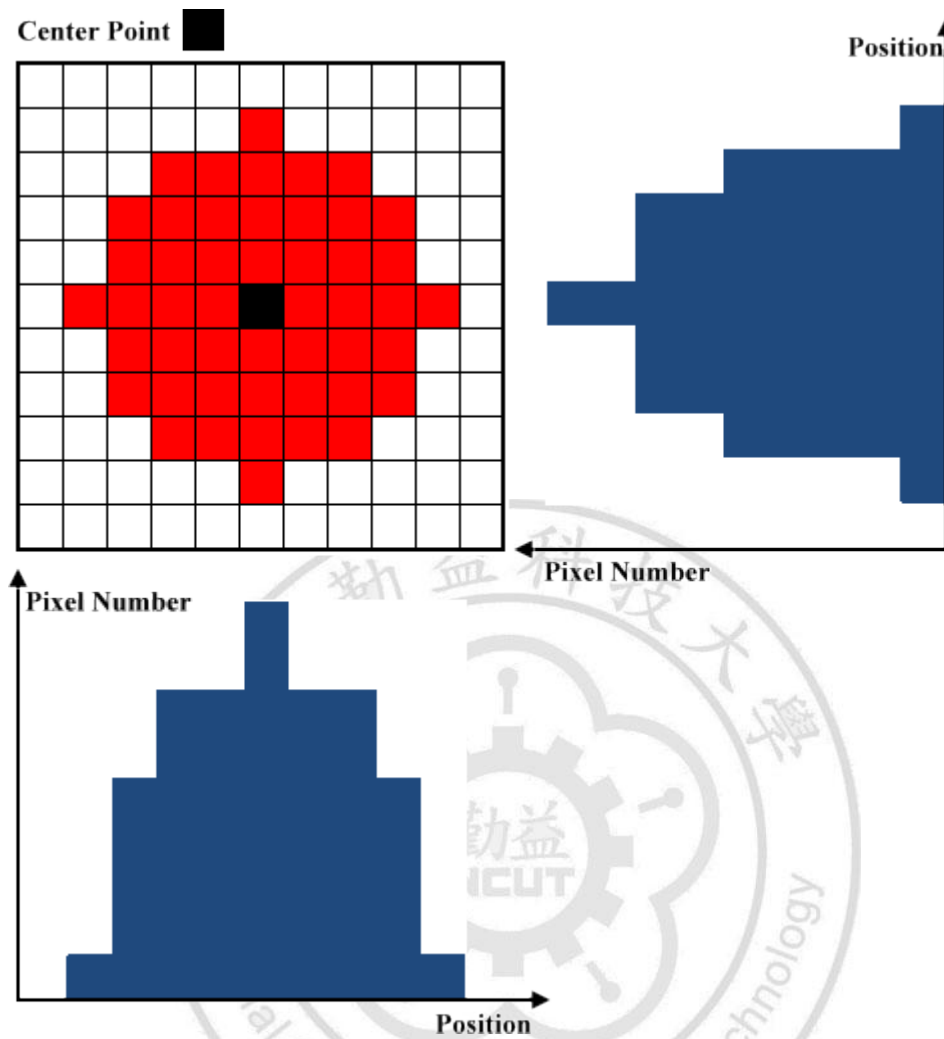


圖4.23 投影分析找尋中心點之示意圖

影像中的亮點物件同樣的也是趨近於圓形，因此，本文對各個亮點物件進行投影分析找出中心點。接下來，以中心點為起點朝上、下、左、右四個方檢查像素，遇到高亮度像素就繼續往原方向前進搜尋，直到非高亮度像素連續出現三次時，把該方向最後出現之高亮度像素位置設定為邊界，如此便能順利的找尋到亮點物件中的上、下、左與右的邊界。圖 4.24 為各個亮點物件中心點的找尋結果，本文使用綠色十字標示中心點的位置，圖 4.25 為各個亮點物件的上、下、左、右邊界位置的搜尋結果，由上與左邊界為起點，右與下邊界為終點劃一矩形，框選

亮點物件。



圖4.24 各個亮點物件中心點標示



圖4.25 各個亮點物件的框選結果

#### 4.5 框選車輛車尾燈(Located Vehicles Taillight)

配對車輛尾燈之前，我們必需區分出中央車道及旁車道的亮點物件(車輛車尾燈)，我們必需防止當不同車輛的車尾燈強度相近時所發生的配對錯誤，因此本文藉由車道線偵測及找尋中央車輛車尾燈，來區分出這些不同車道之亮點物件。其中，車道線偵測的方法在 3.4.1 節中有詳細的敘述，在此不多加陳述。

### 4.5.1 區分不同車道之亮點物件

在這部份，本文使用車道線偵測與中線區分法兩個方法搭配，謹慎的區分出各個亮點物件所屬之車道，圖 4.26 為車道線偵測與中線區分法的關係。當夜間環境照度不足，導致車道線特徵不明顯與車道線段取樣不足，產生無法正常偵測車道線情況時，本文利用前方(中央)車輛的成對車尾燈會橫跨影像中線的特性，找尋前方車輛之車尾燈，如此便可隔開左與右側車道的車輛車尾燈干擾，達到區分不同車道之亮點物件的目的。式(4.17)為判斷中央車輛車尾燈的條件運算式，使條件式成立的兩個亮點物件即為行駛於同車道之車輛的車尾燈，圖 4.27 為在三線道路路情況上的駕駛者車輛行駛於中央車道的示意圖。圖 4.28 為實際影像中線區分之顯示。

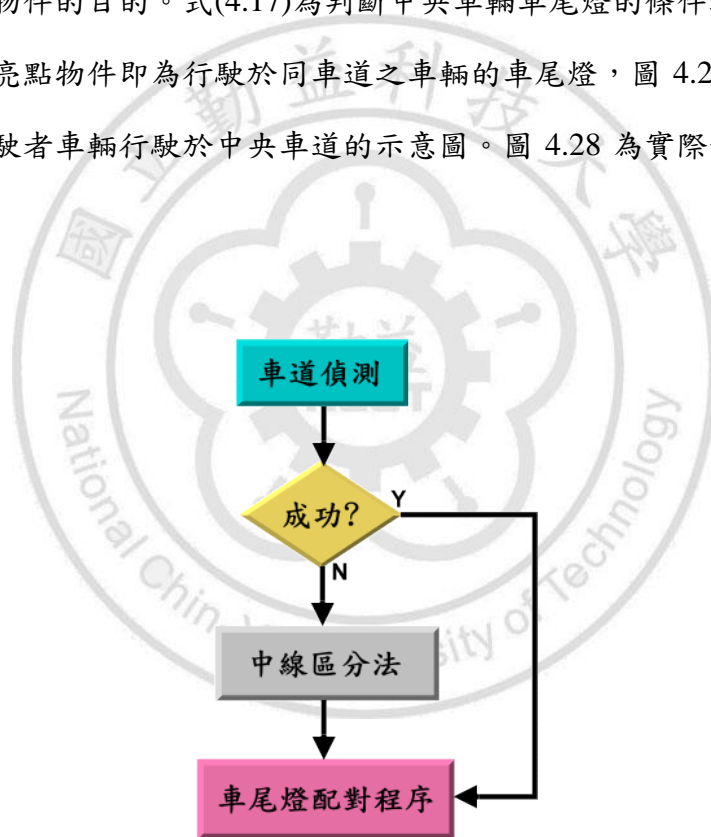


圖4.26 車道線偵測與中線區分法之關係

$$\left[ \frac{O(i) + O(i+1)}{2} - \frac{I_w(max)}{2} \right]^2 \leq \rho(i), \quad i = 1 \dots N, \quad (4.17)$$

$$\rho(i) = \left[ \frac{O(i) - O(i+1)}{2} \right]^2 \quad (4.18)$$

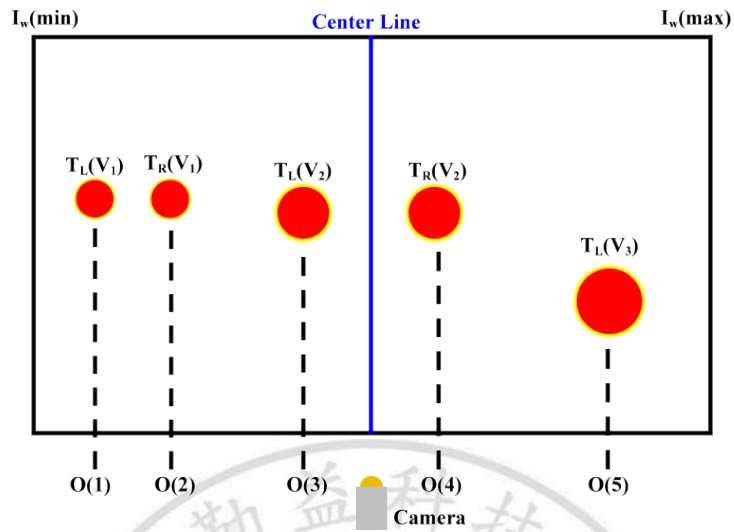


圖4.27 中線區分法之示意圖



圖4.28 實際影像中線區分之顯示

其中， $T_L(V_1)$ 與 $T_R(V_1)$ 為行駛於左側車道上的車輛之左與右的車尾燈，並且離駕駛者車輛較遠， $T_L(V_2)$ 與 $T_R(V_2)$ 則是行駛於同車道之前方車輛的成對車尾燈， $T_L(V_3)$ 為行駛於右側車道，並離駕駛者車輛較近之車輛的左邊車尾燈。本文用  $O(i)$  代表影像中車尾燈物件的編號與索引， $I_w$  為影像之寬度。

若夜間環境下的照度充足，車道線特徵明顯時，將進行車道線偵測找尋目前

所行駛車道的車道遮罩。並利用車道線偵測所產生出來的車道遮罩與經過處理後的亮點物件影像進行 AND 運算，擷取出當前行駛車道的亮點物件。圖 4.29 為 AND 運算處理之結果，可以發現為了完整的擷取亮點物件，本文把直線方程式中的  $x$  截距，擴展車道遮罩的寬度。

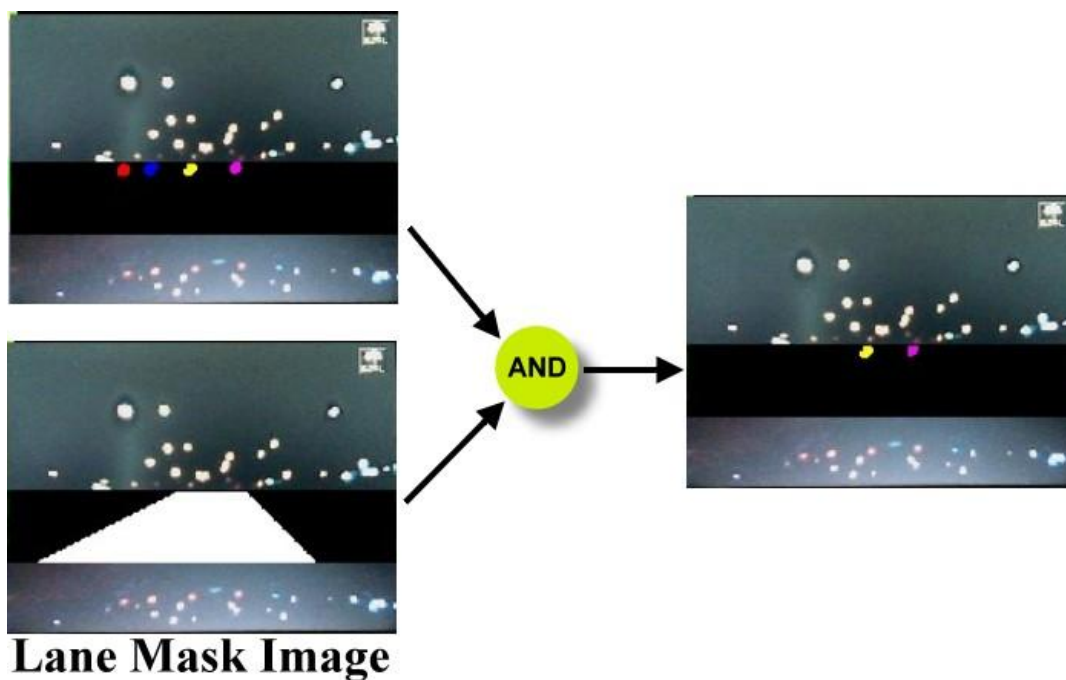


圖4.29 AND 運算處理之結果

#### 4.5.2 車尾燈配對程序(Taillight Pairing Process)

經過以上的處理程序後，本文得到的各個亮點物件的資訊及區分了目前行駛車道及旁側車道的亮點物件，再運用成對車尾燈所存在的某些特性(寬高比例、兩車燈高度比例等)找出同一車輛的車尾燈，並將其標示。其中，本文所提出的車尾燈配對程序參考文獻[11]-[13]將其改良，而步驟如下所述。

**Step1:** 定義某一個亮點物件為 $B_i$ 。

**Step2:** 定義亮點物件中的上、下、左及右邊界的座標各自為 $t(B_i)$ 、 $b(B_i)$ 、 $l(B_i)$ 及 $r(B_i)$ 。

再利用這些座標求得亮點物件的寬度 $W(B_i)$ 及高度 $H(B_i)$ ，圖 4.30 為亮點物件的資訊。

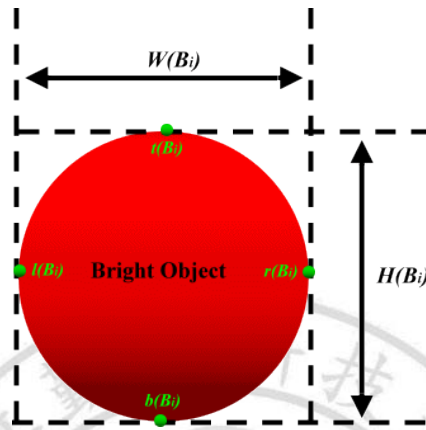


圖4.30 亮點物件之資訊

**Step3:** 定義 $N_h(B_i, B_j)$ 及 $N_v(B_i, B_j)$ 各別為兩個亮點物件間水平方向及垂直方向的像素數量。他們分別由式(4.19)及(4.20)各自求得。當兩個亮點在水平方向重疊， $N_h(B_i, B_j)$ 將會變成負值。相對的，當兩個亮點並無在垂直方向重疊， $N_v(B_i, B_j)$ 也將成為負值。圖 4.31 與 4.32 為水平與垂直方向亮點物件之間的關係圖。

$$N_h(B_i, B_j) = \max[l(B_i), l(B_j)] - \min[r(B_i), r(B_j)] \quad (4.19)$$

$$N_v(B_i, B_j) = \min[b(B_i), b(B_j)] - \max[t(B_i), t(B_j)] \quad (4.20)$$



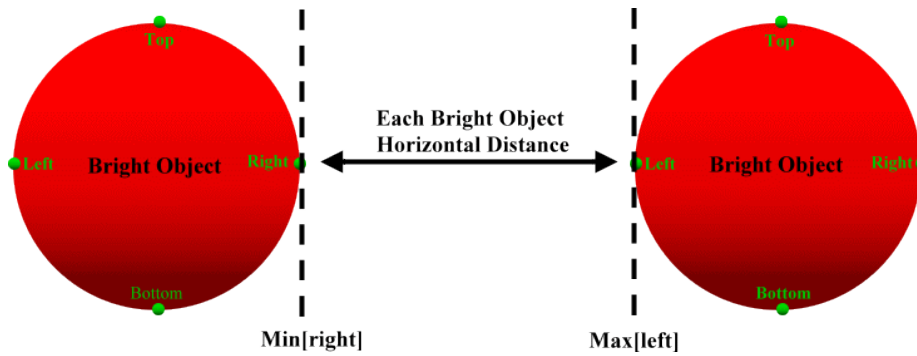


圖4.31 水平方向亮點物件之間關係圖

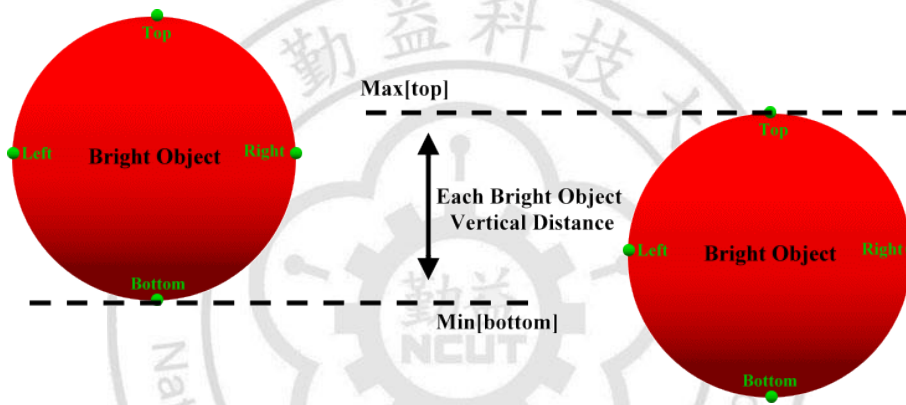


圖4.32 垂直方向亮點物件之間關係圖

**Step4:** 計算兩個亮點物件之間垂直方向重疊比率，換句話說就是計算以兩個亮點物件之間水平線為基礎的偏移量， $O_v(B_i, B_j)$  為兩個亮點之間的垂直方向重疊比率。由公式(4.21)將可以求得  $O_v(B_i, B_j)$  的值，並判斷  $O_v(B_i, B_j)$  有無滿足門檻值  $T_o$ 。其條件式如(4.22)。

$$O_v(B_i, B_j) = N_v(B_i, B_j) / \min[(H(B_i), H(B_j))] \quad (4.21)$$

$$O_v(B_i, B_j) > T_o \quad (4.22)$$

**Step5:** 計算 $H(B_s)/H(B_l)$ ，此為兩個亮點物件的高度比例。並藉由條件式(4.23)判斷其比例有無達到門檻值 $T_{sh}$ 。其中， $B_s$ 為兩亮點之間高度較小亮點， $B_l$ 則為較大者。圖 4.33 為兩個亮點物件的高度比例關係圖。

$$\frac{H(B_s)}{H(B_l)} > T_{sh} \quad (4.23)$$

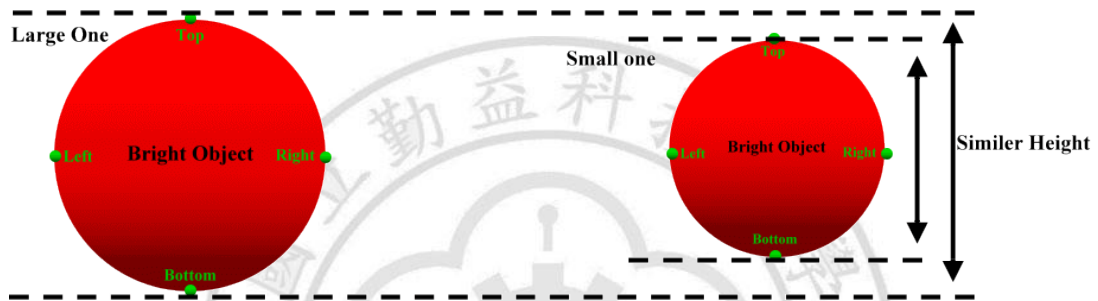


圖4.33 兩亮點物件的高度比例關係圖

**Step6:** 求出成對亮點物件的寬度(車輛寬度)與單一亮點物件寬度比例，是否滿足條件式(4.24)。其中，成對亮點物件的寬度藉由公式(4.25)求出。圖 4.34 為亮點物件與車輛寬度之關係圖。

$$VW(B_i, B_j) < \max[W(B_i), W(B_j)] * T_n \quad (4.24)$$

$$VW(B_i, B_j) = N_h(B_i, B_j) + W(B_i) + W(B_j) \quad (4.25)$$

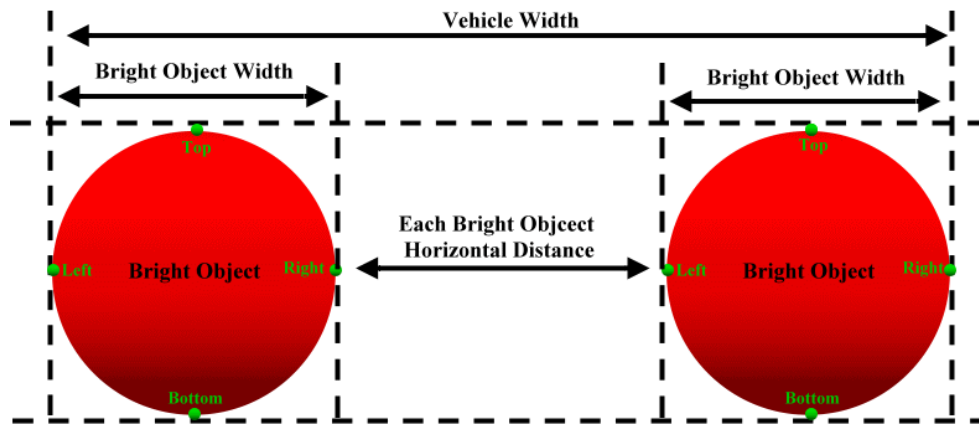


圖4.34 亮點物件與車輛寬度之關係圖

**Step7:** 當上述條件式(4.22)、(4.23)及(4.24)皆成立時，系統將會把這對亮點物件使用矩形的邊界盒框選起來。此時再計算矩形邊界盒的寬高比例(車寬及亮點物件高度)，並判斷是否滿足條件式(4.26)。圖 4.35 為矩形邊界盒之寬高比例關係圖。

$$2.5 < (VW(B_i, B_j)/H(B_l)) < 8 \quad (4.26)$$

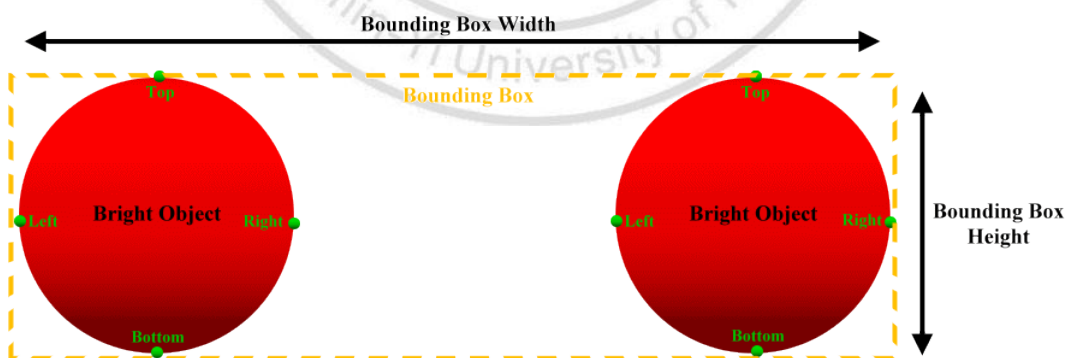


圖4.35 矩形邊界盒之寬高比例關係圖

以上所有條件式均滿足後將會標示出車輛在影像中的位置。其中門檻值 $T_o$ 、 $T_{sh}$

及 $T_n$ 各自為 0.7、0.8、6。以上門檻值皆是根據成對車尾燈的規格所設定。圖 4.36 為車尾燈配對的結果。



圖4.36 車輛車尾燈配對之結果

## 4.6 驗證車輛影像(Vehicle Verification)

配對車輛尾燈完成之後，本文利用連續影像的特性，驗證車尾燈配對後的結果。首先，紀錄前一張影像中前方車輛的左右邊界位置，以供下一張影像驗證使用。下一張影像經由車輛尾燈配對後框選出車輛的位置，此時利用上一張影像中前方車輛的左右邊界位置與下一張影像的車輛的左右邊界位置比對，是否有無過度偏移。如前一張影像的前方車輛的左邊界或右邊界與下一張影像中前方車輛的右邊界及左邊界重疊，本文則就判定此偵測結果是無效的，並進行修正，修正的方法是把上一張影像車輛偵測的結果套用至下一張影像。反之，則是正確的偵測結果。圖 4.37(a)與(d)顯示數個亮點物件的強度與大小相似時的情況，(c)為系統將與前方車輛鄰近之車輛的車尾燈框選的錯誤情況，(d)則為透過上述機制驗證修正的結果。

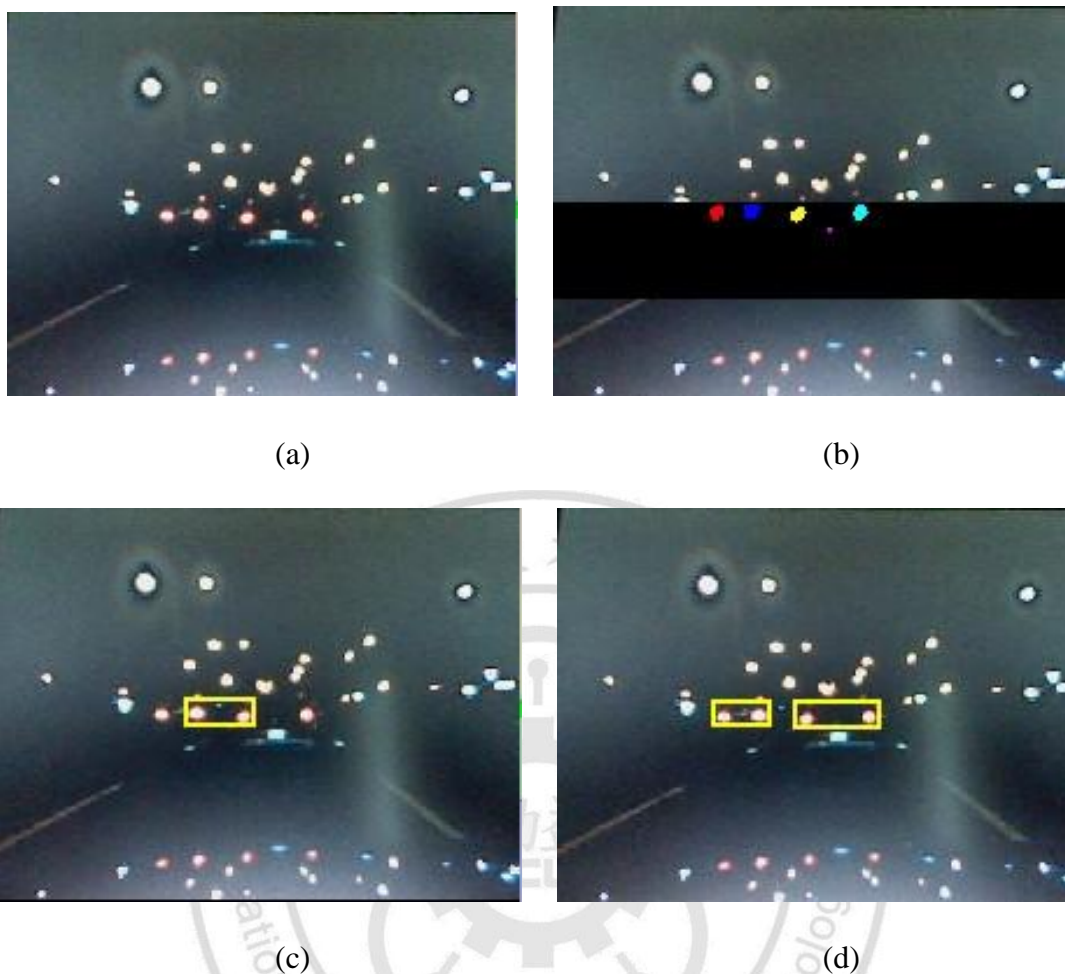


圖4.37 (a)原始輸入影像 (b)影像中亮點物件之強度與大小相似 (c)誤判情況 (d) 經由前一張影像修正後結果

## 4.7 車輛追蹤模式(Tracking Mode)

從影像輸入一直到對車輛影像的驗證，本文使用了多層次的運算來堆疊出車輛偵測演算法。如果每一張輸入影像都經過這麼複雜的影像處理程序，勢必造成相當程度的處理延遲，以 PC 架構的觀點來看也許沒有太多的顧慮，但是在嵌入式平台上，這些運算量已大到足以降低系統的工作效率，即時性也將大打折扣。

我們利用車輛在高速行駛時的特性，也就是移動變化小的特性，利用絕對值

平均誤差法(Mean Absolute Error, MAE) 計算前方車輛影像區域在時間軸上的差異性，如果變動小於門檻值 8，則斷定車輛並沒有改變位置，在下一張影像輸入時跳過偵測演算法，直接進行追蹤。直到發現影像區域的平均差異值大於預設門檻值時，才又重新使用車輛偵測演算法找回車輛。這樣的作法能有效的降低整體運算量。其中，車輛影像的下邊界為車尾燈的底部邊界往下延伸 0.45 倍車寬，上邊界則是由下邊界往上推算 0.85 倍車寬所求得。式(4.27)為 MAE 的求得方法，其中 K 與 L 為車輛影像區域的寬與高。圖 4.38 為追蹤之結果。

$$MAE = \frac{1}{K * L} \sum_{i=0}^K \sum_{j=0}^L [I_t(x + i, y + j) - I_{t-1}(x + i, y + j)]^2 \quad (4.27)$$



圖4.38 車輛追蹤之結果



## 第五章 距離及碰撞時間估測

為了判斷前方車輛與駕駛者車輛之間的安全關係，本文對前方車輛進行距離的估測，然而單純藉由車距的資訊來決定是否為危險的狀況，似乎是不夠嚴謹。前方碰撞發生的主要原因多數是在於兩車之間的速度差距。因此，本論文使用前方車輛的距離與即將碰撞之時間來觸發危險情況時的語音警示。

### 5.1 前方車輛距離估測

前面數個章節中，本論文針對夜間與日間的情況，各別由多階層之影像處理運算找到車輛位置資訊並標示車輛邊界。在日間情況下，本文藉由前方車輛底部邊界，推算前方車輛與駕駛者車輛之間距離。首先，將所擷取的二維影像座標(Image Coordinate)轉換至攝影機座標(Camera Coordinate)，再把攝影機座標轉換至車輛座標(Vehicle Coordinate)。真實世界中的三維座標都稱之為世界座標(World Coordinate)，像是上述所提到的攝影機座標與車輛座標，兩者的差別在於，攝影機座標是以攝影機的位置為原點所建構的世界座標，車輛座標則是以攝影機正下方之地面位置為原點來建立的世界座標。式(5.1)為將世界座標轉換至影像座標的透視公式，式(5.2)則為將影像座標轉換至世界座標的反透視公式。圖 5.1 為二維影像座標轉換至攝影機座標之示意圖。

$$u_i = \frac{fS_u X_c}{Z_c}, \quad v_i = \frac{fS_v Y_c}{Z_c} \quad (5.1)$$

$$\frac{Z_c}{Y_c} = \frac{fS_v}{v_i}, \quad Z_c = \frac{fS_v Y_c}{v_i} \quad (5.2)$$



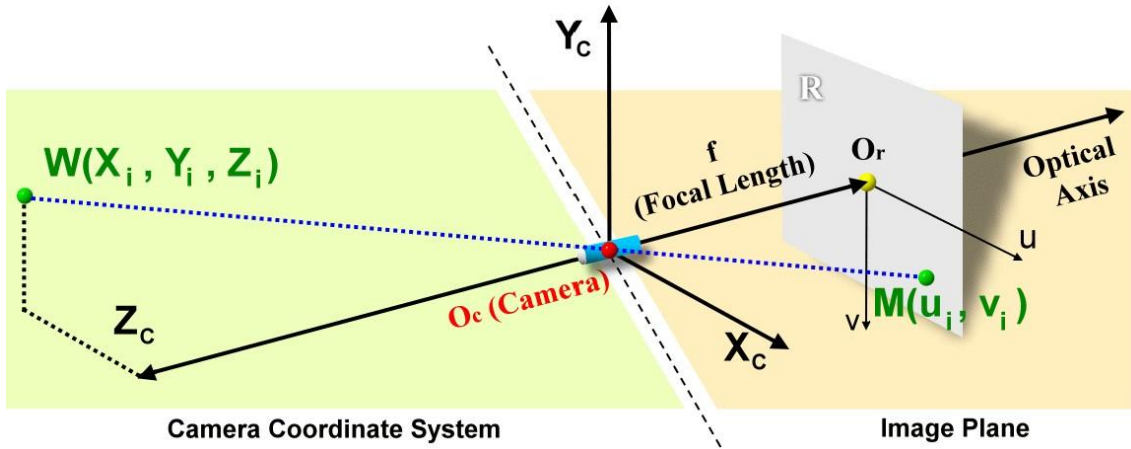


圖5.1 二維影像座標轉換至攝影機座標之示意圖

式(5.1)與(5.2)中， $f$  為攝影機之焦距(Focal Length)， $S_u$ 與 $S_v$ 各別為寬(x 軸方向)與高(y 軸方向)的尺度因子(Scaling Factor)主要用於實際空間中的物理單位與影像中的最小單位像素兩者之間的轉換，而這參數都是攝影機在出廠時就以設定。 $Z_c$ 為攝影機與前方目標物之間的距離， $Y_c$ 為現實世界中前方目標物之高度， $v_i$ 為二維影像中前方目標物件(車輛)的底部位置(y 軸方向)。圖 5.1 中， $W(X_i, Y_i, Z_i)$ 為前方目標物件在攝影機座標中的位置， $M(u_i, v_i)$ 為前方目標物件映射到二維影像中的位置。接下來，把攝影機座標轉換至車輛座標，則必需考慮攝影機架設之高度與傾斜角度，如圖 5.2 所示。而式(5.3)與式(5.4)為轉換至車輛座標後的修正結果。其中，前方車輛之距離為 $Z_v$ ，攝影機之傾斜角度為 $\varphi$ ，攝影機架設高度為  $H$ 。本論文攝影機的架設與道路平行，因此傾斜角度為零。

$$Y_v = \tan \varphi \cdot Z_v \quad (5.3)$$

$$u_i = \frac{e_u X_v}{Z_v}, \quad v_i = \frac{e_v (Y_v - H)}{Z_v}, \quad e_u = f S_u, \quad e_v = f S_v \quad (5.4)$$

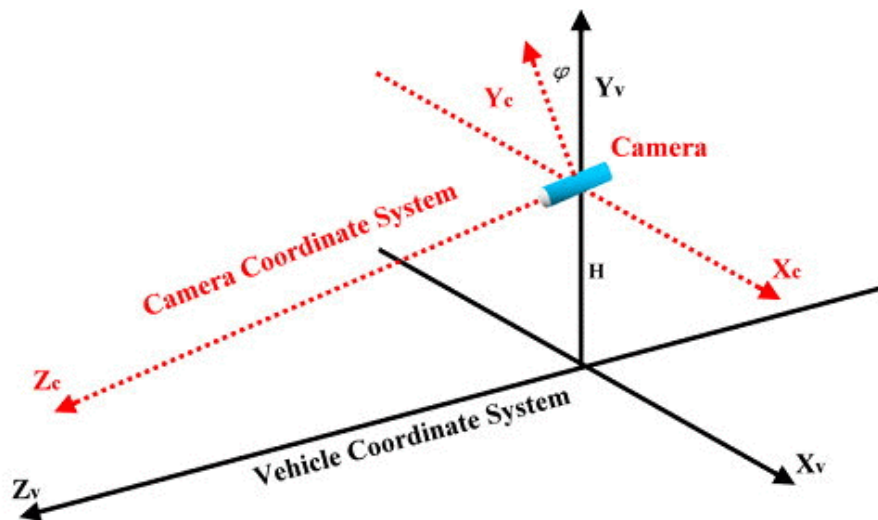


圖5.2 攝影機座標轉車輛座標之示意圖

在夜間情況下，由於車輛底部邊緣特徵不明顯，因此本文利用影像中車輛的寬度與現實車輛的寬度進行距離的估測。如同上述座標轉換之過程，整理出式(5.5)，求出前方車輛距離  $d$ 。 $W_v$  為現實中車輛之寬度， $W_i$  為影像中車輛之寬度， $S_u$  為攝影機的寬度刻度因子。而現實中車輛之寬度，則是根據一般小客車的車輛寬度去設定，經由觀察與統計一般小客車之寬度約為一點八公尺。

$$d = \frac{f \cdot W_v}{W_i \cdot S_u} \quad (5.5)$$

## 5.2 碰撞時間估測

在 5.1 節中計算出前方車輛與駕駛者車輛之間的距離，但是危險的情況定義，是必需經由距離及車輛本身的速度來決定。因此，本文估算駕駛者車輛與前方車輛的碰撞時間(Time to Collision, TTC)[14]，讓駕駛者能夠即時反應，避免事故的發生。式(5.6)為一個簡單的估算碰撞時間  $t_{TTC}$  的數學模型， $d$  為兩車之間距離， $v_r$  為

兩車之間的相對速度，由於我們無法取得前方車輛與駕駛者車輛的行駛速度，因此藉由式(5.7)推算兩車的相對速度可由在  $t$  時間內距離的變化求得， $t$  時間主要是前一張影像至下一張影像的間隔時間，而距離的變化程度是由前一張(n-1)與目前影像(n)相減取得。把式(5.7)代入式(5.6)中可求得式(5.8)， $t_{TTC_n}$  為第  $n$  張影像中的碰撞時間， $\omega_{fps}$  為一秒鐘處理數張影像的頻率， $d_n$  為第  $n$  張影像的車距，求法如式(5.9)，當中的  $f$ 、 $W_v$  與  $S_u$  皆為固定常數，因此用  $C$  代替。最後把式(5.9)代入(5.8)中，並將常數  $C$  約分消除，經整理後得式(5.10)。式(5.10)中只需要系統運算時間與影像中車輛寬度的資訊就可求出碰撞時間，此方法不需藉由額外輸入設備取得兩車之速率。只使用以視覺影像為基礎的計算方式即可達到車輛碰撞時間估測目的。

$$t_{TTC} = \frac{d}{v_r} \quad (5.6)$$

$$v_r = \frac{\Delta d}{t} \quad (5.7)$$

$$t_{TTC_n} = \frac{d_n}{\frac{d_{(n-1)} - d_n}{\frac{1}{\omega_{fps}}}} \quad (5.8)$$

$$d = \frac{fW_v}{W_i S_u} = C \frac{1}{W_i}, \quad C = \frac{fW_v}{S_u} \quad (5.9)$$

$$t_{TTC_n} = \frac{1}{\omega_{fps}} \cdot \frac{C \cdot W_{i_n}}{C \cdot \frac{1}{W_{i_n}} - C \cdot \frac{1}{W_{i_{n-1}}}} = \frac{1}{\omega_{fps}} \cdot \frac{W_{i_n}}{\frac{1}{W_{i_n}} - \frac{1}{W_{i_{n-1}}}} \quad (5.10)$$

## 第六章 基於可拓理論的日夜間整合

本論文將日間與夜間車輛偵測演算法整合於一嵌入式系統中，然而兩個演算法切換的方法也是一個重要的課題。多數研究[15][22]利用天空的色度變化作為判別日夜環境的因素，但是在日與夜之間有許多矛盾、模糊地帶(像是陰天、黃昏、清晨等)，單使用一個環境因素進行判別是較不嚴謹且容易發生錯誤的。因此，為了加強日夜判別的準確與強健性，本文使用可拓理論[3][26]進行日夜間的決策。

### 6.1 可拓理論簡介

中國學者蔡文於 1983 年提出可拓理論[26]，其主要的功用就是解決現實世界中的矛盾、模糊問題。可拓理論是由兩大主軸構成：(1)物元理論(2)可拓集合理論，利用形式化的方法，從定性與定量兩個方向去研究解決問題，接下來將對可拓理論中的物元理論與可拓集合進行介紹。

#### 6.1.1 物元基本理論

物元是作為可拓理論中的邏輯細胞，它是用來描述事物的基本元素，物元概念與定義是以事物的名稱(Name, N)，該事物之特徵(Characteristic, C)與特徵的量值(Value, V)三者所組成的三元組，其中特徵量值 V 也可表示為  $c(N)$ ，以數學式子描述可表示為式(6.1)與(6.2)。

$$R = (N, c, v) \quad (6.1)$$

$$R = (N, c, c(N)) \quad (6.2)$$

由上述得知，物元是把事物、特徵與量值集於統一體中，同時這也是物元組成必要的三要素，其三大要素之物理意義如下所示。

### (1) 事物(Name)

人們為了分辨與認識事物，常使用不同的名稱來代表、稱呼它們，像是電腦、車輛、杯子、手錶等，物元概念與定義中的事物  $N$  就是指事物的名稱(事物)。如果根據事物的屬性，可以分為個事故與類事故，像是課本、國家、機車等是屬於類事故；數學課本、中國、機車  $A$  則是屬個事故。個事故在書寫時一定要有具體的名稱。

### (2) 特徵(Characteristic)

事物的特徵定義主要是能表示事物的特性、功能、行為狀態與事物跟事物之間的關係。同一個事物，可以透過各種形式上的特徵來呈現。如事物小明(人)，有身高、體重、年齡等特徵。特徵根據解決問題的需要可以分為三類：

- (a) 功能特徵：敘述事物的功能或用途，如通訊能力、速度能力等
- (b) 性質特徵：敘述事物的性質，如導熱系數、電子濃度等。
- (c) 實體特徵：敘述事物的實體，如身高、體重等。

### (3) 量值(Value)

事物中某一特徵的數量、程度或範圍等可以被量化的單位皆可稱為該事物特徵的量值。量值又分為數量化與非數量化量值，可以使用數字來表示的量值就稱為數量化量值，如：30 公尺、60 公克等。相反的，無法使用數字來表示的量值就稱為非數量化量值，如：優等、黃色、強等。

當一個事物存在著多個特徵與量值時，我們稱此為多維物元。如果一個事物  $N$  以  $n$  個特徵  $c_1, c_2, \dots, c_n$  及相對應之量值  $v_1, v_2, \dots, v_n$ ，則可以使用一個矩陣描述，如式(6.3)。

$$R = \begin{bmatrix} N & c_1 & v_1 \\ & c_2 & v_2 \\ & \dots & \dots \\ & c_n & v_n \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \dots \\ R_n \end{bmatrix} \quad (6.3)$$

其中， $R$  為  $n$  維物元， $R_i$  為為  $R$  的分物元  $R_i = (N, c_i, v_i)_{i=1,2,\dots,n}$ 。  $R$  也可被以式 (6.4) 表示。

$$R = (N, C, V), C = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix}, V = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} \quad (6.4)$$

### 6.1.2 可拓集合理論

在一般數學所使用的集合，我們稱為古典集合。在古典集合中使用 1 與 0 代表某一對象是否屬於該集合，而模糊集合則是定義於  $(0, 1)$ 。可拓集合定義中，假設  $U$  為論域，若對  $U$  中任一元素  $u$ ，都有一實數  $K(u) \in (-\infty, \infty)$  對應，則可拓集合定義式為式 (6.5)。

$$\tilde{A} = \{(u, y) | u \in U, y = K(u) \in (-\infty, \infty)\} \quad (6.5)$$

式 (6.5) 為論域  $U$  上的一個可拓集合，其中  $\tilde{A}$  的關聯函數為  $y=K(u)$ ， $K(u)$  為  $u$  關於  $\tilde{A}$  的關聯度。 $\tilde{A}$  的正域、負域、與零界皆由式 (6.6)、(6.7) 與 (6.8) 所示，由上述三式可知若  $u \in J_0$ ，則  $u \in A$ ，同時  $u \in \bar{A}$ 。

$$A = \{u | u \in U, K(u) \geq 0\} \quad (6.6)$$

$$\tilde{A} = \{u | u \in U, K(u) \leq 0\} \quad (6.7)$$

$$J_0 = \{u | u \in U, K(u) = 0\} \quad (6.8)$$

### 6.1.3 關聯函數

在實際運用可拓集合解決矛盾問題時，遇到的是各式各樣的實際事物。各種事物都必需透過量化，然後運用在可拓集合中。可拓集合是用關聯函數來刻劃的，因此，本文要建立可拓集合中的關聯函數來解決問題。接下來介紹初等關聯函數。為此先引入距和位值的概念。

#### A 距

古典數學中定義點與點的距離為式(6.9)。而在實域上作用的是點與區間，為了建立關聯函數，我們必須定義點與區間的距離，也就是所謂「距」的概念。假設  $x$  為實域  $(-\infty, +\infty)$  上之任一點，那實域上任一區間為  $X_0 = \langle a, b \rangle$ ，如式(6.10)定義點  $x$  與區間  $X_0$  的距離。圖 6.1 與 6.2 為當點  $x$  位置在區間  $X_0$  外與在區間  $X_0$  內的示意圖。

$$\rho(x, y) = |x - y| \quad (6.9)$$

$$\rho(x, X_0) = \left| x - \frac{a+b}{2} \right| - \frac{1}{2}(b-a) \quad (6.10)$$

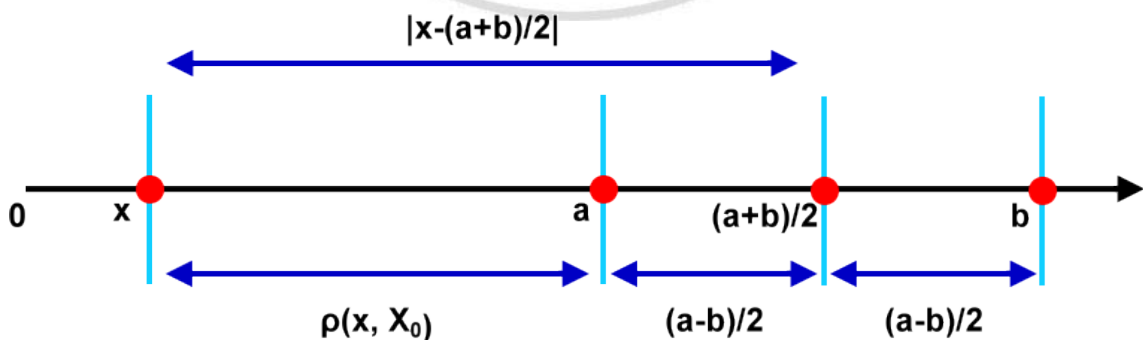


圖6.1 點  $x$  在區間  $X_0$  之外時距的關係



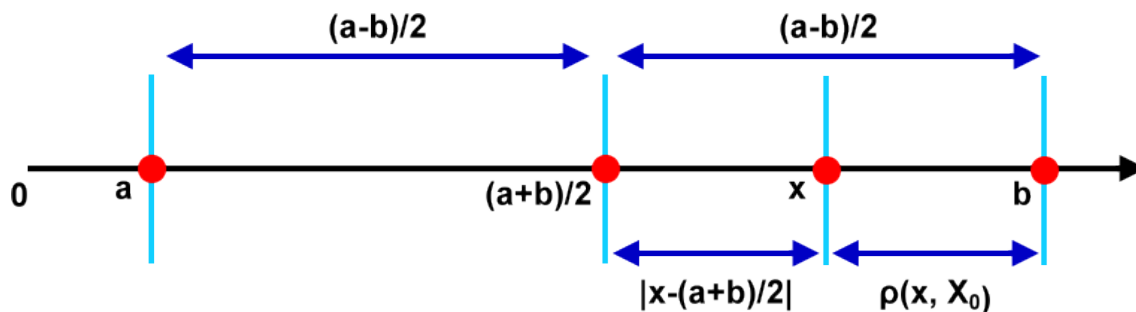


圖6.2 點  $x$  在區間  $X_0$  之內時距的關係

### B 位置值(位值)

在可拓集合與現實問題中，除了需考慮點與區間的位值關係外，也要考慮區間與區間或一個點與兩區間的位值關係。通常假設  $X_0 = \langle a, b \rangle$ ,  $X = \langle c, d \rangle$  且  $X_0 \subset X$ ，則點  $x$  與  $X_0$  與  $X$  之位值之定義如式(6.11)。

$$D(x, X_0, X) = \begin{cases} \rho(x, X) - \rho(x, X_0), & x \notin X_0 \\ -1, & x \in X_0 \end{cases} \quad (6.11)$$

一般來說， $X_0$  為經典域， $X$  為節域。當  $x$  位於經典域內時，位值等於負一；反之，當點  $x$  位於經典域外時，位值等於點  $x$  到兩域的距離差值，點  $x$  所在位置區域與  $D(x, X_0, X)$  之關係如圖 6.3 與 6.4 所示。

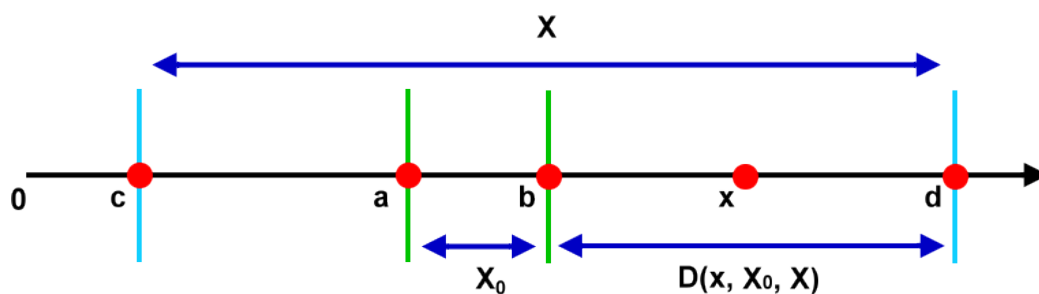


圖6.3 點  $x$  存在於  $b$  與  $d$  區間的位值

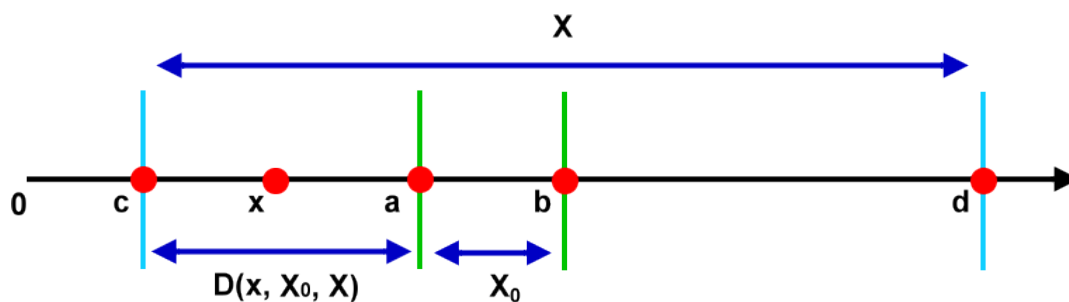


圖6.4 點  $x$  存在於  $c$  與  $a$  區間的位值

### C 初等關聯函數

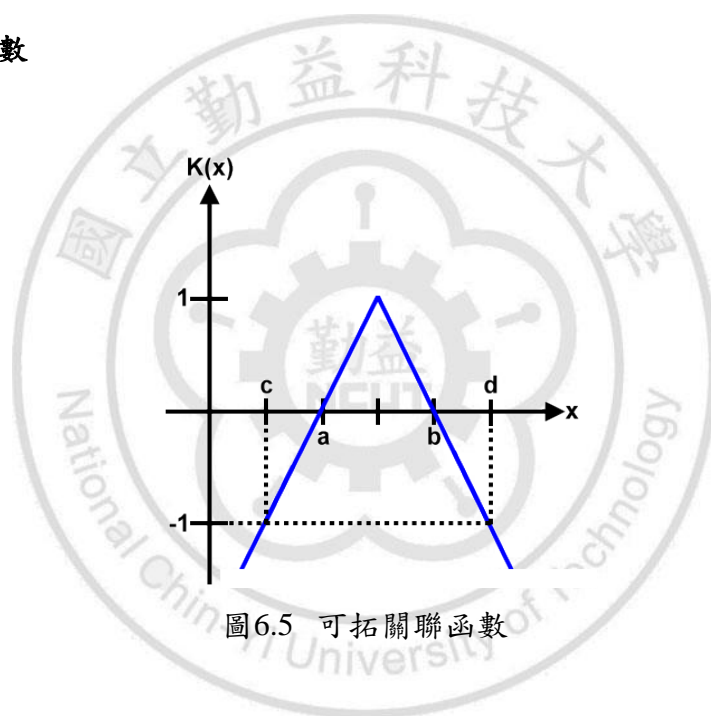


圖6.5 可拓關聯函數

假設兩區域  $X_0 = \langle a, b \rangle$  與  $X = \langle c, d \rangle$ ,  $X_0 \subset X$  且無公共端點, 則其  $x$  關於  $X_0$  與  $X$  的初等關聯函數如式(6.12)。關聯函數可以計算點  $x$  屬於區域  $X_0$  的關聯程度,  $K(x) \geq 0$  代表  $x$  屬於  $X_0$  的程度, 反之,  $K(x) < 0$  則代表  $x$  不屬於  $X_0$  的程度。利用可拓關聯函數可以計算  $x$  與  $X_0$  的程度, 圖 6.5 為可拓關聯函數。

$$K(x) = \frac{\rho(x, y)}{D(x, X_0, X)} \quad (6.12)$$

## 6.2 可拓日夜間辨識方法

本論文使用可拓理論進行日間與夜間天色的辨識，利用影像中天空與道路的亮度灰階值作為特徵來區分日間與夜間之情況，其中加入道路的亮度灰階值特徵，增加取樣的數目，能增加辨識的準確度。本文對天空與道路區域取平均灰階值進行取樣來建立物元，圖 6.6 為天空與道路區域的平均灰階值取樣的範圍，藉由縮小取樣範圍可以降低運算量，取平均值能降低高頻的雜訊干擾(如：夜晚的路燈號誌、日間的烏雲或遠山等)。



圖6.6 天空與道路亮度灰階值的取樣區域 (a)夜間 (b)日間

接著介紹可拓日夜間識別的步驟與程序：

**Step1:** 建立日間與夜間情況下亮度灰階值的經典域與節域範圍，如式(6.13)至(6.16)，

其中 $R_0$ 內的 $V_0$ 為經典域之範圍， $R_p$ 內的 $V_0$ 為節域之範圍。本文各自統計一萬筆影像中天空與道路區域的平均灰階值並計算其變異量，如圖 6.7 與 6.8，藉由這些統計數據資料定義出經典域之範圍，節域的選定是根據各個經典域之上下限而定，其選定範圍僅會影響關聯函數可拓域之範圍。除此之外，

我們加入了道路與天空灰階值相加的特徵，讓決策時有另一個可參考之因素。

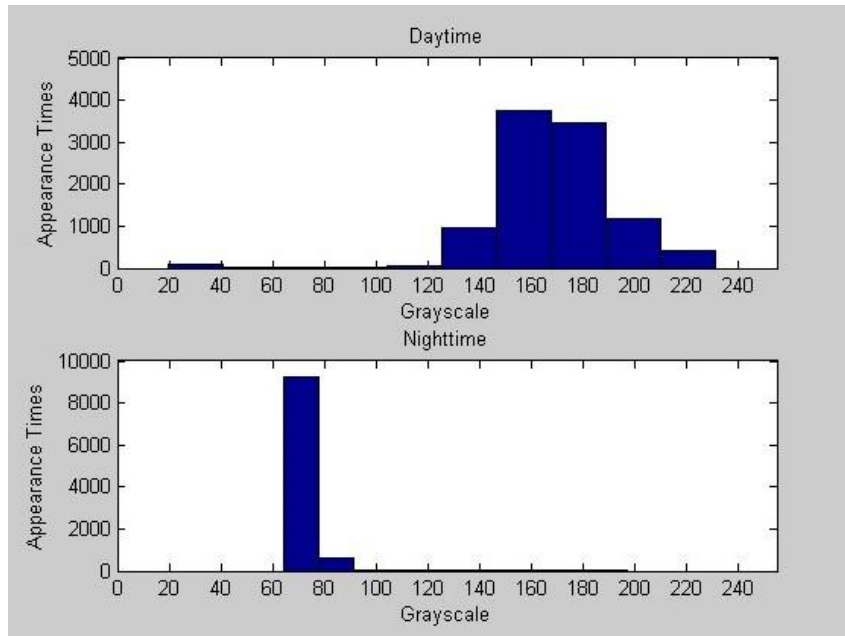


圖6.7 各一萬筆日間與夜間天空灰階值的統計結果

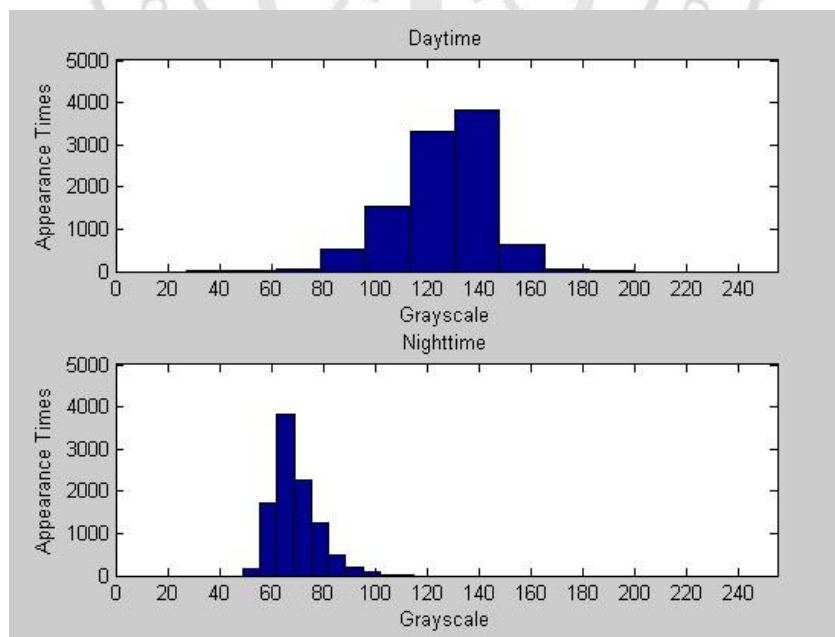


圖6.8 各一萬筆日間與夜間道路灰階值的統計結果

$$R_0 = (P, C, V_0) = \begin{bmatrix} \text{Nighttime} & \text{Sky} & \langle 60, 100 \rangle \\ & \text{Road} & \langle 40, 80 \rangle \\ & \text{Sky\_ \& Road} & \langle 35, 175 \rangle \end{bmatrix} \quad (6.13)$$

$$R_p = (P, C, V_p) = \begin{bmatrix} \text{Nighttime} & \text{Sky} & \langle 0, 120 \rangle \\ & \text{Road} & \langle 0, 120 \rangle \\ & \text{Sky\_ \& Road} & \langle 0, 190 \rangle \end{bmatrix} \quad (6.14)$$

$$R_0 = (P, C, V_0) = \begin{bmatrix} \text{Daytime} & \text{Sky} & \langle 110, 235 \rangle \\ & \text{Road} & \langle 85, 200 \rangle \\ & \text{Sky\_ \& Road} & \langle 180, 300 \rangle \end{bmatrix} \quad (6.15)$$

$$R_p = (P, C, V_p) = \begin{bmatrix} \text{Daytime} & \text{Sky} & \langle 100, 255 \rangle \\ & \text{Road} & \langle 70, 255 \rangle \\ & \text{Sky\_ \& Road} & \langle 170, 350 \rangle \end{bmatrix} \quad (6.16)$$

**Step2:** 確定待識別的對象形成的物元，把測量結果用物元表示，如式(6.17)。

$$R_i = (P, C, V_i) = \begin{bmatrix} p & \text{Sky} & v_1 \\ & \text{Road} & v_2 \\ & \text{Sky\_ \& Road} & v_3 \end{bmatrix} \quad (6.17)$$

**Step3:** 計算待測物元與先前定義的經典域和節域的距與關聯函數，距的計算如先前式(6.10)，關聯函數之計算由式(6.12)整理成式(6.18)所得。

$$K_j(v_i) = \begin{cases} \frac{-\rho(v_i, V_{0ij})}{|V_{0ij}|}, & v_i \in V_{0ij} \\ \frac{\rho(v_i, V_{0i})}{\rho(v_i, V_{pi}) - \rho(v_i, V_{0ij})}, & v_i \notin V_{0ij} \end{cases} \quad (6.18)$$

$$i = 1, 2, 3, \quad |V_{0ij}| = \left| \frac{b_{0j} - a_{0j}}{2} \right|$$

**Step4:** 依照各個特徵的重要性給予權重係數 $a_i$ ，權重係數的分配的比例如式(6.19)。

本論文給予道路亮度灰階值的權重較高，因為天空區域的統計結果得知其變異量較低。

$$a_i = \{a_1, a_2, a_3\} = \{0.4, 0.3, 0.3\} \quad (6.19)$$

**Step5:** 計算待測物元與夜間和日間物元的隸屬程度 $K_j(p)$  (關聯度)。

$$K_j(p) = \sum_{i=1}^n a_i K_j(v_i) \quad (6.20)$$

**Step6:** 最後，找出隸屬程度最大值者，判定該待測物元屬於日間或夜間情況。

$$Result = Max[K_j(p)], j = 1, 2 \quad (6.21)$$

可拓日夜間識別的流程如圖 6.9 所示，首先收集資料並建立經典域與節域，並使用可拓識別方法進行學習，學習過程中，藉由識別並觀察結果是否正確。假如不正確或識別誤差值過大，則調整權重值改善識別結果，完成學習。最後就使用以建立的資料進行可拓日夜識別。

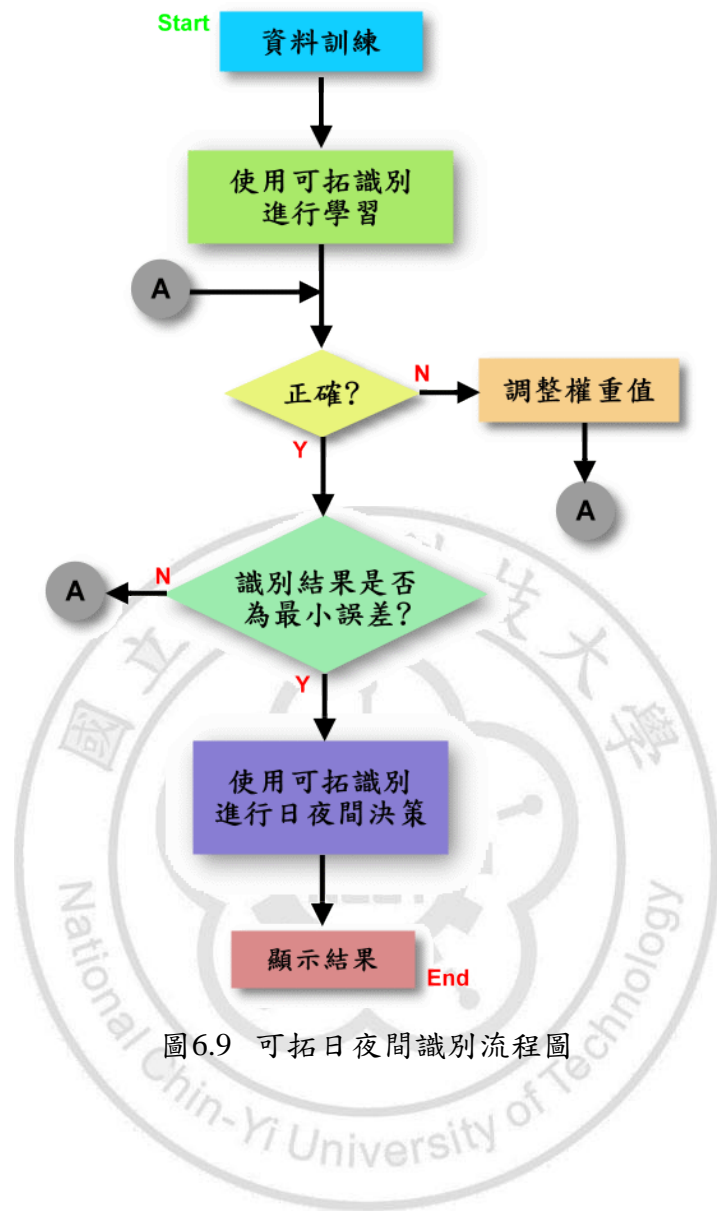


圖6.9 可拓日夜間識別流程圖



## 第七章 實驗結果與分析

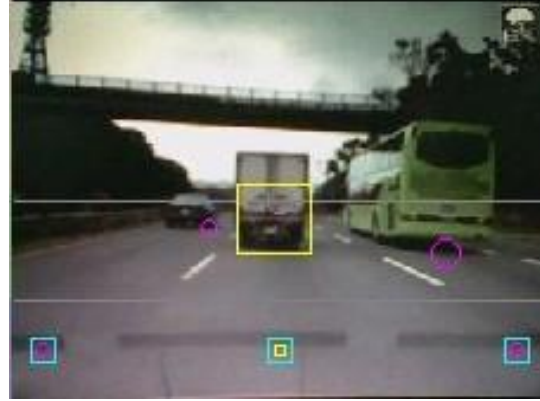
### 7.1 實驗結果

#### 7.1.1 日間車輛偵測結果

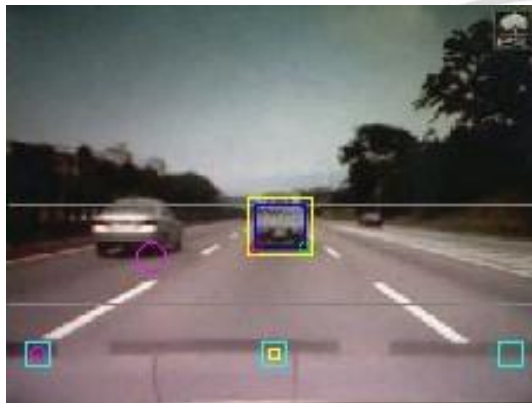
當駕駛者車輛行駛於高速公路上的中央車道時，車輛偵測的範圍為前方三線車道。在影像中下方的正方形框的顯示目前該車道正在執行車輛偵測，紫色圓圈是用來框選旁側車道之車輛，黃色矩形框為標示目前行駛車道之前方車輛所用。圖 7.1(a)為一個影像複雜度最低的車輛偵測情況，圖 7.1(b)為小貨車行駛於同車道上且兩側皆有車輛，由於小貨車形體與一般小客車不同，因此邊界盒之上邊界無法完全框選車輛。圖 7.1(c)為左側車道與目前行駛車道車輛偵測結果，圖 7.1(d)為同車道車輛行駛於較遠與三線道皆有車輛的情況，圖 7.1(e)為目前行駛車道無車輛，右側車道有車輛的狀況圖 7.1(f)為客運車輛行駛於同車道與左側車道有車輛的偵測結果。



(a)



(b)



(c)



(d)



(e)



(f)

圖7.1 中央車道車輛偵測結果 (a)只有一台車輛行駛於同車道 (b)三線車道皆有車輛 (c)右側車道無車輛 (d)三線車道皆有車輛 (e)只有右側車道有車輛 (f) 右側車道無車輛

圖 7.2(a)(b)(c)為同車道與旁車道皆有車輛時的情況，圖 7.2(d)為同車道車輛行駛於表面顏色不均勻的道路上的偵測結果。可以發現行駛於內側時影像左下正方形框並沒有出現，因為經由車道判斷發現左側已無車道，也就不需要進行偵測。圖 7.3(a)(b)為車輛行駛於外側車道的偵測結果。



圖7.2 行駛於內側車道車輛偵測結果 (a) (b) (c)兩線車道上皆有車道 (d)車輛行駛於顏色不均的道路

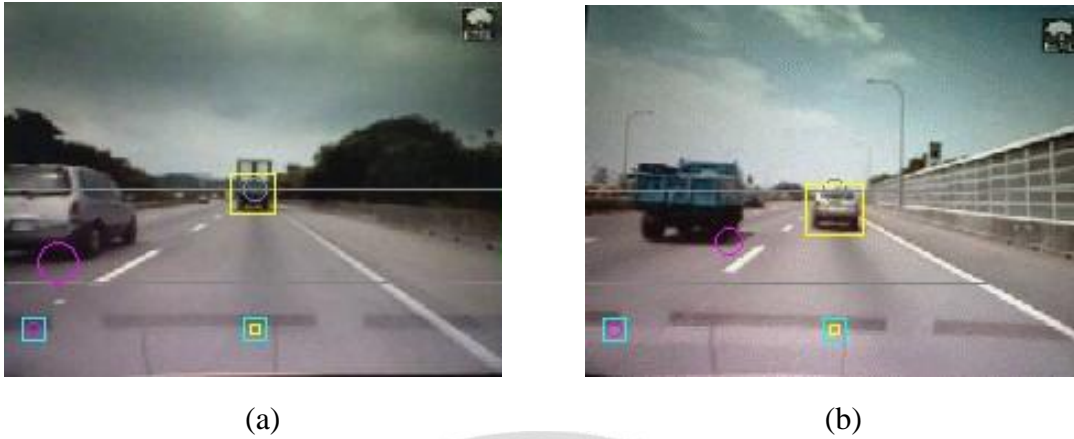


圖7.3 行駛於外側車道車輛偵測結果 (a)(b)兩線道皆有車輛情況

### 7.1.2 夜間車輛偵測結果

在夜間環境進行車輛偵測極為容易受到背景光源強度影響，圖 7.4 系列為在光源強度較強之環境下進行車輛偵測的結果。圖 7.4(a)(b)(c)皆為目前行駛車道與左側車道有車輛之狀況，圖 7.4(d)則為左側來車的車頭燈光照在前方車輛之尾端的偵測情況。其中，淺青色矩形框線為標示旁側車輛的邊界盒，黃色矩形框線為標示目前行駛車道之車輛所用邊界盒，綠色十字所標示處則為亮點物件的中心點位置。

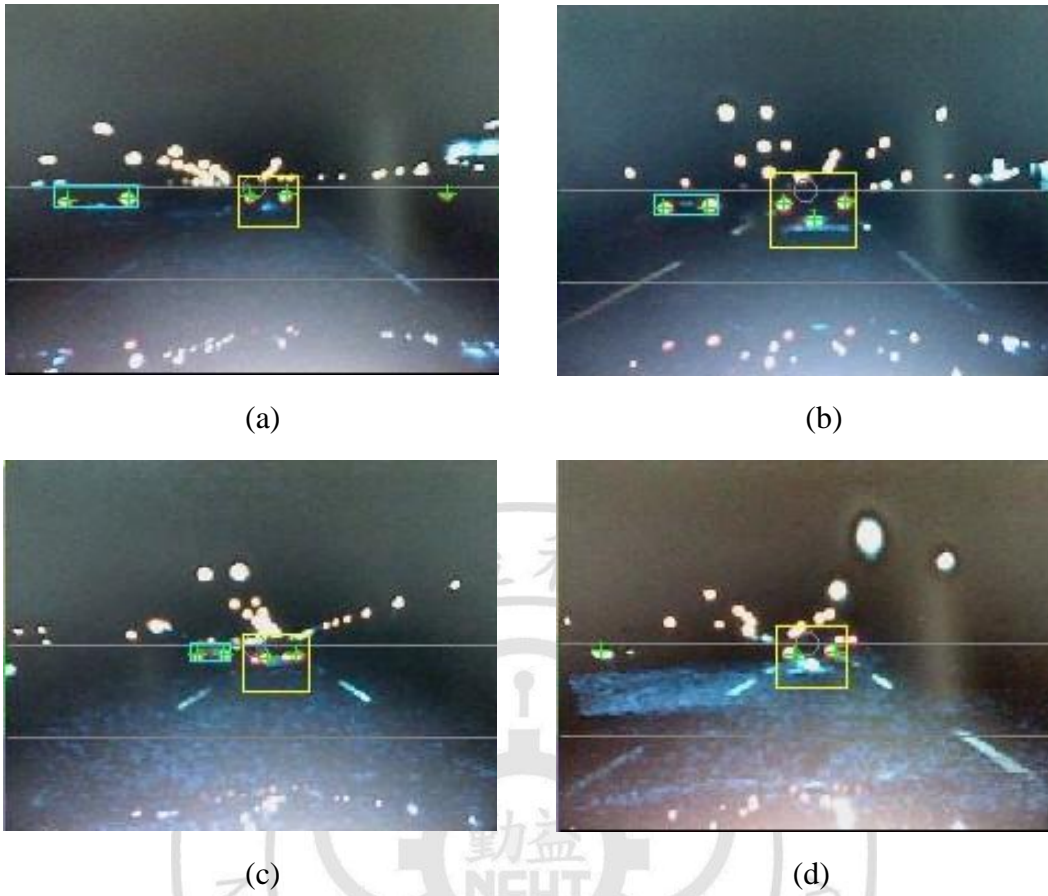


圖7.4 行駛於光源強度較強環境 (a)(b)(c)左側與同車道有車輛情況 (d) 左側來車頭燈照射於前方車尾之情況

圖 7.5 為在光源強度中等情況下車輛偵測的結果，圖 7.5(a)(b)為右側與同車道車輛存在情況，圖 7.5(c)為兩側無車輛之狀況，圖 7.5(d)為左側的兩輛車輛與同車道車輛進行偵測的結果。



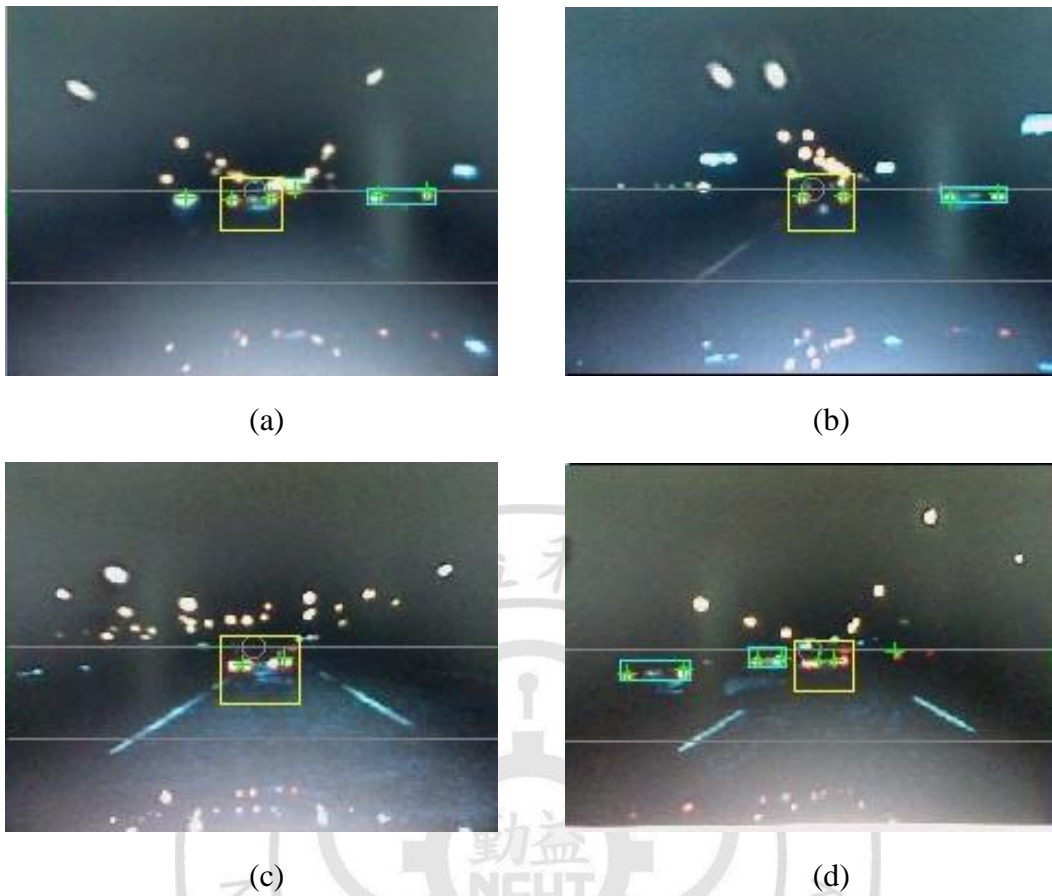


圖7.5 行駛於光源強度中等環境 (a)(b)右側與同車道有車輛情況 (c)旁側無車輛狀況 (d)左側車道行駛兩輛車輛與同車道有車輛之情況

圖 7.6 為在光源強度較弱情況下車輛偵測的結果，圖 7.6(a)為兩側車道無車輛狀況，圖 7.6(b)為左側無車輛之狀況，圖 7.6(c)為右側無車輛的偵測情況，圖 7.6(d)為目前行駛車道無車輛，右側有車輛情況。

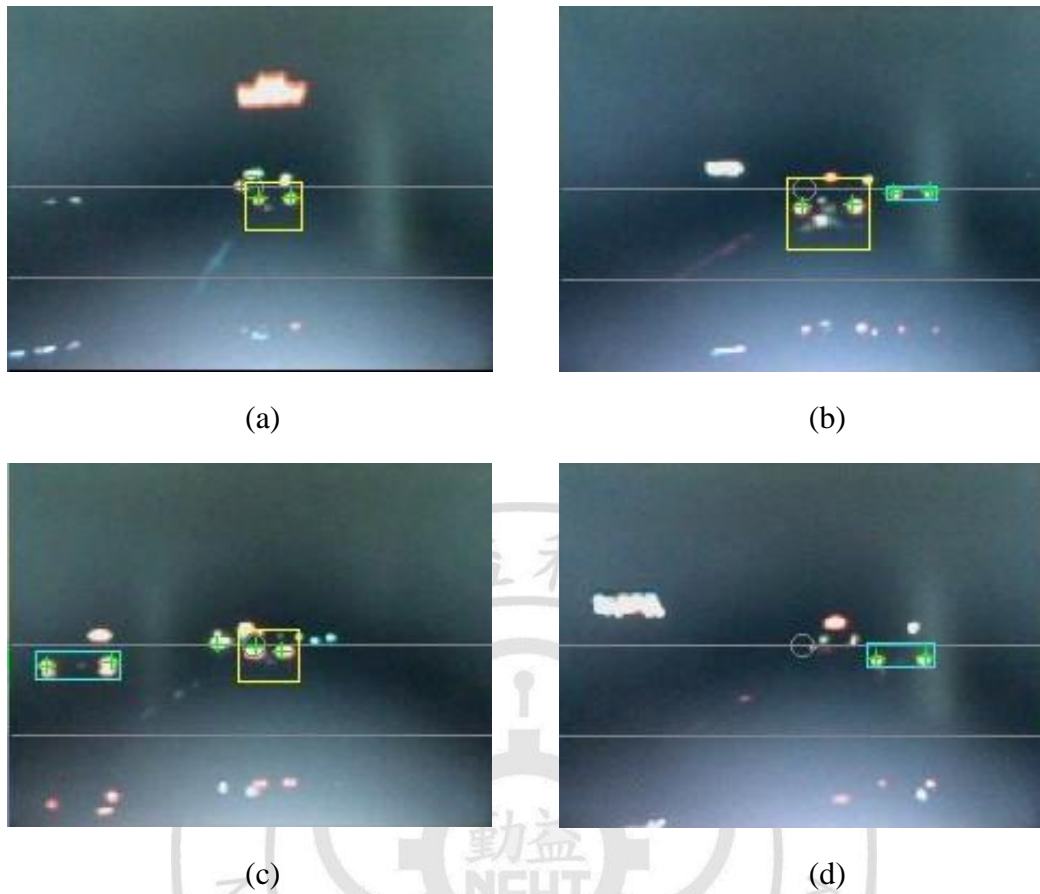


圖7.6 行駛於光源強度較差環境 (a)兩側車道無車輛狀況 (b)左側無車輛狀況 (c)左側無車輛狀況(d)只有右側有車輛情況

### 7.1.3 車輛測距與碰撞時間估測結果

圖 7.7 顯示日間情況下前方車輛距離與碰撞時間估測的結果。碰撞時間估測的結果並不是依車距所決定，而是根據前車與駕駛者車輛的速度變化產生，如此便能提供駕駛者除了車距之外的參考資訊。圖 7.8 則顯示夜間情況下前方車輛距離與碰撞時間估測的結果。





(a)



(b)



(c)

圖7.7 (a)至(c)為日間情況時前方車輛距離與碰撞時間的估測結果



圖7.8 (a)至(c)為夜間情況時前方車輛距離與碰撞時間的估測結果

本論文使用事先已拍攝的三張圖片進行校正攝影機與驗證車輛距離估測模型，將測距目標車輛(被測車輛)分別停放在十公尺、二十公尺、三十公尺處，而攝影機

架設在測試車輛(駕駛者車輛)上朝前方拍攝，擷取十公尺、二十公尺、三十公尺處三張照片，再將照片帶回實驗室使用與車輛偵測相同之模擬測試方法，進行車輛測距校正與驗證。校正與驗證之結果如圖 7.9 所示

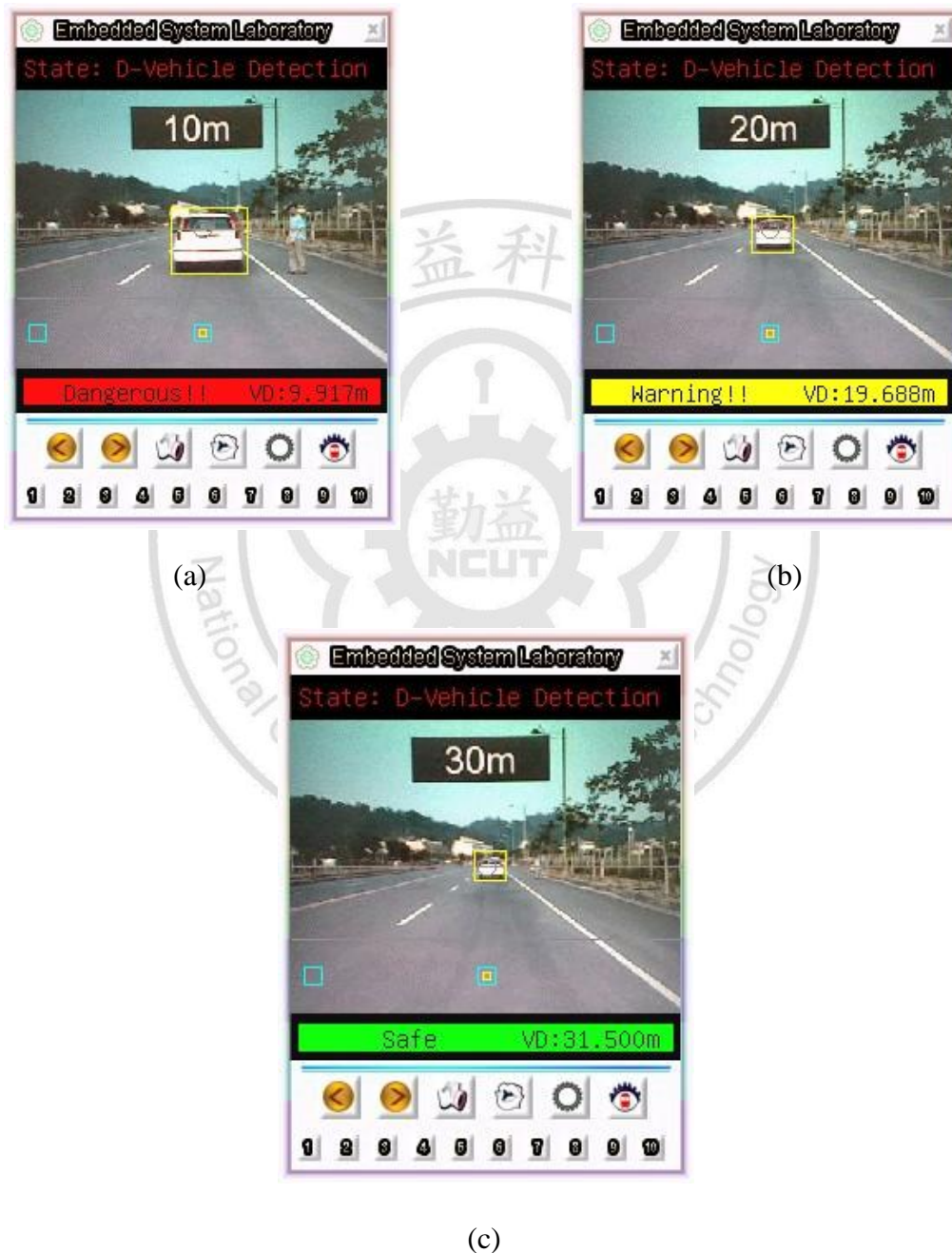


圖7.9 靜態測距驗證與校正結果 (a)十公尺 (b)二十公尺 (c)三十公尺

當前方車輛的距離過近，進入危險距離範圍時，系統會進行連續影像紀錄，其測試結果如圖 7.10(a)至(f)。圖 7.10 中一輛黑色自小客車在近距離危險超車，圖 7.10 (a)中黑色車輛已偏離其車道，因此系統之旁車道偵測無法找到正確的直角特徵。圖 7.10 (b)黑色車輛切入目前車道，系統開始追蹤該車輛並標示車距。圖 7.10 (c)旁車道偵測找到車牌直角特徵而與目前車道偵測重疊。圖 7.10 (d)旁車道偵測找到車輪直角特徵而與目前車道偵測重疊。圖 7.11 (e)與(f)為車輛進入車道後的持續追蹤。

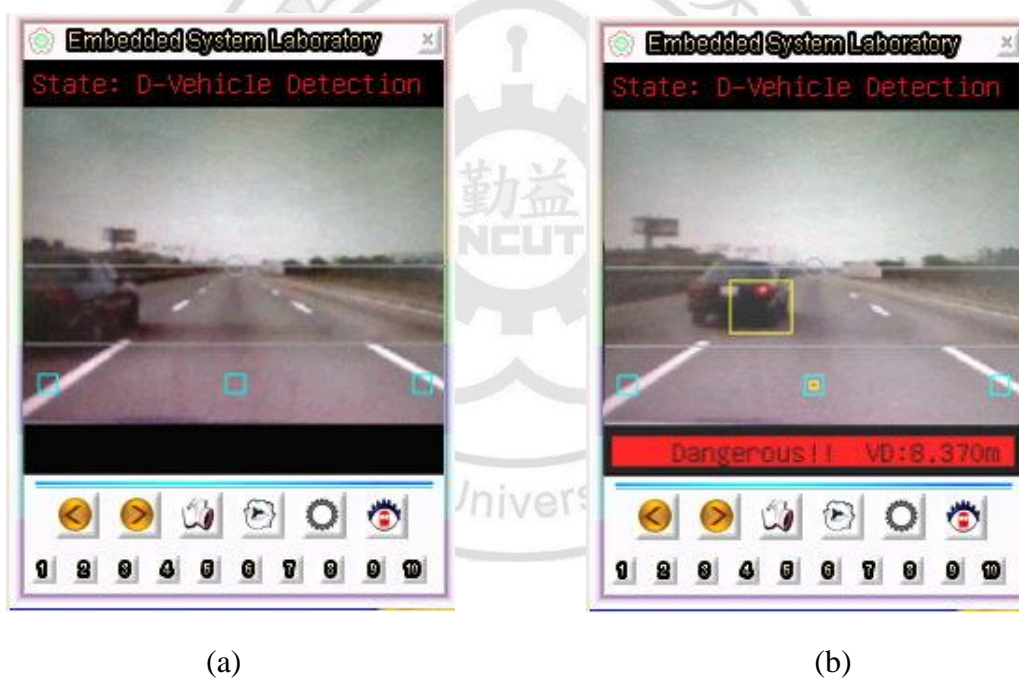


圖 7.10 (a)(b) 為前方車輛危險超車序列影像





(c)



(d)



(e)



(f)

圖7.10(續) (a)~(f)為前方車輛危險超車序列影像

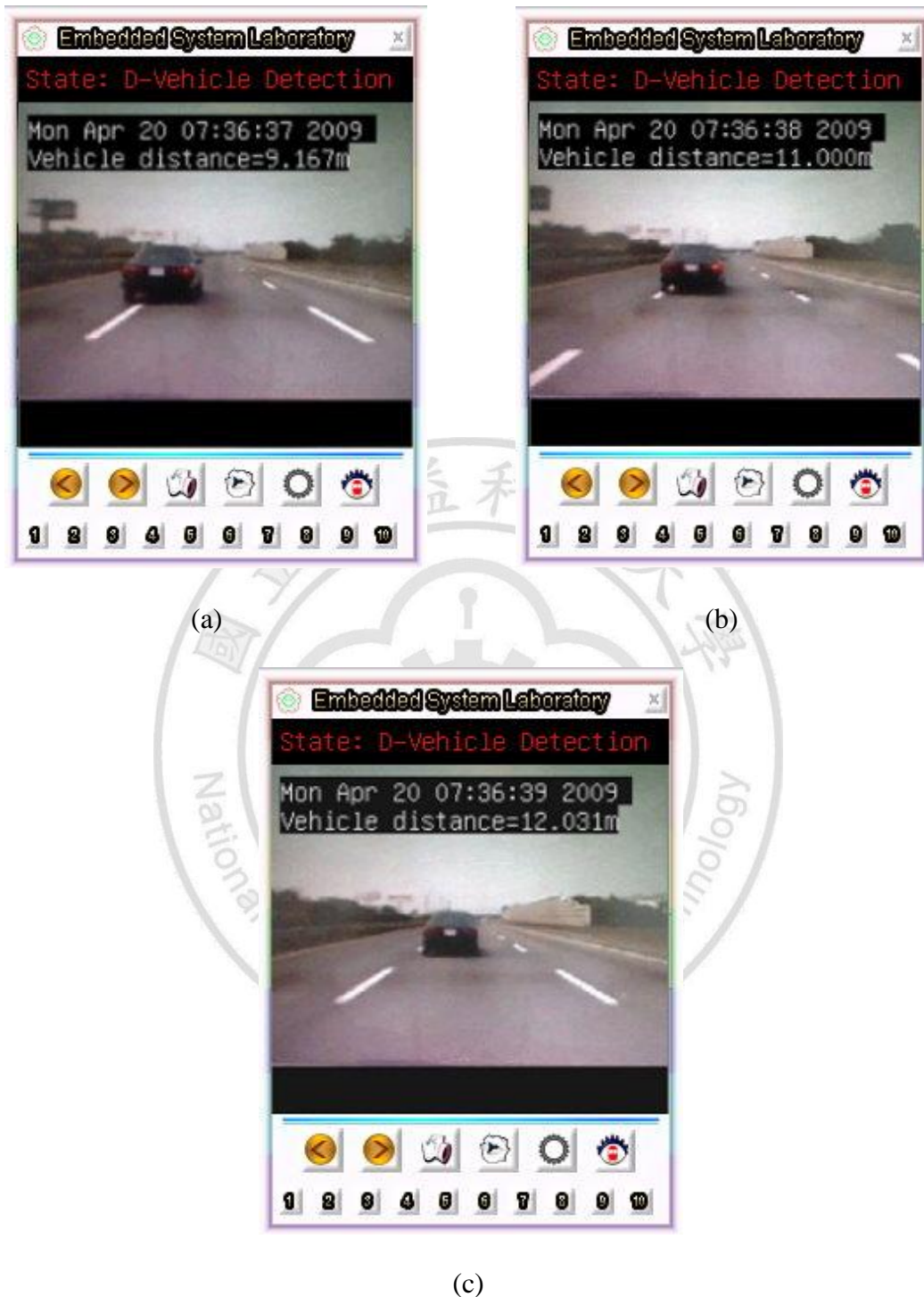


圖7.11 (a)~(c)系統在危險情況下紀錄的影像與資訊

## 7.2 實驗結果分析

表3 車輛偵測、追蹤與影像擷取之處理時間統計

	Maximum Time (ms)	Minimum Time (ms)	Average Time (ms)
Vehicle Detection in Daytime	632.84	69.99	157.16
Vehicle Tracking In Daytime	8.65	0.14	2.08
Vehicle Detection in Nighttime	436.42	111.18	188.87
Vehicle Tracking in Nighttime	5.49	0.17	1.23
Image Capture	397.52	3.44	58.92

本論文在進行車輛偵測時，使用多層次的影像處理程序。本文統計一萬次的影像擷取與車輛偵測、追蹤時間並計算平均值，每張影像擷取所花費的平均時間為零點零五八秒，日間車輛偵測演算法的執行時間平均為零點一六秒，而夜間車輛偵測演算法的執行時間平均則為零點一九秒。假如每一張影像都需要經過繁雜的處理程序，必定會造成系統龐大的運算負擔，這將容易導致系統即時性不足。在高速行駛的道路上每一秒都是會影響駕駛者的安全，車輛行駛速度以時速一百公里為例，平均一秒鐘可移動二十八公尺，也就是當駕駛者分心一秒後車輛又向前移動了二十八公尺。因此，為了在硬體資源有限的嵌入式系統達到即時性的需求，本文提出了車輛追蹤方法，讓每一張影像所花費的車輛偵測時間降低，減少系統整體的負擔。日間車輛追蹤所花費的平均時間為零點零零二秒，而夜間車輛追蹤所耗費時間平均為零點零零一秒，明顯的發現當車輛偵測演算法進入追蹤程序後，平均處理時間約為車輛偵測的七十分之一以上，而最大處理時間也縮小為



七十分之一以上。如此便能大幅降低系統運算量，提升系統執行車輛偵測的即時性。表 2 為統計一萬次車輛偵測、追蹤與影像擷取所花費的平均時間。

本論文在車輛偵測正確率統計裡，將事先拍攝的高速公路行駛影片，擷取日間與夜間內含車輛與無車輛之影像各一百張進行統計。日間的內含車輛影像中，包含了行駛於中央、外與內側車道之情況，夜間則是包含了光源的強、中與弱狀況，而偵測的區域分為目前行駛車道與旁側車道兩種。無車輛影像則是用來驗證系統在無車輛狀況時，是否會將背景或其他非車輛雜訊影響，誤判為車輛。車輛偵測正確率統計如表 3

表4 車輛偵測正確率統計

偵測統計	內含車輛 之圖像 (目前車道)	內含車輛 之圖像 (兩側旁車道)	無車輛之圖像 (目前車道)	無車輛之圖像 (兩側旁車道)
正確率 (日間)	98%	95%	99%	96%
正確率 (夜間)	96%	92%	98%	97%

觀察表 3，可以看出在不論在日間或夜間，旁車道車輛偵測的正確率較低，於日間情況中，其原因在於除了旁車道採用較簡單的找尋直角特徵處理程序外，旁車道使用固定的道路遮罩也容易將道路邊緣的雜訊(護欄與樹木等…)當作車輛影像。而夜間情況下，旁車道則是容易受到護欄的反光板等雜訊與當行駛於內車道時隔壁反方向行駛之車輛的頭燈干擾。但本論文以目前行駛車道作為主要之偵測

與測距對象，旁車道的少許錯誤，並不會影響系統的主要功能。

經由以上的統整分析，本論文所提出的影像處理架構與系統，能在日間與夜間的高速公路執行前方三線道的車輛偵測，並對目前行駛車道之車輛進行追蹤、測距與碰撞時間估測，在模擬測試與實際道路測試中也證明，系統能有效率的提供駕駛者必要的車距與碰撞時間資訊，並透過語音適時的提醒駕駛者保持車距。

### 7.3 實驗結果比較

本論文統整並與分析了許多以視覺為基礎的車輛偵測文獻，並互相比較彼此之間的優缺點。日間車輛偵測部份，Kato 等人[6]使用 MC-MQDF 進行車輛之偵測其優點在於三線道車輛之偵測率高，且能夠辨識出車輛之類型，缺點在於需耗費大量的時間去收集與剪裁車輛之影像，且正確與辨識率和分類器所學習之樣本有極大關聯，與本論文相較之下，本論文使用的演算法之成效雖不及上述文獻，但在車輛偵測的正確率上也是仍在水準之上，並且不用花費多餘的訓練時間。Chern and Shry 等人 [7]則是利用較直覺的方式實現車輛偵測，其方法與本論文所提出方法類似，但由於演算法之架構上 Chern and Shry 等人並無在抗環境雜訊部份進行加強，因此其抗雜訊能力較本論文差，並且我們也優化演算法將其實現於嵌入式系統中。

夜間車輛偵測部份，Wang [13]所提出的演算法架構與本論文部份略同，其車輛偵測正確率也相差不多，但是因影像解析度的不同 Wang and Huang 等人的車輛偵測範圍能達到四十公尺之內，而本論文則受解析度限制在三十公尺之內，但也因解析度較低讓運算量減少才能將演算法實現於嵌入式平台上。Gormer [14]提出之方法架構完整，並且在車輛偵測的正確率上也是較本論文高，其原因在於他們使用多張序列影像進行偵測，但是卻會消耗較多之硬體資源，較不適用於嵌入式系統上，但在偵測的正確性上來看此文獻所提出之演算法架構是較佳的。

## 第八章 結論

### 8.1 結論

本論文提出了一個適用於嵌入式系統的全日車輛偵測處理程序，系統主要功能為日間與夜間情況下前方車輛的偵測與追蹤，透過語音警示告知駕駛人與前方車輛保持安全距離，並在危險情況時紀錄影像、發生時間、與車距等資訊。

在日間車輛偵測部份，本文可以對同車道之車輛進行偵測，除此之外，也使用較省系統資源的方法偵測旁側車道之車輛，並且都能有良好的偵測結果。系統也利用車道線偵測產生車道遮罩，利用車道遮罩濾除不必要之雜訊，並且找出同車道之前方車輛，如此可以減少非同車道之車輛或雜訊的干擾，正確的偵測前方車輛。而夜間車輛偵測部份，本文可以同時偵測同車道與旁車道之車輛，並對同車道之車輛進行測距與碰撞時間估測，並使用車道線偵測與中線區分法把不同車道之車輛區分，分別進行偵測，防止不同車道之車輛相互干擾的問題發生，讓同車道與旁側車道之車輛皆能被系統給偵測。為了降低系統運算量，本論文提出車輛追蹤，在日間情況中使用光流法與平均值絕對誤差法進行追蹤，光流法用來調整已偵測車輛之邊界盒邊緣，用平均值絕對誤差法作為車輛追蹤之啟動與終止條件，而夜間車輛追蹤部份，則是直接使用平均值絕對誤差法判定前一張與後一張影像之變動差距，使用原邊界盒直接進行追蹤框選。日間與夜間所使用的追蹤方法皆可大幅降低車輛偵測之運算量，也有良好的追蹤效果。

本文提出的車輛偵測架構，在日間情況下能排除路面標示線、道路的色差區塊、高架橋之陰影、路面接縫等...干擾因素，正確的偵測車輛。而夜間情況下，雖然車輛偵測的執行效能容易受到光源之影響，但在路燈、道路標誌、旁車車燈、道路線反光板等...外在光源存在時，系統也能無誤的偵測車輛。

本論文所提出的車輛偵測架構經由測試後證實，在日間與夜間車輛偵測、車

道線偵測與車輛追蹤上都得到良好的測試結果。本論文所建構的系統使用較為直覺的方法且具有低運算量的優勢，適合運用於嵌入式平台上實現。

## 8.2 現存問題分析與未來研究方向

本論文所實現的系統，在多數的情況下皆能正確的偵測車輛並在影像中標示其位置，但仍有些待改善的問題存在於本系統中。日間情況中，對於大型車輛(如大貨車、客運…等)與特殊車輛(如油罐車、工程車輛…等)，由於其車身的寬高比例與一般小型車不同，通常會忽略掉車輛上邊界之區域，無法完整的用邊界盒將此類大型車輛框選。文獻[7]中利用水平邊緣偵測強化輪廓，找尋車輛上邊界的位置，但此方法容易受到像是橋梁、山或雲等...背景影響，除非背景近似於靜態影像之天空，否則找尋的車輛上邊界位置也不盡然正確。車輛的上邊界位置並非車輛偵測中的重要資訊，但在研究觀點看來也是值得探討的問題之一。而夜間情況下，在偵測大型車輛與特殊車輛時，因為它們的成對車尾燈比例與一般小型車略有不同，所以在進行車輛偵測時，容易造成錯誤的偵測結果。文獻[23]則是使用車尾燈與車底陰影兩個特徵進行夜間的車輛偵測，先透過車尾燈找尋車輛的大略位置後在搜尋該車輛之陰影確認底部位置偵測車輛，此方法能較正確的偵測出大型車輛，但是遇到光源昏暗的情況下仍是會產生誤判。

在車道線偵測部份，本論文使用直線方程式進行偵測與繪製，因此在直線道路上會有不錯的效果，但對於彎道的車道線偵測所繪製的車道遮罩會無法與實際車道的尾端相符合，如此將會造成遠方車輛的特徵遺失或標示錯誤，影響車輛偵測的正確性。文獻[24]中使用二次曲線近似式取代直線方程式來進行車道線偵測，但對於彎道幅度較大的情況，仍需使用二次以上的曲線近似式，但這樣將會消耗更多的系統效能。ROI 定義的部份，本文是根據車輛在影像中可能會出現的位置自行定義 ROI。自行預設 ROI 可以減少許多計算的時間，但是這樣會讓車道線的

特徵減少，並且當攝影機更換或影像解析度有改變時就必需重新設定預設 ROI，因此演算法的適應性與移植性會較差。文獻[24]提出了在車輛偵測前執行車道線偵測，並藉由車道線偵測的結果定義 ROI，如此便能解決適應性較差之問題，但是這樣車道線偵測的所處理之影像大小較大，將會耗費較多系統的運算效能。

車輛追蹤的方法本文主要是使用 MAE 計算前一張與下一張影像之變化程度，來作為追蹤程序的啟動或關閉的觸發條件。日間情況下，加入光流法去微調框選車輛影像的邊界盒。夜間情況下，則是直接使用上一張影像之邊界盒框選車輛影像。但是，如果一開始便已標示錯誤，MAE 進行判斷後仍持續追蹤，系統也無法立即發現，只能依賴最後的終止條件停止追蹤程序。未來的研究將會朝向使用路徑估測、樣板比對、機率統計等...相關方法改善追蹤的程序。

本論文的測距範圍僅侷限在三十公尺之內，其主要原因在於影像解析度的不足，當系統偵測的車輛超過三十公尺時，車輛的特徵搜尋已經沒有如近距離時精確，每一列像素對應的距離誤差也會越來越大，因此提高影像解析度是改善測距範圍過短的最合適方法。不過，影像解析度的提升會增加系統運算量之負擔，但解析度過低則又會導致正確性下降，因此除了提升嵌入式系統的硬體資源之外，這一部份仍是有許多難以取捨與值得思索之處。

本論文以日間與夜間時段之高速高路作為系統實測的環境，並且將複雜且運算量高之演算法與處理程序精簡成為適合執行於嵌入式系統的程式。讓以視覺為基礎的駕駛者輔助系統更近於商品化。在未來，我們將會考量更多樣的環境因素與功能，像是雨天、一般道路環境、車道線偏移等...，來優化本系統。

## 參考文獻

- [1] TRI 拓璞產業研究所, “智慧型車輛引爆車用電子新發展,” *TRI 產業專題報告* -132, Dec. 2008.
- [2] 台灣交通部統計處, “臺灣地區高速公路交通事故概況統計,” from [www.motc.gov.tw/mocwebGIP/wSite/public/Attachment/f1199704440845.doc](http://www.motc.gov.tw/mocwebGIP/wSite/public/Attachment/f1199704440845.doc) ,2006,
- [3] D. Ye, Y. Q. Yu, and B. Zenq, “Research of Classification Based on Extenics,” in *Proc. IEEE Int’l Conf. on Computational Intelligence and Multimedia Applications*, pp. 129-132, 2007.
- [4] G. P. Stein, O. Mano and A. Shashua, “Vision-based ACC with a Single Camera: Bounds on Range and Range Rate Accuracy,” in *Proc. IEEE Int’l Conf. on Intelligent Vehicles Symposium*, pp.120-125, Jun. 2003.
- [5] Z. Sun, G. Bebis, and R. Miller, “On-road Vehicle Detection: A Review,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 649-711, May 2006.
- [6] T. Kato, Y. Ninomiya, and I. Masaki, “Preceding vehicle recognition based on learning from sample images,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 252-260, Dec. 2002.
- [7] M. Y. Chern and B. Y. Shyr, “Locating nearby vehicles on highway at daytime based on the front vision of a moving car,” in *Proc. IEEE Conf. on Robotics and Automation*, pp.2085-2090, Sep. 2003.
- [8] H. Y. Chang, C. M. Fu, and C. L. Huang, “Real-time vision-based preceding vehicle tracking and recognition,” in *Proc. IEEE Int’l Conf. on Intelligent Vehicles*

- Symposium*, pp. 514-519, 2005.
- [9] A. Giachetti, M. Campani, and V. Torre, "The Use of Optical Flow for Road Navigation," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 1, pp. 34-48, 1998.
- [10] M. C. Lu, C. P. Tsai, M. C. Chen, Y. Y. Lu, W. Y. Wang, and C. C. Hsu, "A Practical Nighttime Vehicle Distance Alarm System," in *Proc. IEEE Int'l Conf. on System, Man and Cybernetics*, pp. 3000-3005, 2008.
- [11] Y. L. Chen, Y. H. Chen, C. J. Chen, and B. F. Wu, "Nighttime Vehicle Detection for Driver Assistance and Autonomous Vehicles," in *Proc. 18th IEEE Int'l Conf. on Pattern Recognition*, pp. 687-690, 2006.
- [12] Y. L. Chen, C. T. Lin, C. J. Fan, C. M. Hsieh, and B. F. Wu, "Vision-based Nighttime Vehicle Detection and Range Estimation for Driver Assistance," in *Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics*, pp. 2988-2993, 2008.
- [13] C. C. Wang, S. S. Huang, and L. C. Fu, "Driver Assistance System for Lane Detection and Vehicle Recognition with Night Vision," in *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems*, pp. 3530-3535, Aug. 2005.
- [14] Gormer, S. Muller, D. Hold, S. Meuter, M. Kummert, A. , "Vehicle recognition and TTC estimation at night based on spotlight pairing," in *Proc. IEEE Int'l Conf. on Intelligent Transportation Systems*, pp. 1-6, 2009.
- [15] S. Y. Kim, S. Y. Oh, J. K. Kang, Y. W. Ryu, K. Kim, S. C. Park, and K. H. Park, "Front and Rear Vehicle Detection and Tracking in the Day and Night Times using Vision and Sonar Sensor Fusion," in *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems*, pp 2173-2178, Aug. 2005.
- [16] P. Y. Hsiao, C. W. Yeh, S. S. Huang, and L. C. Fu, "A Portable Vision-Based



- Real-Time Lane Departure Warning System: Day and Night,” *IEEE Trans. on Vehicular Technology*, vol. 58, no. 4, pp.2089-2094, May 2009.
- [17] P. Y. Hsiao and C. W. Yeh, “A portable real-time lane departure warning system based on embedded calculating techniques,” in *Proc. IEEE Conf. on Veh. Technol.*, pp. 2982–2986, May 2006.
- [18] M. B. Ahmad and T. S. Choi, “Local threshold and Boolean function based edge detection,” *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 674–679, Aug. 1999.
- [19] 李彥鋒, 郭英哲, “以視覺為基礎之嵌入式車輛偵測系統,” 國立勤益科技大學, 2009年7月.
- [20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice-Hall, Inc., 2002.
- [21] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-9, no.1, pp. 62-66, Jan. 1979
- [22] S. Naqumo, H. Hasegawa, and N. Okamoto, “Extraction of Forward Vehicles by Front-mounted Camera using Brightness Information,” in *Proc. IEEE Int’l Conf. on Electrical and Computer Engineering*, pp 1243-1246, May. 2003.
- [23] 鄭凌軒, 陳遵立, “DSP-Based 之車路視覺系統之研究,” 國立中山大學, 2005年
- [24] 賴則全, 吳炳飛, “基於電腦視覺之即時穩健的泛型障礙物與車道偵測行車系統,” 國立交通大學, 2004年
- [25] Alcantarilla, P.F., Bergasa, L.M., Jiméñez, P., Sotelo, M.A., Parra, I., Fernández, D., “Night Time Vehicle Detection for Driving Assistance LightBeam Controller”, In. *IEEE Proceedings of the Intelligent Vehicles Symposium 08*, pp. 291-296, Eindhoven, Netherlands, 2008.
- [26] 蔡文, 陽春燕, 林傳初, “可拓工程方法,” 全華科技圖書, 1987.

## 作者簡歷

### 簡歷

作者陳璿文，出生於民國七十五年九月十九日，臺灣省高雄縣人。民國九十三年畢業於市立中正高工電機科。民國九十七年畢業於國立勤益科技大學電機工程系，取得學士學位。同年進入國立勤益科技大學電機工程系碩士班，於民國九十九年七月完成學業，取得碩士學位。研究興趣包括影像處理、RFID、ARM-based 嵌入式系統。



### VITA

Hsuan-Wen Chen was born in Kaohsiung, Taiwan, R.O.C. on September 19, 1986. He received the B.S. degree in Electrical Engineering from National Chin-Yi University of Technology, the M.S. degree in Electrical Engineering from National Chin-Yi University of Technology. His research interests include image processing, RFID and ARM-based embedded system, etc.

### **Publication:**

- [1] Y. C. Kuo and H. W. Chen, "Vision-based Vehicle Detection in the Nighttime," IEEE Int'l Conf. on Computer, Communication, Control and Automation, vol. 2, pp.361-364, May 2010.

### **榮譽紀錄**

- 國立勤益科技大學九十八學年度電資學院實務專題製作賽：研究所組 第一名
- 九十八學年度全國微電腦應用系統設計製作：研究所組 佳作
- 2010 全國大專校院嵌入式系統設計競賽：創意應用組 設計完整獎

