

國立臺灣大學生農學院生物產業機電工程學系

碩士論文



Department of Bio-Industrial Mechatronics Engineering

College of Bioresources and Agriculture

National Taiwan University

Master Thesis

基於立體視覺之即時障礙物追蹤與避障方法

Real-time Obstacle Tracking and Collision Avoidance Methods

Based on Stereo Vision

莊凱強

Kai-Chiang Chuang

指導教授：林達德 博士

Advisor: Ta-Te Lin, Ph.D.

中華民國 102 年 6 月

June 2013

誌謝



在生機所求學這兩年說長不長說短不短，能順利取得碩士學位實為可貴，心中除了感激還是感激。能完成這篇論文首先感謝我的指導教授林達德老師，在我碩一撞牆期時給我莫大的鼓勵，並不厭其煩的於求學階段點出許多小缺失以建立日後嚴謹的研究態度，更提供了 405 的優質學習環境，使實驗室成為知識的黃金屋。除此之外，特別感謝答應前來擔任口試委員的江昭醴老師和林聖泉老師抽空給予指導。

兩年的研究生活中，感謝實驗室的夥伴們：十分感謝安智學長無論在計畫執行、研究方法、做人處事方面都是個能夠讓我學習並欽佩的模範。感謝俞州、昶志、嘉鴻學長們在口試前的這幾個月中，回想起你們那些名言錦句真是受用無窮。感謝小六、笑點、宗誠學長姐，你們的學習態度讓我時時警惕且不敢懈怠。感謝 Flora 能給予英文上的諮詢，簡直就是一部活字典。感謝韋太在我遇到瓶頸時一同殺出演算法的突圍，總是帶給實驗室無限的歡樂氣氛。感謝同屆的志軒在每個熬夜的夜晚裡總是一同奮戰，並在由田比賽中隻手遮天擊敗對手。感謝大宇、金毛、庭宇在實驗時的幫忙，因為有你們讓我在實驗室更快樂。

另外，感謝同屆的渝澧、佳倫、嘉臨，你們是我這兩年生活中不可或缺的重要元素，一起娛樂、一起分享，讓我的求學生涯中增添了許多色彩。同屆的明宏、小麥、哭哭以及許多生機系義氣相挺的人，在艱困的研究歷程中仍互相鼓勵互相支持，有你們真好。

最後，感謝我的家人，在生活上能給予莫大的經濟及精神支持，讓我無後顧之憂。

要感謝的人太多了不及備載，感謝在這兩年中出現的你們，是你們成就了我。

中文摘要



本研究以兩顆 CCD 攝影機建構一套雙眼視覺立體影像系統藉以作為車輛與農業環境監控預警系統之應用。此系統將可計算出影像中像素的三維資訊，然後利用此資訊投影至上視圖進行障礙物偵測，再設定幾何限制將障礙物標定於畫面中。針對偵測到之移動障礙物影像，以巴氏距離 (Bhattacharyya distance) 進行結合距離與色彩特徵的計算，再匹配以達到追蹤目的，追蹤到目標物除了可計算取得運動的行進資訊與速度外，還可依此判斷該障礙物之動態行為作為警示作用。除此之外，追蹤到的障礙物其座標位置將會被紀錄並利用卡爾曼濾波器 (Kalman filter) 進行軌跡預測。此外，先透過立體視覺對前方資訊進行地圖的建立，再藉由 A* 路徑規劃演算法達到導航與避障的作用。綜合立體視覺系統所得到的環境資訊，以抬頭顯示器之概念加以整合與顯示作為系統的避障模式。經過實驗驗證，本系統已能成功應用於車輛導航和農業的即時環境監測，透過提出的適應性巴氏距離和交叉比對方法在不同場景皆能達到 95% 以上的追蹤準確率，軌跡預測誤差平均約 25 公分。在路徑規劃演算法 A* 的應用下，進行區域式地圖搜尋，其搜尋範圍平均約 49% ，經實驗驗證能有效應用於即時導航。

關鍵詞：立體視覺、巴氏距離、卡爾曼濾波器、A* 演算法

Abstract



In this study, a dual camera stereo vision system was built to apply in the pre-crash warning system of vehicle and agriculture environment monitoring. Based on this stereo vision system, the three-dimensional information of each pixel in the image can be estimated. Then, this information was projected to top-view so as to detect and mark obstacles according to some geometric limited. For these detected moving obstacles, we combined depth and color features and calculate Bhattacharyya distance to match different obstacles between the two continuous images in the video sequence. While the targets have been tracked, the motion models will be built. Meanwhile, the state of obstacles and their speed can also be estimated which is useful for warning. In addition, Kalman filter was employed in location prediction with these models, and then we achieve navigation and obstacle avoidance by means of A * path-planning algorithm. Finally, the concept of head-up display design is applied to integrate with the above information ahead of vehicle, which can help users to make correct decision. After experimental validation, our system is capable of applying in the vehicle autonomous navigation and monitoring of agriculture. With the adaptive Bhattacharyya distance and cross matching methods, the experimental results indicate that the performance of target tracking is over 95%, and the trajectory prediction error is about 25 cm. Besides, local path planning technique can be conducted in real time vehicle navigation with 49% of map searching.

Keywords: stereo vision, Bhattacharyya distance, Kalman filter, A star

目錄



中文摘要.....	i
Abstract.....	ii
目錄.....	iii
圖目錄.....	v
表目錄.....	viii
第一章 緒論.....	1
1.1 前言.....	1
1.2 研究目的.....	3
第二章 文獻探討.....	5
2.1 立體視覺.....	5
2.1.1 攝影機校正.....	5
2.1.2 圖像比對.....	6
2.1.3 計算深度資訊.....	7
2.2 障礙物偵測.....	8
2.3 目標物追蹤.....	9
2.4 導航系統.....	14
2.4.1 路徑規劃.....	14
2.4.2 勢力場.....	18
2.5 立體視覺的應用.....	22
2.5.1 車輛上的應用.....	22
2.5.2 農業上的應用.....	24
第三章 材料與方法.....	25
3.1 系統架構.....	25
3.1.1 硬體架構.....	25
3.1.2 軟體架構.....	29
3.2 立體視覺.....	31
3.2.1 攝影機校正.....	31
3.2.2 立體視覺理論.....	34
3.2.3 圖像匹配與像差影像.....	36
3.3 障礙物偵測方法.....	38
3.3.1 建立上視圖.....	39

3.3.2 團塊法.....	40
3.3.3 設立障礙物條件.....	42
3.4 障礙物追蹤方法.....	42
3.4.1 特徵擷取與匹配.....	43
3.4.2 適應性追蹤演算法.....	47
3.4.3 速度估算方法.....	50
3.5 避障模式.....	51
3.5.1 建立權重地圖.....	51
3.5.2 卡爾曼濾波器.....	54
3.5.3 路徑規劃.....	58
3.5.4 避障標誌辨識方法.....	62
3.5.5 障礙物動態行為分析.....	64
3.5.6 警示系統與導航.....	66
3.6 系統驗證與實驗方法.....	69
第四章 結果與討論.....	71
4.1 立體視覺.....	71
4.1.1 攝影機校正.....	71
4.1.2 像差影像.....	75
4.1.3 距離估測.....	76
4.2 障礙物偵測結果.....	77
4.3 追蹤方法探討.....	83
4.3.1 不同特徵之追蹤效果.....	83
4.3.2 背景影響.....	86
4.3.3 障礙物速度估算.....	87
4.3.4 適應性追蹤方法.....	91
4.3.5 障礙物追蹤結果.....	92
4.3.6 橫向移動與追蹤效果之探討.....	96
4.4 避障模式.....	97
4.4.1 卡爾曼濾波器之探討.....	97
4.4.2 路徑規劃.....	100
4.4.3 避障標誌辨識結果.....	103
4.4.4 導航與警示系統.....	104
第五章 結論與建議.....	106
5.1 結論.....	106
5.2 建議.....	108
參考文獻.....	109

圖 目 錄



圖 2-1 Mean-shift 示意圖	13
圖 2-2 不同演算法搜尋範圍比較	17
圖 2-3 兩種 Potential fields 行為模式	19
圖 2-4 二維地圖中同時具有兩種行為的勢力場	21
圖 2-5 具有兩種行為的勢力場以及移動軌跡	22
圖 3-1 雙眼立體視覺系統	26
圖 3-2 鏡頭穩定架	26
圖 3-3 鏡頭穩定架設計圖	26
圖 3-4 雙眼視覺之可移動式車架	27
圖 3-5 攝影機架示意圖	28
圖 3-6 系統流程圖	30
圖 3-7 核心方法架構圖	31
圖 3-8 校正物映射到影像平面示意圖	32
圖 3-9 極限幾何解釋與物體距離關係之示意圖	35
圖 3-10 像素比對示意圖	38
圖 3-11 立體視覺系統之前方環境上視圖	40
圖 3-12 距離與分層影像的關係示意圖	41
圖 3-13 團塊處理後之上視圖	41
圖 3-14 追蹤方法資訊流程圖	43
圖 3-15 障礙物 (a) 含背景 , (b) 不含背景	44
圖 3-16 障礙物匹配示意圖	46
圖 3-17 障礙物匹配值散布圖	47
圖 3-18 追蹤失敗示意圖	49

圖 3-19 速度與匹配閾值之關係	50
圖 3-20 避障模式示意圖	51
圖 3-21 避障模式流程圖	52
圖 3-22 高斯平滑化係數	52
圖 3-23 膨脹運算使用的結構元素示意圖	54
圖 3-24 節點八相鄰示意圖	58
圖 3-25 進行路徑規劃的網格狀地圖	59
圖 3-26 該節點經過其八相鄰到終點之成本	59
圖 3-27 路徑規劃過程圖	60
圖 3-28 路徑規劃過程圖	61
圖 3-29 路徑規劃分類及其代表的標誌	62
圖 3-30 避障標誌辨識流程圖	63
圖 3-31 障礙物狀態示意圖	65
圖 3-32 障礙物空間平滑化前後軌跡	65
圖 3-33 立體視覺架設於車艙內之示意圖	70
圖 4-1 校正用的棋盤方格影像，擷取自 (a) 左攝影機 (b) 右攝影機	71
圖 4-2 攝影機畫面中心校正示意圖	73
圖 4-3 校正前後比較圖	74
圖 4-4 像差影像結果圖 (a) 原始影像 (b) SGBM	75
圖 4-5 立體視覺之距離估測誤差	76
圖 4-6 障礙物偵測之結果與顯示方法	77
圖 4-7 障礙物表現示意圖 (a) 簡單幾何形狀 (b) 輪廓	78
圖 4-8 立體視覺之影像序列於農場中的障礙物偵測結果	80
圖 4-9 立體視覺之影像序列於校園中的障礙物偵測結果	81
圖 4-10 立體視覺之影像序列於市區的障礙物偵測結果	82
圖 4-11 因距離相同匹配錯誤之結果	84

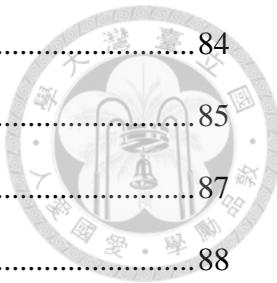


圖 4-12 因色彩資訊相似匹配錯誤之結果	84
圖 4-13 結合距離與色彩資訊之結果	85
圖 4-14 背景去除前後之結果	87
圖 4-15 系統與實際相對速度之關係	88
圖 4-16 速度估算驗證結果	89
圖 4-17 障礙物速度估算實驗結果	91
圖 4-18 適應性巴氏距離改良的結果示意圖	91
圖 4-19 立體視覺之影像序列於校園中的障礙物追蹤結果	93
圖 4-20 立體視覺之影像序列於校園中的障礙物追蹤結果	94
圖 4-21 立體視覺之影像序列於市區中的障礙物追蹤結果	95
圖 4-22 立體視覺在不同距離之水平偵測範圍	96
圖 4-23 障礙物橫向移動實驗示意圖	966
圖 4-24 卡爾曼濾波器探討	99
圖 4-25 軌跡預測與實際軌跡的比較	100
圖 4-26 左影像與路徑規劃結果	102
圖 4-27 路徑規劃之地圖搜尋百分比	103
圖 4-28 避障標誌辨識錯誤的例子 (a) 路徑規劃結果 (b) 辨識結果	104
圖 4-29 避障與導航顯示介面	105

表目錄



表 3-1 硬體設備列表	27
表 3-2 實驗相關參數	48
表 3-3 速度不同的障礙物追蹤成功率	48
表 3-4 卡爾曼濾波器演算法	57
表 3-5 路徑規劃演算法	61
表 3-6 HUD 避障模式資訊對照表	68
表 4-1 攝影機內參數	72
表 4-2 各場景中障礙物偵測的結果	79
表 4-3 不同特徵之成功追蹤率	83
表 4-4 不同影片之追蹤成功率	85
表 4-5 單因子變異數分析之報表	86
表 4-6 去除與保留背景之追蹤成功率	87
表 4-7 障礙物追蹤於各場景相關資訊	92
表 4-8 橫向移動於不同距離之追蹤結果	97
表 4-9 避障標誌辨識結果	104

第一章 緒論



1.1 前言

目標物追蹤在計算機視覺領域中是一個極為重要且具有挑戰性的主題。伴隨著電腦運算速度的提升，以及便宜且易取得的攝影機，促使學者們紛紛投入並利用自動化的動態視頻進行分析，至今已發展出許多目標物追蹤的演算法。利用影像序列追蹤分析主要包含三個步驟：偵測欲追蹤之移動物體、由兩個不同時間所捕捉的畫面進行物體追蹤、分析與辨識各障礙物之行為。目前追蹤物體技術已相當成熟，應用的領域大致可分為六大類，分別為基於運動模式的物體辨識、自動化監視系統、視訊索引、人機介面、監視交通狀況以及車輛導航等應用。

偵測及追蹤技術的應用伴隨著人並與安全息息相關，近幾年全球汽車工業發展蓬勃，加上各國交通道路建設日趨發達，使得汽車的使用量大增，卻造成交通事故經常發生，探究其原因多為駕駛人專注力不佳或是未注意其他車輛而造成擦撞的意外，為了增加駕駛人的行車安全，主要車廠如 TOYOTA 公司，所出產的新型車款皆將防撞安全機制 (Pre-Crash Safety, PCS) 列為重要配備，並開發車輛避障系統。此技術以裝設攝影機、雷達、超音波等感測器最為常見，其中車商 VOLVO 公司利用雷射感應系統開發了車輛行人偵測系統 (Pedestrian Detection System, PCS) 以取得車前資訊；日本的 Mazda 車廠已將追蹤技術應用在車道偏離警示系統 (Lane Departure Warning System) 並用以進行車輛導航；另外，先進停車導引系統 (Advanced Parking Guidance System, APGS) 不但是令人耳目一新的主動式停車輔助技術，也是目前國際大廠投入先進安全車輛的成果，其運用超音波與影像辨識之感測技術偵測車輛周圍鄰近障礙物，再配合電腦計算之路徑引導駕駛者以自動控制車輛進入停車格，以達到更安全有效率之停車動作。

想獲得環境資訊的方法有很多種，常見的距離感測器如紅外線、雷射、立體

視覺、超音波等且各有其優缺點，感測器種類的選定則端看其用途。超音波不受物體顏色及光源影響，Elfes *et al.* (1987) 將機器視覺結合聲納應用在自走型機器人建立室內環境時的障礙物偵測；軍事之海底深度的估測也常以超聲波來取得海中物體的位置資訊；而雷射的高準確度應用受環境限制較少，Garcia *et al.* (2009) 以及 Guivant *et al.* (2000) 將雷射測距儀架設於車輛上於戶外進行地圖重建並導航。立體視覺則是經由兩個攝影機組成並可得大量的環境資訊，包括物體外形，顏色，明暗，陰影等，與上述之其他種類的感測器獲得較為單純的量測資訊相比則有較豐富的環境訊息。一般而言，單顆攝影機能解決大部分簡單應用 (Liu *et al.*, 2007)，但應用層面多在於辨識、追蹤的部分，雖然特定狀況下可達到環境偵測的功能，但僅限於嚴格定義下之空間環境，面對實際應用則相對困難。立體視覺指的是裝設兩個以上攝影機，利用立體視覺原理依據兩攝影機之間距使得左右影像有所差異，分析兩張影像之關係以獲得目標深度為主的資訊，此方法須仰賴有效的圖像匹配 (Correspondence Matching) 以及極限幾何 (Epipolar Geometry) 演算法，同時也具有視覺偵測的優點。由於各種感測器之優缺點不同，近年來結合多種感測器以獲得更大效益的探討相當廣泛 (Mockel *et al.*, 2003; Wender and Dietmayer, 2007)。徐 (2011) 則結合雷射測距儀和數位單眼相機同時取得深度及色彩資訊，於場景中多個位置運用馬達原地旋轉取景，獲得不同位置中 360 度的空間及顏色資訊，將三維座標中坐標資訊的點雲 (Point Cloud) 投影至平面上將色彩與點雲結合。

利用影像序列進行物體追蹤在近年研究尤為常見，主要為找尋前後兩不同影像中物體的特徵點或具代表之物理特性做為匹配過程中的依據。Zhu (2011) 運用尺度不變特徵轉換 (Scale-Invariant Feature Transform, SIFT) 來偵測與描述影像中的局部性特徵進行匹配，以及色彩之密度函數為特徵的均值平均追蹤 (Mean-Shift) 演算法來搜尋影像中欲追蹤物體的位置。另外，Cohn *et al.* (1998) 利用光流法 (Optical Flow) 在不同影像中，亮度恆定且物體運動位移相對小且為剛體運動下等條件考量追蹤選定之物體。

農業領域中立體視覺也廣泛被使用。基於影像處理技術利用立體視覺計算作物之高度及寬度等物理特性以便計算其生長率(Chapron *et al.*, 1993)，另外 Billingsley (2005) 曾利用立體視覺偵測農場周圍之野生動物，以避免農作物遭到破壞；立體視覺在農業上的應用範疇也包含追蹤技術，但早期攝影機並不普及 Ollis 和 Stenz (1996) 僅以黑白攝影機利用演算法及影像處理找出影像中收割機的位置，來追蹤收割作物時的情形。

1.2 研究目的

在探討以上眾多文獻之後，許多感測器可獲得前方物體資訊，但各有其缺點，雷射搭配單一攝影機難以將兩者完美結合，無法給予每個座標點正確的色彩資訊，主要是因為攝影機無法提供深度資訊且必須知道畫面中每個像素所代表的實際距離，攝影機與雷射之間座標轉換關係需精確才得以準確獲得前方色彩及深度資訊。市上銷售之遊戲機使用之 Kinect 攝影機雖能同時獲得深度及色彩資訊但卻受紅外線干擾而受限於室內的使用。而紅外線則是容易受各種熱源與光線干擾且偵測的距離也有較大限制，須配合雷射使用但也造成成本上的增加。反觀，使用雙攝影機以模仿人類的雙眼視覺系統，可達成立體成像和測距的目的，且攝影機價格相較其他感測器更為廉價，故此方法較為合乎成本效益且不受限於室內或室外等特定場景。為了能將立體視覺應用於農業監測與車輛安全上，本論文主要的研究方向為三個部分：障礙物追蹤、避障方法，並以校園、市區、農場做為實驗驗證的場景。以下利用條列的方式說明本研究之具體分項項目：

1. 由 1.1 節提及的文獻可知運用單一攝影機下所發展之演算法普遍可解決部分追蹤問題，但現實情況常會遇到物體顏色相似度過高的情形，因此單顆攝影機所提供的色彩資訊不足以解決這類型問題；然而各種距離感測器則僅能提供片面的深度資訊而僅以此訊息卻無法有效追蹤多個並排的物體，而此種情形尤為常見。因此，如欲同時解決上述之特殊情形



- 下的物體追蹤，色彩與距離資訊的結合是必要的。
2. 車輛對障礙物的安全避障與導航概念源自於機器人領域，強調在當了解周圍環境狀況後進行正確的判斷跟規劃，自走車除了必須偵測前方資訊之外，亦會利用路徑規劃 (Pathfinding) 進行避障與導航的功能。然而，路徑規劃僅能將事先利用感測器所得到的環境資訊納入考量，因此先建立地圖 (Grid Map) 再根據目標方向與定位引入價值函數 (Cost Function) 作為演算法計算最小成本的規範。本文的避障方法將依據偵測與追蹤到之障礙物進行動態行為判斷以及運動模型的建立，再利用演算法達成障礙物的位置估測和路徑規劃，如此一來便有足夠資訊提供給駕駛員進行行車安全的判斷達到警報之效果。
3. 立體視覺已有眾多發展卻鮮少將立體視覺同時應用於農業車輛與行人偵測及追蹤上。因此，本研究之實驗場景主要以農業車輛安全上之應用為主軸進行系統的驗證跟效能測試。

第二章 文獻探討



2.1 立體視覺

立體視覺為兩個或多個攝影機所構成，其主要藉由兩攝影機的像素來進行特徵匹配以及計算深度值，為了減少左右兩影像特徵匹配上的難度，通常假設兩影像的光軸為水平對齊的，然而不論硬體加工多精密，都會有些許誤差，因此必須透過軟體來進行修正。

2.1.1 攝影機校正

在 3D 電腦視覺中，攝影機校正 (Camera Calibration) 指的是求取攝影機的內在參數 (Intrinsic Parameters) 與外在參數 (Extrinsic Parameters)。前者可以用來描述攝影機座標 (Camera Coordinates) 與影像座標 (Image Coordinates) 之間的轉換關係，一般而言，攝影機內部機構與鏡頭不變動則內在參數是固定的；後者則是用來描述世界座標 (World Coordinates) 其與攝影機擺放的位置和方向有關，意指如果移動攝影機，其外在參數就必須再重新校正一次。一旦有了內外參數，就可以從影像座標去推算三維資訊。

目前已發表過許多校正方法 (Zhang, 1999; Tsai, 1987)，依據使用的校正物不同而主要可分成四類：使用三維物體校正、自我校正、使用二維平面校正、使用一維長度校正。

三維物體校正主要是使用 2 或 3 個互相垂直的平面來校正，由於校正物體在三度空間須提供長寬高的幾何特性因此此種校正方法可以獲得非常精密的校正結果，但缺點是需製作校正參考物體。

自我校正方法在 1997 年由 Hartley (1997) 提出，其方法只需一個固定場景，不須利用任何校正物，而是以自然環境中至少三張影像中獲得的平面上對應的特

徵點，並分析其不同影像中的位置來校正。

二維平面校正是目前較常使用的方法 (Sturm and Maybank, 1999; Zhang, 1999)，其製作的平面須不會任意扭曲，精準的棋盤方格平面，比起三維參考物可說是簡易許多，使用者僅需拍攝該平面在各個不同角度的旋轉或位移所產生的影像，即有足夠的資訊可求得攝影機的內外參數。

一維校正物是利用一長度已知的物體，固定一端並移動另一端來估測攝影機的參數。其最大優點是對二維或三維校正當需要同時校正多台交互重疊之攝影機參數時會需要校正點同時可以讓多組攝影機看到，較不易達成，而對一維校正法來說固定端並不需要同時被多組攝影機看到，因此實際運用限制較少，此方法是由 Zhang (1999) 提出。

以上所提及的四種校正方法，主要目的都是為了得到左右兩個影像平面之間的旋轉、平移關係，再投影使之同時滿足光軸平行對齊的幾何限制以方便正確計算出立體視差。

2.1.2 圖像比對

圖像比對主要是通過找出左右兩影像之對應關係，其技術被普遍認為是立體視覺中最困難也最關鍵的問題，主要原因是左右影像有：光學失真、平滑表面的鏡面反射、投影縮減、透視失真、低紋理、重複紋理、透明物體、重疊和非連續的差異等問題。因此，是否找出正確的對應特徵點將直接影響最後計算視差的結果，而後有許多研究都是針對對應點匹配的演算法做討論。

演算法依據其搜尋對應特徵點的範圍可分為區域性與全域性，區域性的方法在比對時僅考慮特徵點附近的像素，處理速度較快但對於沒有滿足同軸幾何限制的影像，容易發生比對錯誤；全域性則是搜尋整張影像的像素，相對的計算時間較久，但適應性較好，較可修正誤差。

區域性方法中，匹配代價是圖像比對的基礎，實際是對不同視差進行灰階相似性測量，常見的有灰階差的平方 SD (Squared intensity differences)、灰階差的絕

對值 AD (Absolute intensity differences)，另外在求匹配代價時可設定一個上限值，來降低之後累加的過程中因錯誤匹配造成影響。疊加的目的是為增強匹配代價的可靠性，而根據原本的匹配代價不同，可分為 SAD (Sum of squared intensity differences) 、SSD (Sum of absolute intensity differences) 、NCC (Normalized cross-correlation) 等，其中 NCC 是進行捲積運算，直接找出特徵值。在完成累加之後即可獲得視差值，只需在一定範圍內選取累加匹配代價最好的點 (SAD 和 SSD 取最小值，NCC 取最大值)。

全域性常見的方法是 Graph Cut，Boykov (2001) 將其運用在雙眼視覺的圖像匹配上，其作法是將兩左右影像每個像素點都以向量連接，並由限制條件給予該向量一個值，構成向量網圖，將特徵匹配從原本像素比較，轉換成匹配成功的數量大小問題，進而得到的是考慮所有可能狀況後的最佳解，有效降低比對錯誤的情況。

2.1.3 計算深度資訊

雙眼立體視覺三維距離量測是基於視差原理，由圖像匹配找出之左右影像對應特徵點後可算出其視差，而攝影機在經過校正後，可獲得滿足光軸平行對齊的幾何限制，只要已知攝影機焦距 f ，以及利用經過校正及硬體上的設定所得的兩基線之間距離，根據賴 (2012) 所使用的計算視差公式可得到以左影像中特徵點的像素二維座標與真實空間座標的轉換關係，如下式

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{\|T_B\|}{d - (C_{x_L} - C_{x_R})} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f \end{bmatrix} \begin{bmatrix} x - C_{x_L} \\ y - C_{y_L} \\ 1 \end{bmatrix} \quad (2.1)$$

其中 T_B 為基線向量，也就是兩攝影機之間的向量， x 、 y 為特徵點在左影像中的像素座標。 d 為立體視覺計算出的像差值， C_{x_L} 和 C_{y_L} 分別是左右攝影機校正後畫

面中心 x 的座標 C_{y_L} 則是左攝影機校正後畫面中心的 y 座標， f 為左攝影機之焦距。

2.2 障礙物偵測



障礙物偵測是電腦視覺領域與影像處理過程技術上用來決定該物體是否屬於特定狀況下我們所關注的目標。障礙物偵測的概念是用於主動搜尋一範圍中所有資料，生活中較常應用的包含影像檢索、影像監控、汽車與行人偵測、道路安全駕駛系統等。因此要如何從影像序列中找出特定障礙物是一個須努力的方向。

機器學習 (Machine Learning) 是一種設計和分析使計算機自動學習的演算法，並從數據中自動分析獲得規律來對未知數據進行預測，有利於我們進行分類與判斷，而監督式學習 (Supervised Learning) 是機器學習中的一種技巧，可透過訓練資料中建立一個模式並依此模式推測新的物件。機器學習已有十分廣泛的應用，如計算機視覺、生物特徵識別、資料探索、DNA 序列、指紋辨識等。因此，在進行機器學習前最重要的步驟是障礙物的特徵提取，若能計算出屬於該障礙物之具有代表性特徵，將可以節省偵測流程的複雜度並提升偵測的準確度。常見的特徵如直方統計圖、物體方位、外型、紋理與色彩特徵，藉由萃取出的特徵並蒐集大量樣本便可進行監督式學習。監督式學習方法包含類神經網路 (Artificial Neural Networks, ANN)、決策樹 (Decision trees)、Adaptive Boosting (AdaBoost)、支持向量機 (Support Vector Machines, SVM) (Yilmaz *et al.*, 2006)。

類神經網路是一個能夠學習並且可總結歸納的系統，由輸入層、輸出層、隱藏層組成，運算是透過許多非線性的運算單元 (Neuron) 和位於這些運算單元之間的眾多連結，以平行且分散的方式進行運算並輸出純量結果。透過訓練樣本的校正以及統計學方法，可調整各層的權重並建立一個可表達局部空間結構的模型，其過程稱為自動學習，此模型將是一個可自動偵測識別的系統 (Rowley *et al.*, 1998)。

機器學習中，決策樹是一個預測模型，樹中的節點為每個可能的結果，而分

叉路代表可能的路徑，由樹的根節點到各個階層的葉節點僅會有單一輸出，可用於分析數據，此為條件機率下的分類，透過此方法可分次處理較複雜的情況以達最終分類結果 (Safanian and Landgrebe, 1991)。

AdaBoost 由 Freund and Schapire (1995) 提出，此種演算法是二元分類器，並利用多個弱分類器，針對訓練樣本來學習，將每一回合最高的分類器取出，依據誤差值給予適當的權重，同時也調高此分類器無法正確分類的樣本權重，如此一來在下一回合的運算中，就可以針對這些錯誤分類的樣本去找較佳的分類器，再將它取出，重複此循環，最後將這些被選出的弱分類器依權重累加，成為一個能有效辨識的強分類器。

支持向量機是一種機器學習的分類方法，主要以統計學習的方式發展而成。Boser *et al.* (1992) 提出利用非線性的分類法，藉由尋找兩個不同類別之間的最佳分隔面來做分類，此平面為非線性，稱為超平面 (Separating Hyper-plane)，此超平面通常是選擇邊限 (Margin) 最大的平行線。其先利用已知類別資料進行訓練，進而由這些訓練資料選出支持向量 (Support Vector)，並以此產生訓練模型，然後將以此模型對測試資料點進行分類。

2.3 目標物追蹤

視頻中影像序列在近年來的研究尤為常見，為了能夠將偵測到的障礙物進行運動模型建立與預測或是畫出軌跡，需匹配各個不同時間的影像所偵測到的障礙物。追蹤演算法以上一個物件資訊估測現在時刻目標物件的資訊，以及在畫面中比對 (Match) 並尋找符合目標物件特徵的區域。常見的物體追蹤方法有點追蹤 (Point Tracking) 以及核心追蹤 (Kernel Tracking)。

首先我們必須初步決定目標物的位置，原先在影像視頻中我們對影像內容一無所知，在固定攝影機的狀況下如果我們可以知道前後兩張影像之間的變化，就可以將運動中的目標找出來。光流 (Optical Flow) 是關於視頻中的物體運動檢測概念，用來描述相對於觀察者運動所造成觀測目標的運動，光流法實際是利用像

素點的強度隨時間的變化進而推斷出物體移動速度的方法，每一時間 t 皆以一個三維 (x, y, t) 向量表示指定座標的瞬時速度。Lucas-Kanade (1981) 算法就是重要的光流法之一，藉由假設在不同影像中，亮度恆定，不同影像時間間隔中物體運動位移相對微小並且為剛體運動的條件考量之下，進行影像間選定之特徵的運動模式估測與進行追蹤。若以 I 代表亮度，式 (2.2) 為亮度恆定式，對被追蹤部分像素不隨時間變化則進行微分

$$f(x, t) = I(x(t), t) = I(x(t + dt), t + dt) \quad (2.2)$$

$$\frac{\partial f(x)}{\partial t} = 0 \quad (2.3)$$

以時間連續的假設來說，在實際應用上指的是時間變化相對於物體運動需足夠小，以一維空間為例，則

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial t} = 0 \quad (2.4)$$

$$\frac{\partial I}{\partial x} = I_x, \quad \frac{\partial x}{\partial t} = v, \quad \frac{\partial I}{\partial t} = I_t \quad (2.5)$$

基於前述三種假設，實際上影像的亮度並非恆定，時間也可能因攝影機的設定不符合假設，為了求得實際速度，便利用牛頓法進行迭代，使速度趨近最佳值。擴展應用至二維空間則光流方程式如下

$$I_x u + I_y v + I_t = 0 \quad (2.6)$$

$$\nabla I^T u = -I_t \quad (2.7)$$

其中 $u = \begin{bmatrix} u \\ v \end{bmatrix}$ 為二維速度分量， $\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$ 為二維光流分量，光流方向為 $u = -\frac{I_t}{|\nabla I|}$ 。

但是對於大多數的攝影機來說，大而不連貫的現象與假設是相違背的，因此需要一個較大的窗口來捕捉大的運動，因此提出金字塔法將影像解析度降低，並由低解析度至高解析度進行追蹤，以此提升追蹤的穩定度。

假設在初始的條件上，已初步知道目標物的位置後，我們可以將連續影像中的目標物以各種方式描述，Yilmaz *et al.* (2006) 提出的通盤性介紹物體追蹤方法

介紹指出，追蹤障礙物的表示方法大致有點描述、簡單幾何圖形、物體輪廓、骨架模型等。點 (Point) 方式是其中一種描述，點追蹤是基於目標物前一個時間點的狀態包含了位置與運動的資訊，接著以點追蹤的演算法描述不同時間畫面間物件點位置的關係以便進行估測，或是解決不同畫面之間的對應關係。要如何找尋特徵點作為追蹤的點相當重要，這些點必須不易受雜訊的影響，並且在兩個正交的方向上都有明顯的導數，此點又稱作角點 (Corner)，至今已發展出許多尋找角點的方法，如 Harris (1988) 利用灰階強度的二階導數矩陣以及 2004 年 Lowe 提出的 SIFT (Scale-Invariant Feature Transform) 方法。卡爾曼濾波器 (Kalman Filter) 是一種點追蹤的方法，計算機視覺中常見其應用，其將目標位置、速度的測量值利用動態信息將原始影像中具有的雜訊濾除，並估計或預測目標的位置。應用方面如 1969 年阿波羅計畫中飛船的導航電腦便使用這種濾波器來預測軌道。卡爾曼濾波器的基礎建立在高斯雜訊，並以式 (2.8) 描述系統物理狀態的線性隨機微分方程式

$$x_k = F_k x_{k-1} + B_k u_k + w_k \quad (2.8)$$

其中 F_k 是作用在 x_{k-1} 上的動態矩陣， B_k 是作用在系統 u_k 的輸入控制模型， w_k 為過程雜訊 (Process noise) 並假設為常態分佈變異數為 Q 且均值為零，對於時刻為 k 真實狀態 x_k 的一個測量狀態 z_k 可以下式 (2.9) 來表示

$$z_k = H_k x_k + v_k \quad (2.9)$$

H_k 是觀測模型，描述狀態如何映射輸出， v_k 為觀測雜訊 (Measurement Noise) 為常態分佈變異數為 R 且均值為零，而初始狀態至觀測狀態的每一時刻雜訊均線性獨立。對於滿足式 (2.8)(2.9) 線性隨機微分系統，過程與觀測都是高斯白雜訊 (White Gaussian Noise) 條件，卡爾曼濾波器是最佳的資訊處理器，並可估算系統的最佳化輸出 (Welch, 1995)。

核心追蹤的核心 (Kernel) 指的是用來表現物體的外型，當目標物可以簡單的幾何形狀如橢圓形或是長方形表示時，特徵就可以從此區域得到，此區域稱為

樣板 (Template)，樣板是根據前一個影像中的目標物所建立的，可以是顏色或是輪廓特徵組成，進而利用影像進行比對並找尋出與目標物件樣板相似的區域。平均移動演算法 (Mean-Shift) 是一種利用已知物體樣板來進行追蹤的演算法。由於早期在進行資料分析與估測中，非參數式的資料點密度之梯度運算上，常用的方法為均值位移法，其特性是尋找資料群中的極值時，易於收斂並且其移動方向皆會指向梯度最大的方向進行一對，以求得資料群的極值。因此特性，分別由 1975 年 Fukunaga 等人，Comaniciu and Meer (1999) 提出使用在影像處理之應用中，並在文章中證明了 Mean-shift 算法在滿足一定的條件下可以收斂到最近的一個概率密度函數的穩定點。均值位移法裡，假設有一組資料 x_i ，其中有 n 個資料點，存在於 d 維度之座標空間中，此資料可用一範圍為 h 之多變量核心密度估算方法 (Multivariate Kernel Density Estimate) 可搭配核心公式 $K(x)$ 且針對資料 $x(i)$ 建立模型，多變量核心密度估算可以表示為式 (2.10)

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (2.10)$$

在計算密度梯度前，先假設

$$g(x) = -K'(x) \quad (2.11)$$

計算資料密度梯度及式 (2.10) 多變量核心密度估算公式 $\hat{f}(x)$ 之梯度 $\nabla \hat{f}_K(x)$

$$\begin{aligned} \hat{\nabla}f(x) &= \nabla \hat{f}_K(x) = \frac{2}{nh^{d+2}} \sum_{i=1}^n (x - x_i) K'\left(\frac{x - x_i}{h}\right) \\ &= \frac{2}{nh^{d+2}} \sum_{i=1}^n (x - x_i) g\left(\frac{x - x_i}{h}\right) \\ &= \frac{2}{nh^{d+2}} \times \left[\sum_{i=1}^n g\left(\frac{x - x_i}{h}\right) \right] \left[\frac{\sum_{i=1}^n x_i g\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^n g\left(\frac{x - x_i}{h}\right)} - x \right] \end{aligned} \quad (2.12)$$

上式中包含了一個簡單的均值位移向量 M_h 。由上式可再結合 M_h ，重新以 $\hat{f}(x)$

與梯度 $\nabla \hat{f}_k(x)$ 表示 M_h ，如式 (2.13)

$$M_h = \frac{\sum_{i=1}^n x_i g\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^n g\left(\frac{x - x_i}{h}\right)} - x$$

$$M_h = \frac{h^2 \nabla \hat{f}_K(x)}{2 \hat{f}(x)} \quad (2.13)$$

上式 (2.7) 可知，可以藉由正規化資料密度之梯度求得均值位移向量 M_h ，有關 Mean-shift 公式其含意如下圖 2-1 所示。均值位移法的應用式可將前一張影像所偵測到之目標物進行影像像素灰階強度或色彩資訊的直方統計圖，等到下一個時間點之影像進行均值位移法運算，找到相似的影像區塊區塊，藉以追蹤目標物。在此計算方法中，若目標物在不同影像中有大小或角度變化時，將不易偵測到其位置。

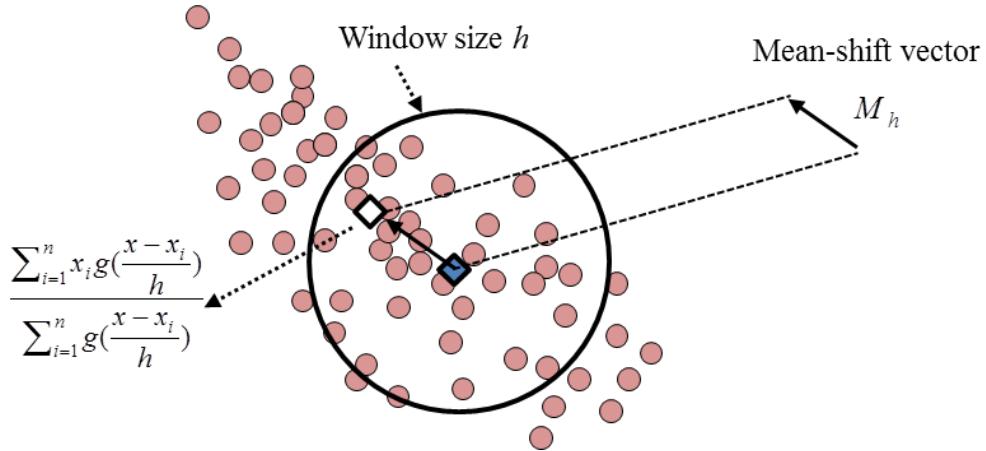


圖 2-1 Mean-shift 示意圖

為了改善此缺點，Bradski and Kaehler (1998) 將其改良，稱為 Continuously Adaptive Mean-Shift (Camshift)，針對均值位移法對於障礙物大小與角度改變而效能降低的缺點進行改良。在此方法中在決定下次搜尋視窗位置時，Camshift 方法採用的是當前目標物影像中計算其零階矩 (Zeroth Moment) 與一階矩 (First Moment) 來決定搜尋起始位置，零階矩 M_{00} 定義如式 (2.14)

$$M_{00} = \sum_x \sum_y I(x, y) \quad (2.14)$$

另外，一階矩的計算如下

$$M_{10} = \sum_x \sum_y xI(x, y), M_{01} = \sum_x \sum_y yI(x, y) \quad (2.15)$$

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}} \quad (2.16)$$



其中， (x_c, y_c) 代表均值搜尋視窗在新的影像中的起始位置。新的搜尋視窗大小是由零階矩 M_{00} 所決定，此方法最初應用在人臉追蹤上。結合了原本的 Mean-Shift 演算法以及改良的適應性區域搜尋法，適應性區域法的使用過程是引用顏色的機率函數概念，將每一個像素利用 Hue (H)，Saturation (S)，Value (V) 色彩模型中的 H 分量進行色彩直方圖的統計，並以背景投影 (Back Projection) 方法找出該目標物在下一個影像中的位置。搜尋視窗大小與當前的目標物影像之像素有關，因此可假設新搜尋視窗大小單位為 s ， s 可表示如式 (2.17)

$$s = 2 \cdot \sqrt{\frac{M_{00}}{255}} \quad (2.17)$$

根據當前目標物的影像大小可利用 s 來動態計算新的搜尋式窗大小為 s 的倍數。如此重複執行即可達到連續追蹤物件的效果。

2.4 導航系統

2.4.1 路徑規劃

車輛對障礙物的安全避障與導航概念源自於機器人領域，強調在了解周圍環境狀況後進行正確的判斷跟規劃，自走車除了必須偵測前方資訊之外，亦會利用路徑規劃 (Pathfinding) 達到避障與導航的功能。然而，學者將路徑規劃稱為最短路徑問題 (Shortest Path Problem)，是研究圖形理論 (Graph Theory) 的重要問題，各種最短路徑演算法的搜尋範圍跟方式亦有所差異，時間複雜度也不同，即

使路徑目標皆是最短路徑，但經過路段可能不盡相同，將其應用在即時資訊環境中時，當環境地圖不斷改變，路段便會進行更新，以符合即時性的需求。Hwang (1992) 將路徑規劃問題所使用的演算法分成兩類：最佳解的最佳演算法；近似解的啟發式演算法。前者顧名思義能夠找尋出最短路徑且獨一無二，是一開始最先被提出來解決最短路徑問題的方法，但是最佳解演算法的時間複雜度隨著路網的規模成指數成長，若要應用在即時導航系統上顯然不適合，因此啟發式演算法的提出解決了此問題，能在有限時間內求得次佳解，於時間複雜度與正確性之間取得平衡。路徑規劃演算法之中又以代克思托演算法 (Dijkstra's Algorithm) 和 A* 演算法 (A-Star Algorithm) 的應用最為常見 (Ferguson *et al.*, 2005; Yao *et al.*, 2010)，以下將進行此兩種演算法的說明。

Dijkstra's algorithm 是路徑規劃演算法中最早被提出 (Dijkstra, 1959)，面對最短路徑問題時首先必須建立地圖，常用的表示法有格線地圖 (Grid Map) 和四元樹區塊分割法，最後再根據目標方向與定位引入價值函數 (Cost Function) 作為演算法計算最小成本的規範，而地圖將由許多節點所組成。假設任意兩節點之間存在一條捷徑，或稱成本最少的路徑，則問題將可被簡化成最短路徑是由許許多多個節點與節點間找出的最短路徑所組合而成，這種概念又可稱為貪婪法 (Greedy Method)，指的是由各階段的區域最佳解 (Locally Optimal Solution) 累加即可在最後找出全域的最佳解 (Globally Optimal Solution)。Dijkstra's algorithm 的演算過程首先先將地圖中所有的節點分成兩個集合 S 和 Q ，分別放「已拜訪過點的集合」以及「未拜訪過點的集合」，初始狀態 S 為空集合，起點 O 的成本標籤為 0，記為 $L_{(O)} = 0$ ；初始化時其他節點的成本暫時標籤為無窮大，記為 $L_{(i)} = \infty$ 。

只要 Q 集合裡有未拜訪過的點，每次都會從集合 Q 中選出一個節點，此節點必須是集合 Q 中暫時標籤 $L_{(i)}$ 最小值者，並將此點移到 S 集合中，其餘暫時標籤便會隨時更新，直到所有的節點都從集合 Q 中轉移到集合 S ，代表每條節線 (i, j) 都已經被拓展，拓展的概念為：若存在一條從節點 i 到節點 j 的節線，則從

起點 O 到節點 i 的最短路徑由節點 (i, j) 拓展一條從起點 O 到節點 j 的路徑，路徑成本為 $L_{(i)} + w(i, j)$ ，如果小於目前 $L_{(j)}$ 值，則以 $L_{(i)} + w(i, j)$ 取代 $L_{(j)}$ 。到達終點 D 時，每條節線 (i, j) 都只被拓展一次。演算法結束時 $L_{(j)}$ 儲存的便是從起點 O 到節點 j 的最短路徑，如果路徑不存在， $L_{(j)}$ 為無窮大。以下為上述描述 Dijkstra's algorithm 的步驟：

Step 1 : 起點 O 的成本標籤為 0 ， $L_{(O)} = 0$ ，其餘非起點的節點 i 皆設為暫時標籤 $L_{(i)} = \infty$ 。 $\Gamma(i)$ 為連結節點的節點集合，令 $i = O$

Step 2 : 更新節點 i 的 $\Gamma(i)$ 中為暫時標籤的節點： $L_{(j)} = \min[L_{(j)}, L_{(i)} + w(i, j)]$

Step 3 : 在所有暫時標籤中選擇 $L_{(j)} = \min[L_{(j)}]$ 及節點 j

Step 4 : 令節 j 點為永久標籤且令 $i = j$

Step 5 : 若 $i = D$ ，即為起點 O 到終點 D 的最短路徑，如果 $i \neq D$ 則回到 Step 2

經過上述說明可知不管節點數有多少，Dijkstra's algorithm 都會拜訪所有的節點，以時間複雜度為 $O(n^2)$ 來看，計算所花費的時間會隨節點數成指數成長。

A* 演算法在 1968 年由 Hart 等三人提出，以最佳解優先搜尋法 (Best first search, BFS) 搜尋地圖節點之路徑最佳方法，在此搜尋演算法中，只要其點與終點之間有路徑存在，則保證搜尋出最少花費成本之路徑，並且導入了歐式距離作為啟發式 (Heuristic) 距離估算方法，計算中繼點與終點的相對直線距離，以此做為選擇下一節點的依據，相較於 Dijkstra's algorithm 全域性的地圖搜尋演算法，A* 演算法屬於區域式的地圖搜尋，雖無法保證找出最短路徑，但可提升運算速度並有效降低時間複雜度，在實際應用上幫助是相當大的。由下圖 2-2 可清楚的看

出白色部分兩種演算法的搜尋範圍之間的差異，而紅色線段為路徑規劃的結果。

在過去的研究，這兩種演算法的應用也不算少，總結本節的敘述說明傳統的 Dijkstra 演算法是一種全域性的地圖搜尋演算法，可保證找出最短路徑，曾應用於智慧型車輛的避障與導航 (Li *et al.*, 2009)，但運算速度較慢使得實際運用較受限制。隨著 A* 演算法的出現，試圖以較少的運算量找出適合的路徑，主要核心為將全域式搜尋改良為區域式，可提升運算速度，雖然無法保證路徑為最短，但提升運算速度的優勢使得 A* 演算法廣泛的被應用在機器人或車輛的即時導航系統 (Yershov *et al.*, 2011)。然而，對於戶外地圖隨著機器人或是車輛的移動而改變，除了須更新地圖外，也必須重新計算路徑即使 A* 降低系統運算量，然而地理上的直線距離並無法反映現實的交通狀況而造成了戶外應用的限制，為了克服此問題，Dynamic A* 演算法應運而生，簡稱 D* (Stentz, 1994)，主要優勢在於不用發現環境有些許變化就重新計算，提升了戶外環境的靈活性與適應性。針對導航中的機器人需同時移動並隨時進行路徑規劃來說，D* 演算法是較佳的選擇。然而，以相對靜態的路徑指引判斷來說，僅在特定的條件之下才執行演算法，因此 A* 演算法即可達到良好的效果。

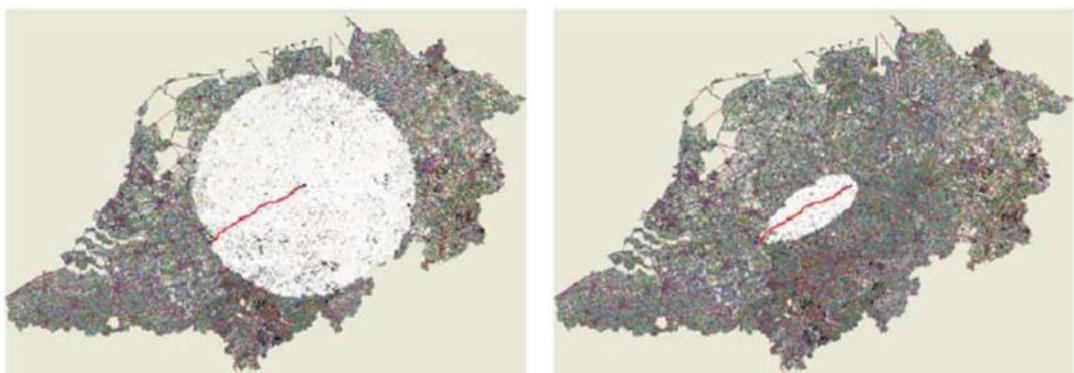


圖 2-2 不同演算法搜尋範圍比較 (a) Dijkstra's algorithm , (b) A* algorithm
(網路資源)



2.4.2 勢力場

除了路徑規劃之外，在進行地圖建置以及路徑規劃演算法運算時，通常會逐漸在空間中開發出一個無障礙的安全區域或稱為自由空間 (Free Space)。這種表示方法引入不同於價值函數的勢力場函數 (Potential Function)，而是定義地圖上不同行為的梯度向量，這些梯度向量可視為能量場或勢力場。這種方法漸漸應用在車輛行駛或機器人之避障與導航 (Koren *et al.*, 1991; Jia *et al.*, 2007)。

勢力場 (Potential Fields) 的概念是由 Andrews and Hogan (1983) 所提出的，該篇文章所提出的方法是將障礙物施加一個排斥力於移動中的機器人，而終點目標則施予一個吸引力。而後，在機器人領域逐漸成為一個廣泛被應用的路徑安全規劃方法。相較於路徑規劃演算法來說，它並不是一種量化空間的方法，而是一種利用向量或磁場概念來表示物體靠近時的趨勢，向量可幫助導航中的物體進行速度變化和定位。Potential fields 根據性質分成兩種行為，如下圖 2-3 (a) 稱為目標尋找行為 (SeekGoal Behavior)，具有吸引力的勢力場，由許多周圍向量集合指向中心圓圈的目標稱為目標勢力場，一旦進入到其勢力範圍，行駛中的物體便會朝藍色目標移動。相較於吸引力的勢力場，圖 2-3 (b) 稱作避障行為 (Avoid Obstacle Behavior)，具有排斥的勢力場，其向量集合由中心障礙物向外發散，排除障礙物影響的半徑範圍之外，向量大小隨距離中心障礙物的遠近由零逐漸增加，一旦進入到此影響範圍則會逐漸遠離中心障礙物直到向量大小為零。在進行導航時二維環境會同時包含這兩種行為，障礙物可能隨時間地點數量也會不同。

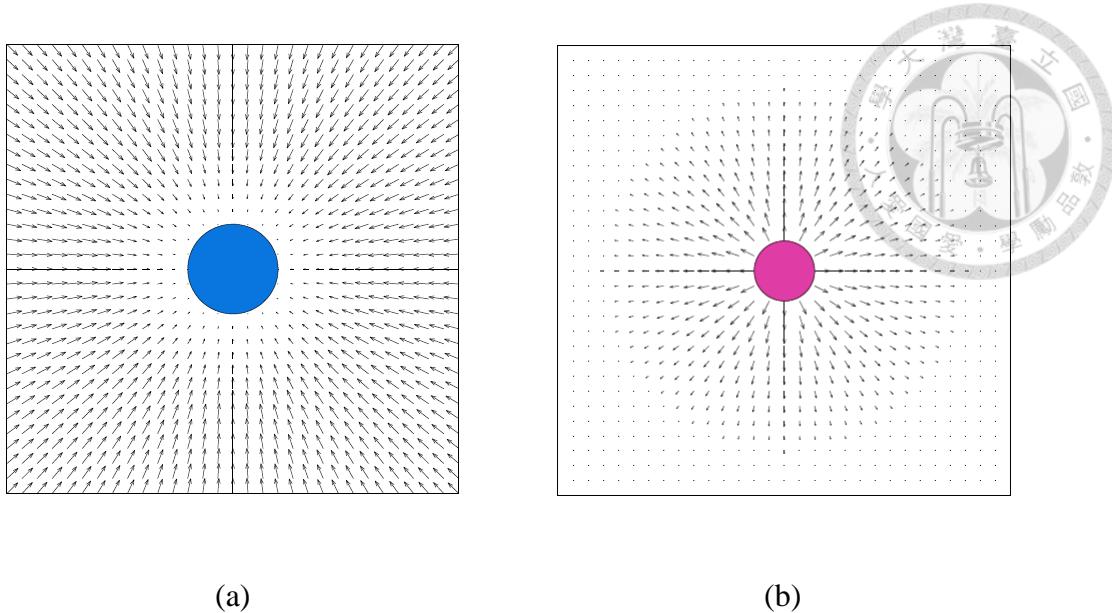


圖 2-3 兩種 Potential fields 行為模式 (a) Seek goal behavior , (b) Avoid obstacle behavior (Goodrich, 2008)

以一個物體在二維地圖進行導航的例子，我們可以想像勢力場是一種將現在位置向量 $v = [x, y]^T$ 映射到下一個位置 $\Delta = [\Delta x, \Delta y]^T$ 的梯度向量的表現。首先我們以分量 Δx 和 Δy 來定義圖 2-3 (a) 行為所產生的向量 v ，步驟如下

Step 1 : 定義 (x_G, y_G) 為目標的中心座標位置， r 為半徑。 $v = [x, y]^T$ 代表 (x, y) 位置向量

Step 2 : 計算位置與目標中心點的相位角以及距離， $\theta = \tan^{-1}(\frac{y - y_G}{x - x_G})$

$$d = \sqrt{(x - x_G)^2 + (y - y_G)^2}$$

Step 3 : 根據下列條件設定 Δx 和 Δy

若 $d < r$ 則 $\Delta x = \Delta y = 0$

若 $r \leq d \leq s + r$ 則 $\Delta x = \alpha(d - r) \cos \theta$, $\Delta y = \alpha(d - r) \sin \theta$

若 $d > s + r$ 則 $\Delta x = \alpha s \cos \theta$, $\Delta y = \alpha s \sin \theta$

到此已經建立了梯度向量與座標位置的關係，當位置與目標中心距離 d 越小，則驅動物體移動的力量越小，可由圖 2-3 (a) 看出箭頭長短的變化，當 d 小於半徑 r 時向量大小等於零。而公式中的 s 則是代表勢力場的影響範圍程度，在影響範圍最遠出有向量最大值， α 為一常數。同理，避障的行為模式也可以以梯度向量來表示：

Step 1 : 定義 (x_O, y_O) 為目標的中心座標位置， r 為半徑。 $v = [x, y]^T$ 代表 (x, y)

位置向量

Step 2 : 計算位置與目標中心點的相位角以及距離， $\theta = \tan^{-1}(\frac{y - y_O}{x - x_O})$

$$d = \sqrt{(x - x_O)^2 + (y - y_O)^2}$$

Step 3 : 根據下列條件設定 Δx 和 Δy

若 $d < r$ 則 $\Delta x = -\text{sign}(\cos \theta)\infty$, $\Delta y = -\text{sign}(\sin \theta)\infty$

若 $r \leq d \leq s + r$ 則 $\Delta x = -\beta(s + r - d) \cos \theta$, $\Delta y = -\beta(s + r - d) \sin \theta$

若 $d > s + r$ 則 $\Delta x = \Delta y = 0$

根據上面式子以及圖 2-3 (b)，可觀察出向量均以障礙物中心向外發散式的型態，向量梯度隨著障礙物影響範圍由中心向外遞減，而座標位置落在障礙物單位圓內則向量梯度趨近無限大。上式中 sign 為一個可根據不同座標位置回傳正負符號的函式， β 為一常數，可用來調整障礙物勢力場的強度。

實際上地圖會同時包含上述兩種行為且障礙物種類不只一種，因此可依據使用者根據不同障礙物設定影響之範圍。接著將探討如何結合多種勢力場，以

利應用在各種狀況，以圖 2-4 為例，一個二維地圖上擁有兩種不同行為模式所結合而成的勢力場。根據上述兩種不同行為進行的向量分析後，我們得知目標行為模式會產生吸引趨勢的 $\Delta_G x$ 和 $\Delta_G y$ 兩方向的梯度向量，障礙物行為模式對影響範圍發散式的產生向量梯度 $\Delta_O x$ 和 $\Delta_O y$ ，因此只要經過向量的合成便可進行運算，如下式 (2.18)

$$\Delta x = \Delta_O x + \Delta_G x, \Delta y = \Delta_O y + \Delta_G y \quad (2.18)$$

透過式 (2.18) 得到該勢力場的梯度向量 Δx 和 Δy ，如此便可計算出速度 $v = \sqrt{\Delta x^2 + \Delta y^2}$ 以及方向 $\theta = \tan^{-1}(\frac{y - y_O}{x - x_O})$ ，若是將物體放在如圖 2-5 的勢力場，便可看清楚出物體移動時的軌跡。

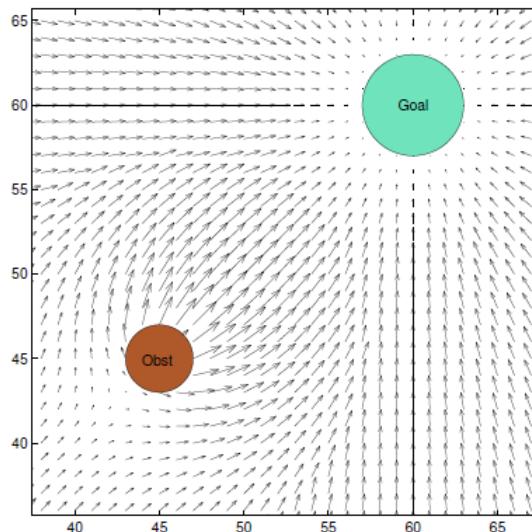


圖 2-4 二維地圖中同時具有兩種行為的勢力場 (Goodrich, 2008)

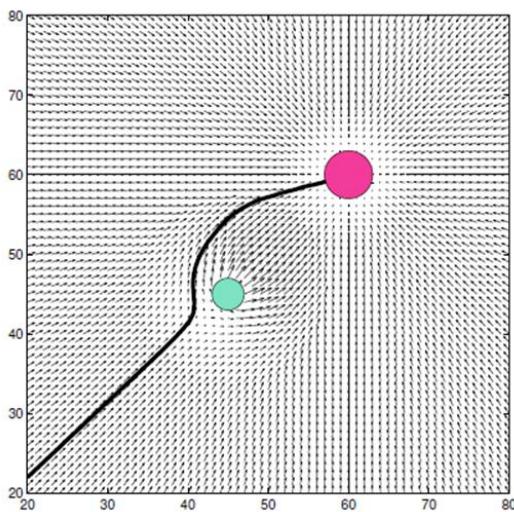


圖 2-5 具有兩種行為的勢力場以及移動軌跡 (Goodrich, 2008)

2.5 立體視覺的應用

2.5.1 車輛上的應用

近年來智慧型車輛的發展與多項產業相互關聯，並以車輛工程為核心。然而，較多的前瞻性車輛工程藉由創新電子材料、感測裝置、生技燃料、油電混合技術、燃料電池作為配備，逐漸降低傳統機械的使用比重。再加上保持行車距離裝置、抬頭顯示器、智慧型主被動安全系統、車輛防撞系統、車輛定位導航系統等發展為方向，以提升駕駛員行車安全並降低工作負擔與舒適車輛的智慧化，故許多汽車廠商皆以此為目的開發各種技術。以下將針對目前市場中已開發的技術產品以及立體視覺在車輛上應用的發展過程做說明。

在車輛領域中，所謂的立體視覺，是能夠感知空間物體的實際距離，通常以影像處理技術來達成目的，其原理是利用兩顆攝影機擷取影像，並透過影像匹配再根據幾何關係計算深度。早期的應用為單顆攝影機，所開發的技術通常會搭配其他感測器的使用，如豐田汽車的撞擊警示系統 (Pre-Crash Safety System, PCS) 是利用汽車上裝設紅外線感測器與攝影機，以達到在一定安全距離內幫駕駛人踩

剎車的動作。著名車商 Volvo 公司針對行人安全研發一套應用雷射與攝影機的行人偵測及完全主動煞車系統 (Pedestrian Detection with Full Auto Brake, 簡稱 PDFAB)，在必要時警告駕駛並預先加壓剎車。相較於單攝影機，立體視覺透過多攝影機將影像資訊由二維擴增至三維，最常見於障礙物偵測功能的開發，透過深度感知技術，獲得較精準的障礙物三維空間座標與相對位置，如 Subaru 公司所推出之雙攝影機立體視覺的系統。未來除了繼續發展低成本、高效能的安全輔助系統，並可根據外在的三維資訊，開發即時定位和地圖建構等技術，將其應用在車輛導航。

立體視覺在車輛於道路環境的應用依照技術發展的過程可分為車輛偵測、車輛追蹤、車輛行為分析 (Sivaraman and Trivedi, 2013)。早期的車輛偵測技術以裝設單顆攝影機為主，可裝設於車輛來彌補後照鏡無法看到之車身附近的死角或是特定區段的道路監控，主要技術可分為基於外觀特徵搭配機器學習方法 (Sun *et al.*, 2006) 和運動狀態為基礎來偵測 (Diaz *et al.*, 2008)。然而，基於物體於影像中的運動來偵測物體的方法較外觀特徵來的普及，以外觀來偵測車輛須仰賴具有代表性的特徵和機器學習方法，基於運動狀態的偵測方法則是透過物體追蹤、運動狀態、軌跡預測等方法。兩顆攝影機構成的立體視覺除了保留了單顆攝影機擁有的影像資訊外，更可得到深度資訊，其應用性也較高，逐漸取代了單顆攝影機。為了能更進一步分析車輛的動態行為甚至預測車輛的位置，以追蹤技術為議題的討論逐漸增加。根據穩定的偵測方法，再針對車輛於道路的運動模式使用不同的預測器進行追蹤和預測，大致可分為三類：一般車輛行駛為線性運動之下的卡爾曼濾波器 (Franke *et al.*, 2005)、用來描述非線性的狀態如轉彎的行為，則用改良前者的擴展式卡爾曼濾波器 (Extended Kalman Filter, EKF) (Barth and Franke, 2009) 以及擁有線性與非線性兩者運動模式的粒子濾波器 (Particle Filter) (Catalin and Nedevschi, 2008)。當立體視覺的偵測與追蹤技術到達一定水準後，有學者去探討車輛的運動行為，應用的範圍包含車輛在道路上的軌跡和速度 (Sivaraman *et al.*, 2011)、十字路口上車流量 (Barth and Franke, 2010) 以及不同車

輛運動模型的建立 (Kasper *et al.*, 2011)。



2.5.2 農業上的應用

隨著世界人口數不斷增長，糧食和農產品的需求量亦隨之增加，面臨此問題要如何克服將是十分重要的。在台灣的早期，勞動力為農業重要生產要素與資源之一，台灣農業勞動力曾在經濟發展過程中扮演重要的角色，隨著科技發展及技術進步，機械化漸漸取代人力的農業自動化，優點是可以解決勞工短缺問題及改善工作環境、提高農業生產力及產品品質、發展高科技農業及開發水準。立體視覺在農業上的應用及文獻不勝枚舉，若根據農業上應用的功能來分主要有：定位與偵測、追蹤與導航、量測。

立體視覺在影像中偵測目標物是最常見的應用，先利用演算法進行攝影機校正之後便可根據立體視覺理論得到正確的三維座標。一般來說，利用雙眼視覺攝影機進行偵測的主要流程依序為影像擷取、影像前處理、特徵擷取、偵測與影像分割，為了從影像中取得感興趣的區塊，特徵的擷取和影像分割是很重要的。Rovira-Más *et al.* (2005) 曾利用 Hough transform 找出作物線，也有根據作物的外觀形狀及顏色做為特徵進行標定並採收的例子 (Ollis and Stentz, 1996; Lv *et al.*, 2011)。此外，農業工具機如曳引機或中耕機在田間進行播種或翻土的工作相當普遍，尤其國外的農地通常面積較大相對工作時間長，因此有將立體視覺應用在農業車輛的導航以協助駕駛員 (Kise *et al.*, 2005; Rovira-Más *et al.*, 2004)。

近年來，為了解決台灣農地面積狹小與颱風所造成的農業虧損的問題，集約式栽種農作物方式的「植物工廠」日益發展，植物工廠透過硬體設備進行環境控制與系統整合，以便進行生長環境與生育狀況的觀察，最終目的在於全年性的生產栽培。林等人 (2012) 以立體視覺估算植物的面積與建立生長模型，並進一步分析生長速率與環境因子的關係找出最佳的環境參數。除此之外，國外也有利用立體視覺觀察溫室中植物的生長情形 (Tilneac *et al.*, 2012)。

第三章 材料與方法



3.1 系統架構

本研究主要核心分為兩部分：獲得三維點雲和障礙物追蹤與避障演算法，獲得三維點雲的方法是利用雙眼立體視覺系統，並進行像差影像的計算，在這之前必須先將攝影機進行校正，校正包含消除徑向與切向的畸變，以及透過計算獲得兩攝影機之間的旋轉平移關係，使之滿足同軸的限制。在獲得視差影像之後，再進一步利用三角幾何關係推得影像中的每個像素點真實的世界座標。

一旦立體視覺獲得三維點雲後，利用影像處理技術與特徵比對的方法進行障礙物的偵測與追蹤，將影像序列中各個感興趣的目標物找出並將其關係連結起來同時記錄其座標資訊，並判斷目標物的動態行為。接著，藉由卡爾曼濾波器可進行目標物的狀態位置預測，結合歷史與未來資訊後便可透過路徑規劃演算法進行避障，如此一來則能有效指引駕駛員做正確的行車判斷。本節將依據硬體架構與軟體上的系統流程圖做詳細說明。

3.1.1 硬體架構

本研究建立一套雙眼立體視覺系統，包含一組雙眼立體視覺機構、影像擷取卡、以及執行軟體系統的桌上型電腦。雙眼立體視覺機構由兩個影像解析度 640×480 的攝影機組成，其鏡頭焦距為 16 mm，在理想的雙眼視覺系統中，在兩個 CCD 攝影機須於同一水平上，且光軸需平行且對齊之限制下，可確保校正與圖像匹配的準確度並可減少運算時間，根據實驗經驗，每次進行實驗前必須進行鏡頭校正。為了節省校正所花費的時間，設計一個鏡頭固定機構以避免鏡頭晃動需再次校正的麻煩，如圖 3-3。影像擷取部分則是採用 USB 介面的四通道擷取卡並連結至電腦觀測左右影像，如圖 3-1、3-2，詳細規格說明請見表 3-1。



圖 3-1 雙眼立體視覺系統



圖 3-2 鏡頭穩定架

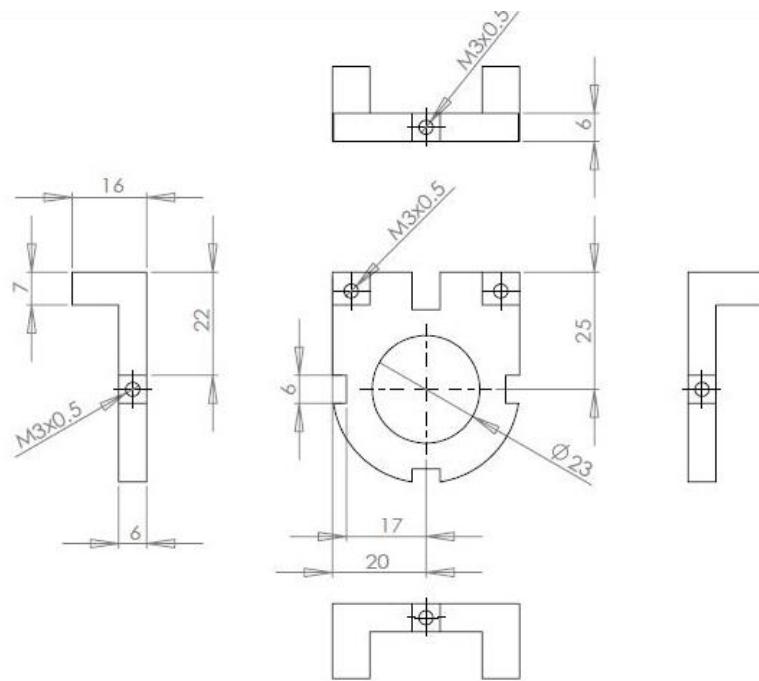


圖 3-3 鏡頭穩定架設計圖

表 3-1 硬體設備列表



設備	數量	規格說明
彩色 CCD 攝影機	1	影像解析度：640 × 480，鏡頭焦距：16 mm
影像擷取卡	1	微宇科技 4eyeDeluxe，USB 介面
桌上型電腦	1	影像擷取解析度：320 × 240，30 fps CPU：Intel® Core™ i5 3.30GHz 作業系統：Microsoft® Windows7 64 位元
		記憶體：AMD DDR3 4GB × 1 個

由於實驗多在戶外場景進行且依實驗設計的不同須調整攝影機高度，為此將設計一個可依實驗需求進行高度調整並可架設此系統之載具以利實驗的進行，如圖 3-4 所示。

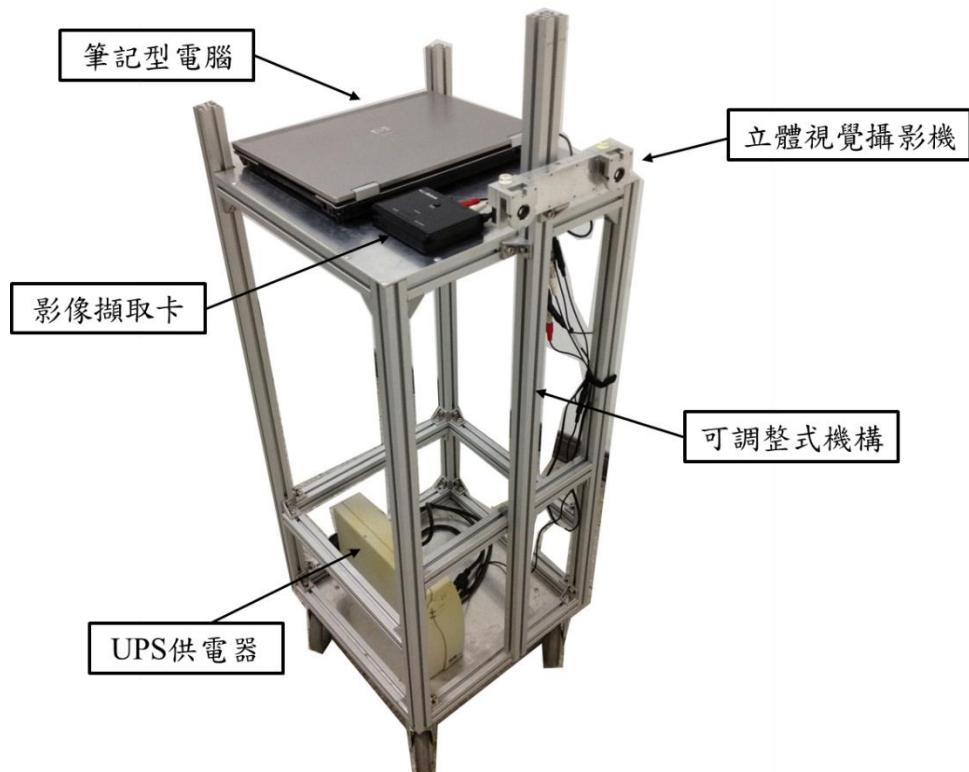


圖 3-4 雙眼視覺之可移動式車架

雙眼立體視覺計算像差值時，需介於左右兩攝影機可視之重疊範圍才可計算，由圖 3-5 可看出當攝影機間距為 T_B ，單顆攝影機的垂直可見視角為 $2\theta_v$ ，因此物體與攝影機之間的距離至少需大於 D_i ，距離 L 處則有垂直長度 I 的交集範圍， L 越大則交集範圍越大，但是相對的解析度會較低，經過幾何關係的計算可得下列式子：

$$I = 2L \tan(\theta_v) - T_B \quad (3.1)$$

$$D_i = \frac{T_B}{2 \tan(\theta_v)} \quad (3.2)$$

實際使用的攝影機， T_B 為 15 公分， θ_v 為 9.8 度，若以有效距離 $L=30$ 公尺來定義，經過計算可得 I 約為 10.8 公尺， D_i 為 43 公分。表示距離攝影機 30 公尺處可拍攝到水平方向 10.8 公尺的範圍。由於這樣的特性，軟體的撰寫上保留此彈性，讓使用者可依據實驗需求來設定須進行之最遠偵測距離，以獲得攝影機可擷取到的水平範圍。

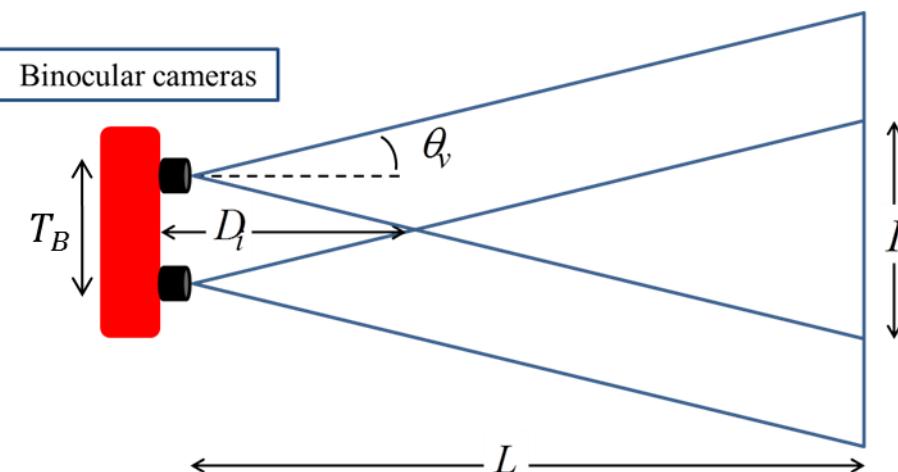


圖 3-5 攝影機架示意圖

3.1.2 軟體架構

本研究使用 Open Source Computer Vision Library (OpenCV) 函式庫，並以 Nokia[®] Qt 的 GUI 介面作為程式開發平台，搭配 Microsoft[®] Visual C++ 2010 的編譯器。

說明系統流程圖之前，在實作部分我們可依照執行之順序切割成主要四大核心，分別是立體視覺、障礙物偵測、障礙物追蹤、避障模式，如圖 3-7 所示，圖中四大核心之內部執行方法將與圖 3-6 的顏色相互對應。系統流程圖（圖 3-6）之說明如下：

- 立體視覺：

以攝影機為環境感測裝置的雙眼立體視覺系統主要目的是獲得影像中每個像素的三維資訊，假設已將攝影機進行校正，系統執行影像取得後，首先利用圖像匹配將左右兩個影像之角點特徵進行比對找出其相對關係，如此才能進一步計算影像中的每個像素所對應世界座標的三維座標，而這個過程會產生像差影像，像差影像可以做為代表影像中物體之實際距離的深度影像。

- 障礙物偵測：

障礙物偵測之技術在於如何將立於地面的物體在影像中給標定出來。偵測方法是根據前一步驟所得出的像差影像進行上視圖投影，再由上視圖進行團塊法處理，如此一來將可得到一個至數個立於地面的障礙物，又稱作團塊 (Blob)，藉此再依據設定之障礙物條件進行濾除並將其標定。

- 障礙物追蹤：

假設前一步驟已將感興趣之障礙物偵測出來，而追蹤的目的在於連結影像之時間序列的關係，這個過程又稱為影像註冊 (Image registration)，執行方法有兩個步驟，分別是特徵擷取以及比對。根據立體視覺系統，

主要有影像的色彩與距離資訊，因此將結合兩特徵 (Feature fusion) 後再進行比對以達到追蹤目的。

- 避障模式：

避障模式除了要能給予駕駛員引導，進一步做正確的行車判斷外，亦需盡可能呈現前方的環境資訊。因此，一旦追蹤到障礙物，便可判斷障礙物的狀態是逼近或遠離，亦可根據估算出障礙物的速度，進而預測障礙物未來的位置。接著，避障的主要方法是利用視圖，設定前方障礙物的權重後，再利用路徑規劃演算法達到避障目的，最後將設計一顯示介面將有助於導引駕駛員的資訊投影上去。

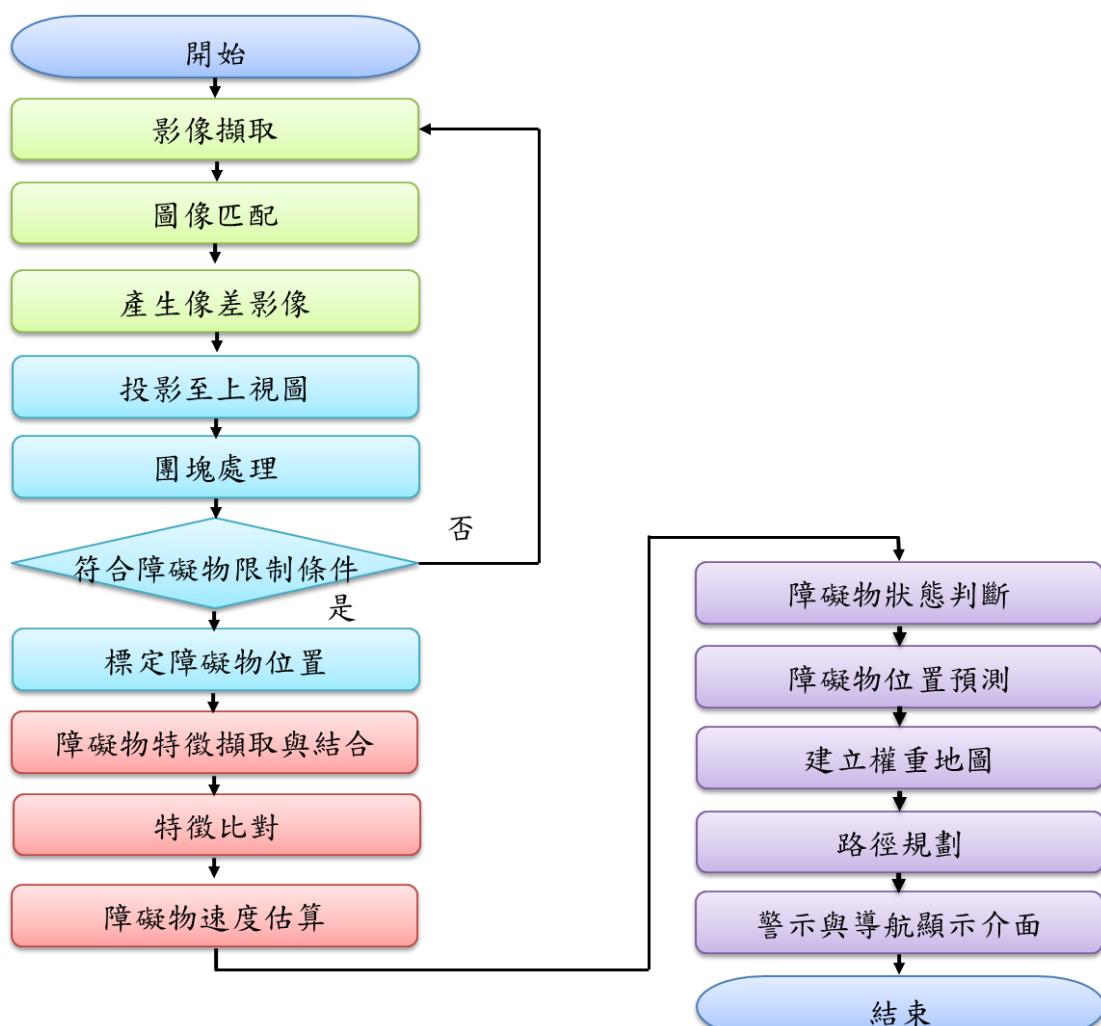


圖 3-6 系統流程圖



圖 3-7 核心方法架構圖

3.2 立體視覺

3.2.1 攝影機校正

為了讓立體視覺系統左右影像滿足光軸平行對齊的限制，本研究採用賴(2012)所開發之二維校正方法進行雙攝影機校正，主要分成幾個步驟：拍攝校正圖形、搜尋方格角點、計算攝影機內參數 (Intrinsic matrix) 和畸變參數 (Distortion coefficients)、計算主要矩陣 (Essential matrix) 和基礎矩陣 (Fundamental matrix)、重新投影左右影像，使其滿足同軸的幾何限制，這些過程可由 OpenCV 之內建函式直接運用。一旦校正完成，我們將可利用立體視覺理論計算像差影像。

- 拍攝校正圖形：

校正時，使用二維平面棋盤方格作為校正物，須注意每個方格的長寬要相同，拍攝前須確認方格的長寬都足夠精確，且方格平面不能有浮凸的現象，也就是要完全平貼於參考平面上。固定好後盡可能的以各種角度拍攝棋盤方格，也要儘量涵蓋各種角度，並確認每一個方格都出現在兩台攝影機的畫面中，由於影像周圍相較中間較容易有扭曲的現象，需使校正版填充整個畫面，以利後續的計算，如此才能求出好的係數解。

- 搜尋方格角點：

為了計算實際線條與影像中線條之間扭曲的關係，需要藉由格子板上角

點與角點之間的連線來取得。在偵測方格角點之前，因拍攝時周圍的背景與環境的光線不易控制，因此使用 Jain *et al.* (1998) 所提出的自適應閾值分割進行初步的過濾，以 3×3 周圍的平均作為閾值，大於此值設定為 255，其他設為 0，此方法針對光線不平均有不錯的效果。接著使用 Sobel 梯度矩陣進行兩個方向的微分，若兩個方向皆有較大值，則視為整數座標的方格角點。實作時可用 `findChessboardCorners` 和 `cornerSubpix` 函式完成。

- 計算內參數 (Intrinsic matrix) 和畸變參數 (Distortion coefficients) :

計算內參數和畸變參數是利用 Bradski and Kaehler (2008) 的方法做說明，如圖 3-8 所示，藉由旋轉與平移矩陣 W 可將校正物座標投影至影像平面座標：

$$W = [R \quad t] = [r_1 \quad r_2 \quad r_3 \quad t] \quad (3.3)$$

其中 R 是旋轉矩陣， t 是位移矩陣。

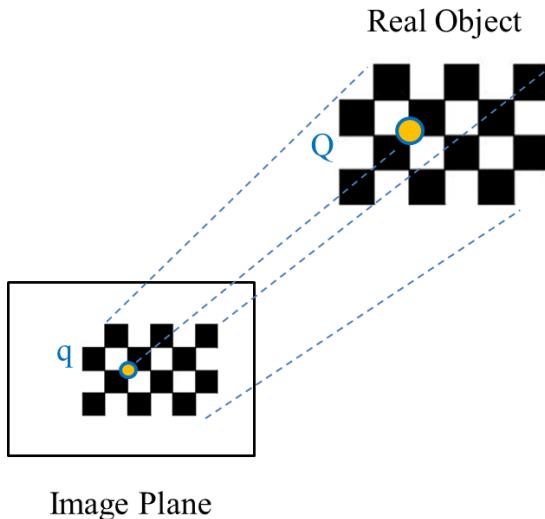


圖 3-8 校正物映射到影像平面示意圖

此外，若將攝影機的內參數矩陣 M 也需考慮進去，經過改寫後如下式

(3.10)、(3.11)

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$q = sMWQ \quad (3.11)$$



其中 f_x 、 f_y 為鏡頭分別在 x 、 y 方向上的焦距， C_x 、 C_y 為 x 、 y 方向上
的偏移量， s 為調整大小的係數。 q 為影像平面的方格角點座標， Q 為
真實世界的角點座標。如此一來，便可利用數學方法將校正後的座標帶
入推算出畸變參數的解，詳細的推導過程可參考賴 (2012)。實作時可用
OpenCV 之 `calibrateCamera` 函式完成。

- 計算主要矩陣 (Essential matrix) 和基礎矩陣(Fundamental matrix) ：

主要矩陣，或稱作外參數矩陣，包含的是左右兩台攝影機之間的旋轉與
平移等幾何關係。而基礎矩陣和主要矩陣的功能一樣，但其為像素座標
之間的關係。由兩顆攝影機拍攝的大量棋盤方格，可直接以 RANSAC
計算出基礎矩陣 (Fischler and Bolles, 1981)，但先消除了畸變後再計算
可得到較好的結果。實作時可用 OpenCV 之 `stereoCalibrate` 函式完成。
- 重新投影左右影像，使其滿足同軸的幾何限制：

獲得基礎矩陣之後，為了減少影像重新投影導致的變形，可將其拆成兩
個旋轉矩陣，使左右兩個攝影機的成像平面各自旋轉朝向前方，此時並
沒有水平對齊，因此還需利用一個旋轉矩陣 R 來達到此效果，此矩陣的
第一個向量可利用兩個攝影機投影中心之間的平移向量 T_B 帶入：

$$R = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix} \quad (3.12)$$

$$e_1 = \frac{T_B}{\|T_B\|}$$



另外， e_2 和 e_1 必須正交，因此可選擇沿著圖像平面，並和 e_1 內積為 0 的一個向量：

$$e_2 = \frac{1}{\|T\|} \begin{bmatrix} -T_{By} & T_{Bx} & 0 \end{bmatrix}^T \quad (3.14)$$

同理， e_3 必須和 e_1 、 e_2 正交，因此可以利用外積得到：

$$e_3 = e_1 \times e_2 \quad (3.15)$$

最後只需要將左右影像再乘上此矩陣 R ，即可使左右影像水平對齊。

實作時可用 `stereoRectify` 函式完成，並以 `initUndistortRectifyMap` 函式計算原始影像和校正後影像對應關係，而以後每當有新擷取的影像，便可用 `remap` 函式將其快速轉正。

3.2.2 立體視覺理論

極線幾何 (Epipolar Geometry) 為立體視覺之重要一環。此幾何關係可用來解釋雙攝影機的物體成像與實際物體的關係，如下圖 3-9 所示。其中 O_L 、 O_R 分別為左右攝影機的光學中心，其連線稱為基線 (Baseline) P 為某一目標物， P_L 、 P_R 分別為 P 在左右影像上的投影點。而 O_L 、 O_R 、 P 三點構成的面稱為極平面 (Epipolar Plane)，該平面與左右兩影像平面的交線稱為極線 (Epipolar Line)，所謂滿足光軸平行對齊的限制就是計算出兩個影像之間平移與旋轉關係之後將兩平面校正到水平對齊，使得 P_L 、 P_R 出現在同一條極線上 (Trucco and Verri, 1998)。當空間中一點投影成像於左右兩影像平面上時，就如圖中所述的三角關係。利用此三角關係，式 (3.16)、(3.17)、(3.18)，並經由攝影機校正以及硬體上的設定，可以得到攝影機的焦距以及兩攝影機間基線的距離，因此可以求出物體距離雙眼

立體視覺系統的距離。

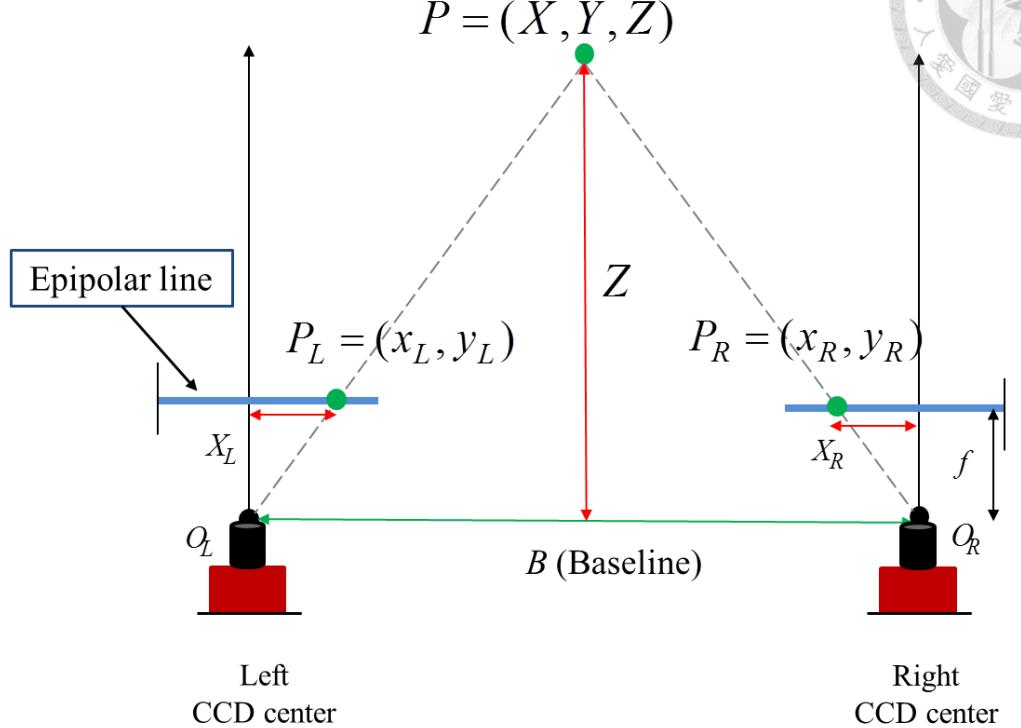


圖 3-9 極限幾何解釋與物體距離關係之示意圖

$$\frac{X}{Z} = \frac{x}{f}, \frac{Y}{Z} = \frac{y}{f} \quad (3.16)$$

$$\frac{X_L}{f} = \frac{X}{Z}, \frac{X_R}{f} = \frac{X - B}{Z} \quad (3.17)$$

$$Z = \frac{B \cdot f}{X_L - X_R} = \frac{B \cdot f}{d} \quad (3.18)$$

符號說明： (X, Y, Z) 是自行定義的目標物 P 的空間座標， X_L 與 X_R 分別為 P_L 、 P_R 投影點到所屬成像平面之中心距離， B 是兩攝影機鏡頭的水平距離， d 則定義為影像的像差值。

需注意的是以上三式的推導過程中是將雙眼視覺系統視為沒有任何旋轉與偏移的理想情況下（兩攝影機光軸平行對齊）計算而得。但是實際狀況中若因後端應用之要求，而加入旋轉的角度以及平移等參數，則需要將以上三式加上旋轉以及平移矩陣加以修正，如下式 (3.19)、(3.20) 所示，使修正過後的目標點空間

座標值為 (X', Y', Z') ， $R(\omega, \varphi, \kappa)$ 為旋轉矩陣。



$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R(\omega, \varphi, \kappa) \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix} \quad (3.19)$$

其中 $R(\omega, \varphi, \kappa)$ 之 ω 、 φ 、 κ 分別為 \hat{x} 、 \hat{y} 、 \hat{z} 方向的旋轉角度。 X_0 、 Y_0 、 Z_0 分別是世界座標的原點。由於是將地平面視為高度為 0 的位置，因此，將雙攝影機系統裝設於車上或其他載具時，考慮相對於原點的關係，才可以適當的將障礙物高度計算出來，下式 (3.20) 為從上式 (3.19) 中推導而出的公式。因為裝設時立體視覺系統具有一定的高度，倘若立體視覺系統維持水平方向，將無法完全偵測到前方 5 公尺內之障礙物。因此，我們將立體視覺系統以 x 方向下傾斜 ω 度，視 y 、 z 方向之旋轉角度為理想的 0 度， h 為系統架設的高度，可推導出式 (3.20)。

$$X' = X$$

$$\begin{aligned} Y' - h &= \frac{Y - Z \tan(\omega)}{\cos(\omega) + \tan(\omega) \cdot \sin(\omega)} \\ Z' &= \frac{Z + Y' \sin(\omega)}{\cos(\omega)} \end{aligned} \quad (3.20)$$

3.2.3 圖像匹配與像差影像

經過上一節的敘述得知，極線幾何理論推得的式 (3.16) (3.17) (3.18) 可求得像差影像 (Disparity image)，但是校正過後仍有一個未知數 d ，也就是像差值。像差值的求法需透過各像素點在左右影像中的對應位置，就可以計算出每個像素的 d 值，即可獲得像差影像。

本研究中，沿用賴 (2012) 所開發之計算像差影像之方法，其中延續採用 Birchfield *et al.* 所提出之點對點圖像比對方法算得代表深度資訊的像差影像 (Birchfield *et al.*, 1999)，以此為基礎，並利用 Intel 公司所提供之 OpenCV 開放程式碼之影像處理函式庫所提供之半區域式區塊比對法 (Semi-Global Block Matching, SGBM) 作為圖像比對的方法。實作時可用 OpenCV 以 StereoSGBM 類別宣告物件來設定各項係數並計算像差影像。

在 SGBM 的方法中，採用了價值函數 (3.21) 和動態規劃最佳圖像匹配的觀念，以左影像中一條水平像素序列做為基準和右影像相近的位置進行數條水平像素序列的比對。如此強健了左右影像彼此水平偏移誤差的容忍性，也更能找出可能匹配位置。

$$C(p) = N_{occ}k_{occ} - N_m k_r + \sum_{i=1}^{N_m} d(x_i, y_i) \quad (3.21)$$

其中 N_{occ} 是被遮蔽的像素量， N_m 是找到對應點的像素量，函式 $d(x, y)$ 是兩對應點的像素差， k_{occ} 是遮蔽的處罰係數， k_r 是正確對應的獎勵係數。對於每個像素點以此函式所計算出的最小值對應的位置作為其對應點，即可計算出像差影像。但實際上常會因為雜訊的干擾，造成錯誤的匹配點卻比正確的對應點有更小的價值函數值。因此實作時以 Hirschmuller (2008) 提出的另一種改良的價值函數，其中的概念為必須檢查初步計算出來的像差影像，考慮像素周圍的變化，變化太大則應該給予較大的懲罰值，因此定義出像差影像的能量函式 $E(D)$ ，而最終目的是讓 $E(D)$ 越小越好：

$$E(D) = \sum_p C(p, D_p) + \sum_{q \in N_p} P_1 T \left[|D_p - D_q| \right] + \sum_{q \in N_p} P_2 T \left[|D_p - D_q| > 1 \right] \quad (3.22)$$

其中 N_p 是像素點 p 周圍的像素群。第一項是所有像差值為 D_p 的代價函數總和。

p_1 、 p_2 是代價常數 (注意 p_1 要小於或等於 p_2)。 T 函式會在括弧內等式成立時為 1，否則為 0，因此第二項表示當 p 、 q 兩點像素差等於 1 時， $E(D)$ 要增加 p_1 ，第三項則表示像素差大於 1 時， $E(D)$ 要增加 p_2 。

此方法中，使用了 Sobel 運算子取得影像邊緣，然後設定區塊大小進行正規化，在此區塊中，最大與最小的像素灰階值必須介於該閾值內才會進行匹配。SGBM 圖像比對方法的主要優點為：(1) 可處理紋理特徵的大塊影像區域。(2) 對於兩相異點間的相似度計算提出新方法克服影像取樣所造成的問題。(3) 刪除非相似點以增加運算速度。像素比對示意圖如下圖 3-10 所示。

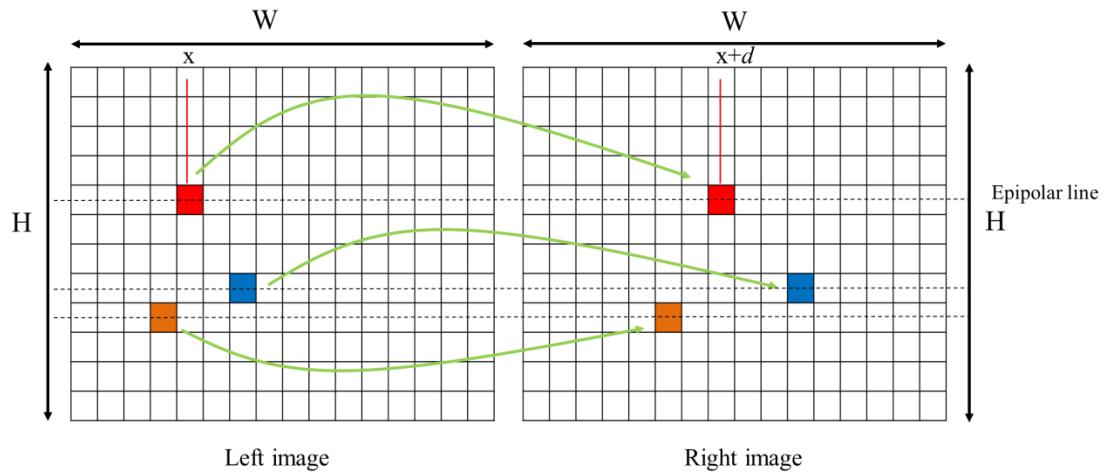


圖 3-10 像素比對示意圖

3.3 障礙物偵測方法

障礙物偵測其困難之處在於如何將立於地面上的障礙物與地面資訊分開，由於深度影像在地面資訊呈線性關係，若障礙物是垂直於地面上的一個平面，障礙物可視為具有相同深度的影像區塊，而道路則是線性增加或減少的關係。因此，可利用上視圖與團塊法 (Blob) 將障礙物由深度影像中計算且標示出來，並配合特定障礙物的特徵與幾何條件的限制將我們所關注的障礙物從影像中找出。



3.3.1 建立上視圖

上視圖建立的方法是參考 Pocol *et al.* (2008) 的研究改良而成。首先必須定義上視圖的幾何條件，根據攝影機的規格得知所採用的是 16 mm 的鏡頭，視角為 19.6 度，在 3.1 節曾計算出雙眼立體視覺之範圍，若定義障礙物距離攝影機最遠處為 30 公尺，則可拍攝到距離 30 公尺處水平方向約為 10.8 公尺的範圍。當上視圖的有效範圍決定後，另一重要參數為上視圖的解析度，即網格數目。假設上視圖的橫軸方向呈線性關係，縱軸為對數關係，如下式

$$Col_{TV} = \frac{l_{TV}}{C} \quad (3.23)$$

$$Row_{TV} = \log_{(1+k)} \left(\frac{L_{TV}}{L_0} \right) \quad (3.24)$$

上式中 C 為所設定之橫軸方向寬度， Col_{TV} 為計算出來的網格數； L_0 為縱軸方向的最小距離，預設為 2.5 公尺， k 為縱軸方向網格的延伸比率， Row_{TV} 為計算出來的網格數，示意圖如圖 3-11。因此可藉由此關係求出上視圖每個網格所代表的實際距離與寬度。當影像中每個像素的三維座標求出後，即可利用所在的橫軸方向與深度距離資訊將其投影至上視圖中。在本研究所採用的影像解析度為 640×480 ，若以此大小之影像進行團塊處理，除了不易取出障礙物之外，速度也較慢。然而投影至上視圖之後再進行團塊處理，則可以將處理範圍縮小至 100×125 ，因此可大幅提升團塊處理的速度。有關團塊處理法將於下節做說明。

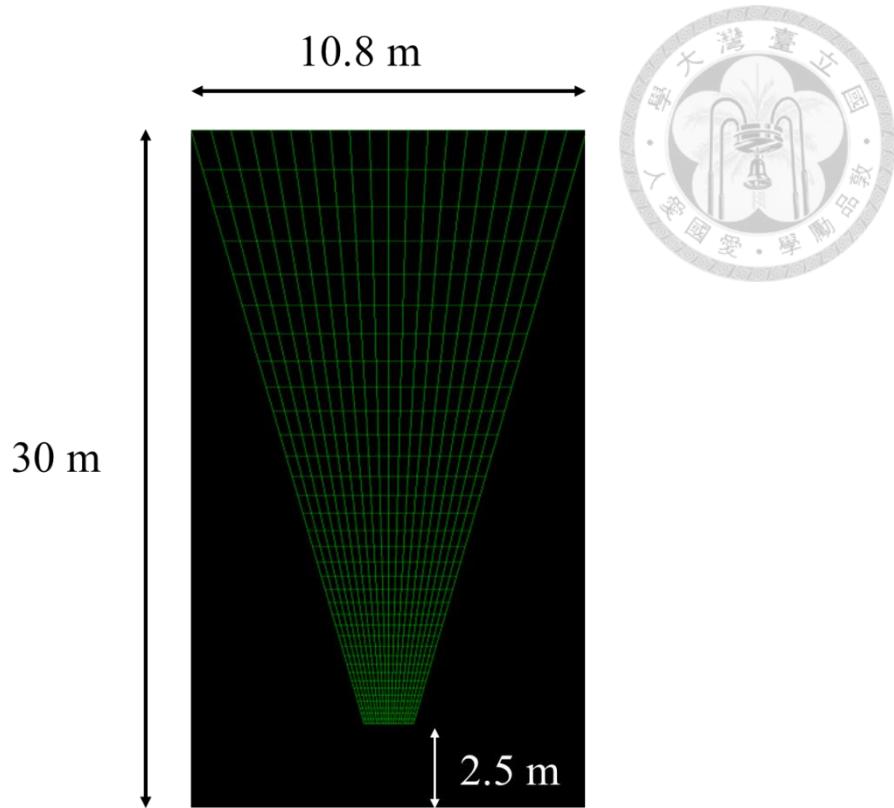


圖 3-11 立體視覺系統之前方環境上視圖

3.3.2 團塊法

團塊法 (BlobMethod) 是一種常用於電腦視覺的區域偵測法，目的大多是為了偵測點或局部相較於周圍差異較大的明暗或色彩區域以達到偵測的目的。然而團塊法不同於邊緣偵測 (Edge Detection) 或角點偵測 (Corner Detection) 是利用計算梯度變化來處理，而是運用分層的概念 (Scaling) 將色彩或明亮程度分成數個層級，最後區域中若符合同一個層級之中設定之像素累計數量便將其視為團塊 (Blob)。

根據上述的團塊法概念，當像差影像計算得到後，我們依據具有深度資訊的像差影像進行影像分層，共分為 N 個層級 ($N=255$)，以 N 張影像來表示，這些分層所代表的意義如圖 3-12 所示。在像差影像中，像差值越大其距離攝影機越近，像差值越小則越遠。產生這些分層後，我們將每個像素所計算出的深度值投影至上視圖，再設定累計像素的量當作閾值，便可將每一層中之團塊 (Blob) 取出。然而，實作時團塊處理法會將許多小雜點 (Noise) 框選出來，為了解決此問

題，我們定義障礙物需具特定的幾何條件來將雜點濾除，這些條件包含所框選出來的團塊影像必須具有高度值，並且團塊的上下邊界應具有高度差才能被視為障礙物，這些限制將在下節說明，團塊處理法的結果如圖 3-13。

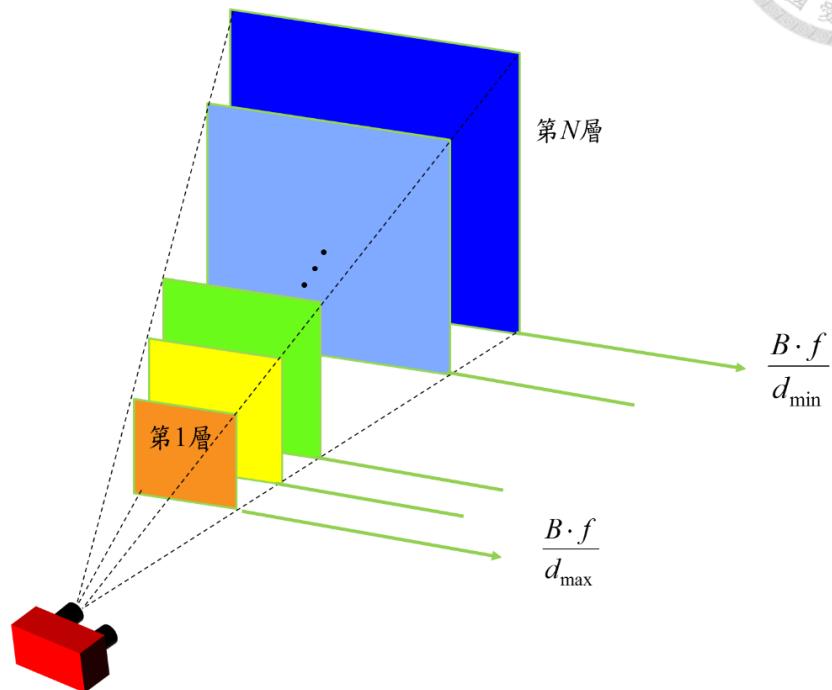


圖 3-12 距離與分層影像的關係示意圖

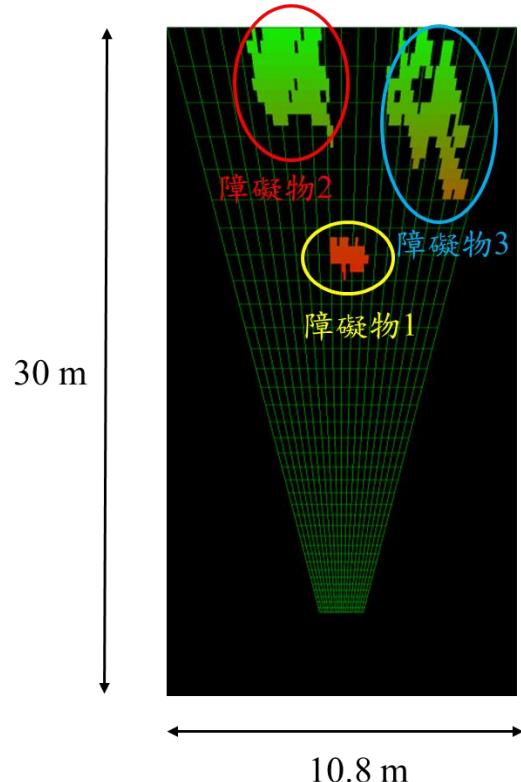


圖 3-13 團塊處理後之上視圖



3.3.3 設立障礙物條件

上節提到團塊法的執行已將影像中符合條件之物體做了初步的篩選，然而若不加以限制則會框選出不必要的雜點，因應實際之需求可再進一步定義障礙物的幾何條件，將感興趣的障礙物作進一步的濾出，以下將列舉我們所定義的障礙物條件：

- A. 投影至上視圖的障礙物距離須介於 30 公尺範圍內。
- B. 投影至上視圖的像素其高度必須介於設定範圍內（預設 20）。
- C. 投影至上視圖之每個網格的像素量必須大於設定值（預設 20）。
- D. 障礙物之影像大小必須大於設定值。
- E. 障礙物之實際寬度必須大於設定值。

設計以上之條件主要在於可濾除不必要之雜點，以及針對我們所設定的障礙物進行長度、寬度的篩選，並藉由上視圖與影像之間的對應關係將障礙物框選出來。由於攝影機解析度僅 640×480 ，為了避免圖像比對的失誤造成像差值的誤算，因此若需偵測較遠的物體，則可考慮更高解析度以及更高倍率的鏡頭。然而增加解析度雖可擴大偵測範圍，但會使運算量增加而降低處理速度。

3.4 障礙物追蹤方法

影像中物體追蹤技術主要分成兩個步驟，首先必須偵測出欲追蹤之障礙物，接著藉由障礙物特徵找出前後影像中相互對應的物體做匹配達成追蹤目的，因此找出具代表性之障礙物特徵為之重要。

本研究基於雙眼立體視覺系統擁有多色及距離資訊，巴氏距離是用來計算兩個統計值方圖之間的相似程度，因此分別計算 HSV 色彩空間模型中顏色特徵與距離特徵的巴氏距離並結合作為匹配的準則。然而，根據實驗經驗巴氏距離會因

障礙物相對攝影機之移動速度而造成偵測上的跳動，為了克服此問題則需先估算障礙物的速度，並以統計的方式改良發展出適應性巴氏距離來更新匹配的閾值。

下圖 3-14 說明了本文追蹤障礙物時的資料流程圖，並於後續說明其演算過程。

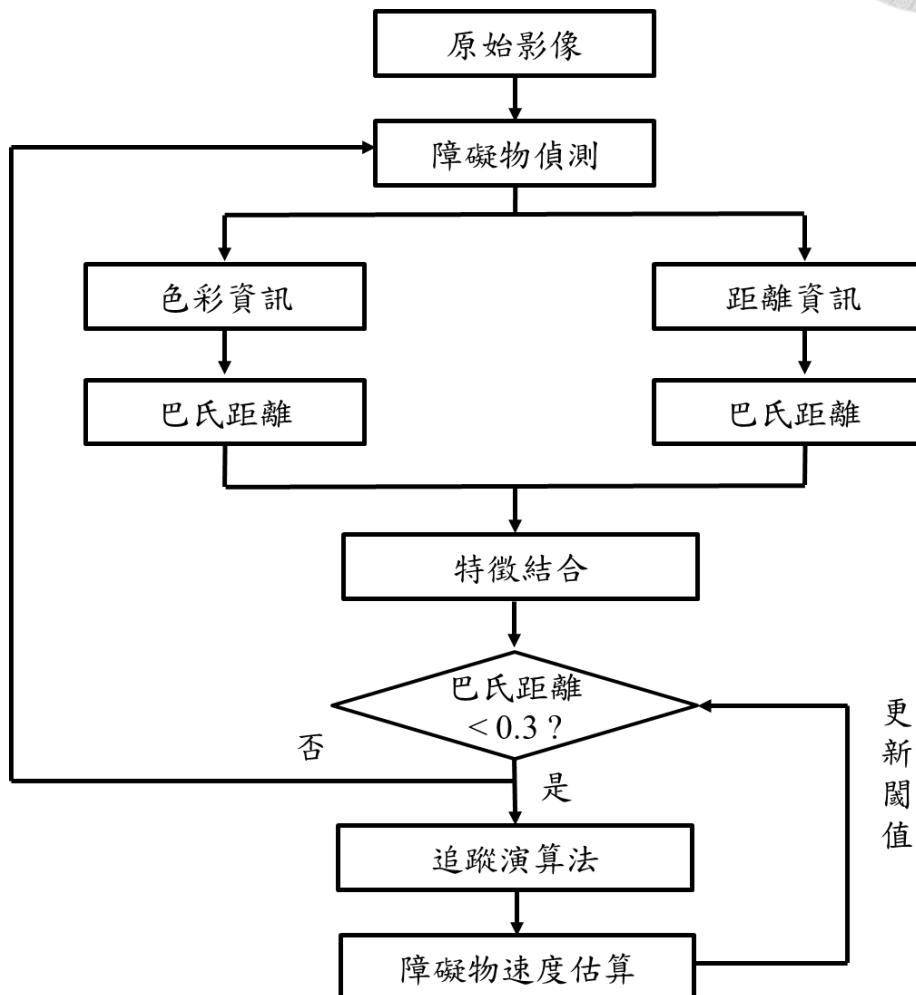


圖 3-14 追蹤方法資訊流程圖

3.4.1 特徵擷取與匹配

(1) 影像色彩資訊

在將目標物從影像序列中分離後，我們可獲得障礙物之彩色資訊，此為一有效之特徵。Sural *et al.* (2002) 曾使用 HSV 色彩模型之分量並考慮 N 個強度之直方圖統計來比較，進而判斷是否為相同物體。巴氏距離 (Bhattacharyya Distance)

可用於測量兩密度分布函數之相似度並且應用於分類上，因此本文利用巴氏距離計算影像序列上與各候選障礙物的兩兩統計直方圖相似程度以進行匹配。巴氏距離經正規化後可表示為

$$d(H_1, H_2)_{color} = \sqrt{1 - \frac{1}{\sqrt{H_1 H_2 N^2}} \sum_I \sqrt{H_1(I) H_2(I)}} \quad (3.25)$$

H_1 及 H_2 分別為 N 個灰階強度之直方統計圖，以本研究為例 $N = 255$ 。然而，起初進行障礙物偵測時，是利用簡單的幾何圖形（矩形）來表現障礙物，做法是將影像上所框選出各個團塊的最左上角和最右下角的點座標計算矩形的長和寬，然後再框出。但是這種利用簡單的矩形所框選出的障礙物會參雜不必要的雜訊，所以在進行巴氏距離的計算之前，可先開啟一張原先框選出的障礙物影像大小設定像素的灰階大小為 0，並將團塊中的各像素依照原先影像中的位置和灰階值進行投影，如圖 3-15。

為了使色彩直方圖只單純計算障礙物的部分，將已偵測到不含背景的影像且去除灰階值為 0 之色彩直方圖進行巴氏距離的計算即可，故 I 值範圍為 $[1, N]$ ，經過計算後的兩張影像其色彩資訊越接近則巴氏距離值會越接近 0。



圖 3-15 障礙物 (a) 含背景，(b) 不含背景

(2) 影像深度資訊

立體視覺可進行像差影像的計算並獲得影像上各像素之三維座標。根據色彩

特徵的巴氏距離計算，同理，我們可將偵測到之障礙物影像以各個像素的三維座標取出深度分量進行直方圖統計，並計算深度資訊特徵的巴氏距離。

$$d(H_3, H_4)_{distance} = \sqrt{1 - \frac{1}{\sqrt{H_3 H_4 N^2}} \sum_D \sqrt{H_3(D) H_4(D)}}$$
(3.26)

H_3 及 H_4 分別為可偵測的最遠距離經過 N 次取樣累計之直方統計圖。如式 (3.27)， N 為取樣數，也就是直方統計圖橫軸的個數， $DisZ$ 代表設定的障礙物偵測最遠距離， L 代表組距。以本研究為例 $N=255$ ，偵測距離設定為 30 公尺，則會以組距 30/255 公尺的方式進行分組的直方圖統計。

$$L = DisZ / N \quad (3.27)$$

(3) 特徵結合

根據本節 (1)、(2) 巴氏距離的計算後，已經得到兩個有效的特徵並可單獨拿來進行匹配達到追蹤的目的。然而，特徵結合在近年來物體追蹤和辨識領域已廣泛被探討與使用，根據 Yang *et al.* (2003) 的歸納主要可分為平行 (Parallel Feature Fusion) 跟串聯 (Serial Feature Fusion) 兩種。若定義兩特徵像量為 α 、 β 分別為 A、B 個維度，則平行結合會以線性方式 $y = \alpha + i\beta$ 進行特徵的組合， i 代表權重大小又稱結合係數，結合後的維度為 $\max\{\dim A, \dim B\}$ 且此特徵獨立的；串聯結合則以下式表示：

$$\gamma = \begin{pmatrix} \alpha \\ \theta\beta \end{pmatrix} = \begin{pmatrix} d_{color} \\ \theta \cdot d_{distance} \end{pmatrix} \quad (3.28)$$

串聯後的維度為 $(A + B)$ ，並以歐式距離進行結合 (Yang *et al.*, 2003)， θ 為結合係數，即兩特徵之權重關係，通常會根據實驗結果找出適當的值 (本研究之 $\theta=1$)。經過實驗觀察顯示利用串聯結合特徵的方式所得到的效果較佳，因此結合上兩式 (3.25)、(3.26) 後可得匹配的特徵值表示為

$$d_{match} = \sqrt{d_{color}^2 + d_{distance}^2} \quad (3.29)$$

(4) 匹配方法

本研究所採用的障礙物比對策略為前後兩張影像之障礙物進行交叉比對法 (Cross Matching)，當連續影像在 $t=0$ 和 $t=1$ 兩個時間的障礙物分別為 $K_0 = (k_{0,0}, k_{0,1} \dots k_{0,N-1})$ 和 $K_1 = (k_{1,0}, k_{1,1} \dots k_{1,M-1})$ 之集合表示， N 、 M 代表障礙物數量，以圖 3-16 為例 $N=2$ ， $M=3$ 進行特徵值計算後得到的集合 $d_{0 \rightarrow 1} = (d_{0,0}, d_{0,1} \dots d_{0,N-1})$ 表示第 1 個障礙物在 $t=1$ 對應到 $t=0$ 時匹配候選之集合以此類推，在這個結合中的最小值稱為匹配值，在 $t=0$ 對應到 $t=1$ 時的特徵值集合則表示為 $d_{1 \rightarrow 0} = (d_{1,0}, d_{1,1} \dots d_{1,M-1})$ ，在計算每個障礙物的匹配值之後可以用式 (3.30) 表示。最後只要在式 (3.30) 中找尋雙向的匹配值相同並且確定匹配值小於所設定的閾值即完成了交叉比對。交叉比對的最大優點在於當前後影像中的障礙物數量不同時可以避免該時間才出現的障礙物匹配錯誤的情形。

$$\begin{aligned}
 Index_{1 \rightarrow 0} &= \min \{d_{i,0}, d_{i,1} \dots d_{i,N-1}\} \\
 Index_{0 \rightarrow 1} &= \min \{d_{j,0}, d_{j,1} \dots d_{j,M-1}\} \\
 \begin{cases} \text{if } min \leq threshold_{Bha} \\ i = [0, M-1] \end{cases} \\
 \begin{cases} \text{if } min \leq threshold_{Bha} \\ j = [0, N-1] \end{cases} \tag{3.30}
 \end{aligned}$$

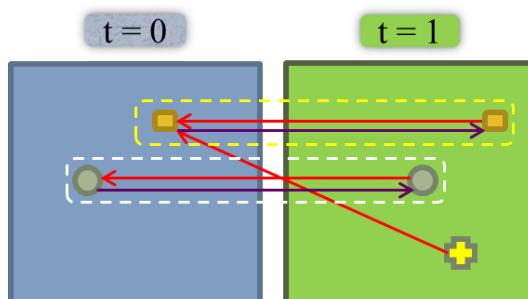


圖 3-16 障礙物匹配示意圖

為了能找出適合的閾值做為障礙物匹配準則的初始值，設計一個實驗進行統計，實驗過程中以三個障礙物互相比對取得匹配值，並記錄所有候選障礙物匹配值之散布圖，如下圖 3-17，每次候選的匹配值之中會有一個最小值，以紅色圈圈表示，其它則為藍色。因此資料群可區分成藍色非最小值的匹配點，和紅色最小值匹配點兩類，結果可看出最小值的匹配點均落在單位圓半徑 0.3 之內，因此決定初始設定的閾值為 0.3。

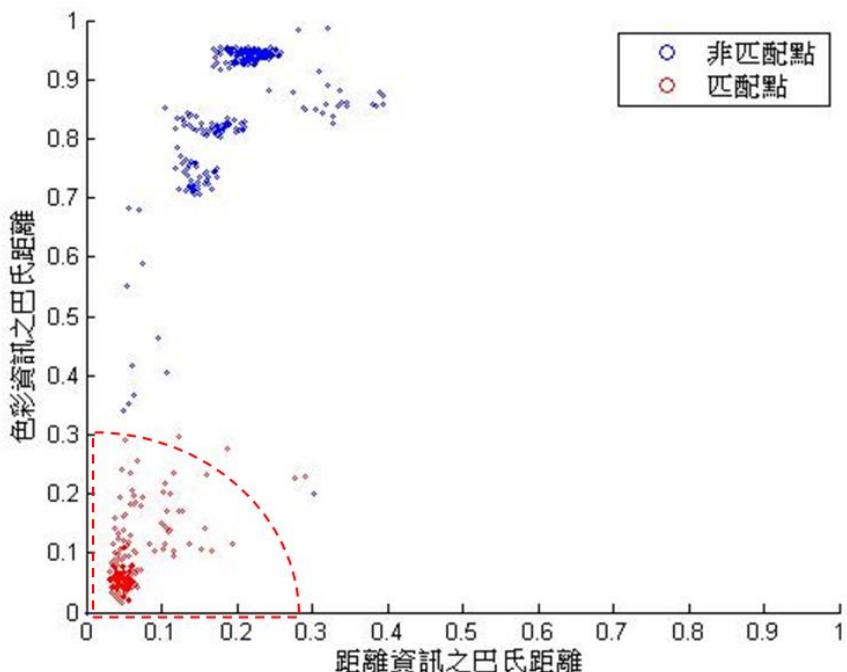


圖 3-17 障礙物匹配值散布圖

3.4.2 適應性追蹤演算法

我們定義追蹤成功建立在計算巴氏距離需在設定的閾值 0.3 以下才視為前後影像為相同障礙物，在上一節曾利用圖 3-17 說明設定閾值的原因，然而根據實驗進行追蹤結果發現，速度對系統進行障礙物追蹤有一定的影響，不管在行人或是自行車的偵測，速度越快都會降低追蹤的成功率，因此設計一實驗將不同速度與其對應之追蹤成功率與巴氏距離做統計，如下表 3-2、3-3，假設實驗者以直線等速度移動，則此時的平均速度以碼表時間 t_{timer} 跟實際進行立體視覺與障礙物偵

測計算出的障礙物平均深度 $DisZ_{system}$ ，再經過單位轉換所得。



$$v = \frac{DisZ_{system}}{t_{timer}} \text{ (km/hr)} \quad (3.31)$$

追蹤失敗代表所偵測到之障礙物相似度太低。探究其原因發現多半是如下圖 3-18 的情況，因計算像差影像的錯誤導致團塊法沒有框選出整個行人，最後計算巴氏距離會因為大於 0.3 而追蹤失敗，但我們知道實際上前後兩張影像應屬同一個障礙物，這種情形在障礙物行進速度較快的形況下會相對增加。

表 3-2 實驗相關參數

攝影機高度	110 公分
偵測範圍	5~15 公尺
障礙物高度	行人高度約 167 公分 自行車高度約為 168 公分
系統速度	約 10~15 fps

表 3-3 速度不同的障礙物追蹤成功率

		平均速度 (km/hr)	追蹤成功率 (%)	巴氏距離 平均值	巴氏距離 標準差	樣本數
行人	跑	16.37	87.09	0.204	0.091	41
	快走	5.93	99.02	0.131	0.055	102
	慢走	4.72	99.17	0.142	0.056	121
自行車	快騎	14.07	83.33	0.198	0.092	42
	慢騎	8.20	94.44	0.141	0.077	90



圖 3-18 追蹤失敗示意圖

根據上表 2 可看出速度對系統進行障礙物追蹤有一定的影響，不管在行人或是自行車上速度越快都會使得追蹤成功率下降。巴氏距離是一種計算兩張影像相似度的一種方法，其範圍介於 0 到 1 之間，值越小代表相似度越高，根據實驗統計結果我們觀察到障礙物移動速度越快巴氏距離的平均值與標準差相較於其他兩者都來的高，這代表立體視覺系統所偵測到障礙物影像的跳動程度較大，換句話說，速度增加會降低系統的穩定程度。這種狀況並非偶然，我們觀察到自行車也有相同的趨勢。

前面提到利用巴氏距離作為追蹤的準則，為了要克服障礙物行進速度造成系統穩定程度下降而影響追蹤成功率。根據不同速度之巴氏距離平均值及標準差，將其做為系統偵測到障礙物跳動的量化指標將有利於設定不同狀態下障礙物追蹤時設定的閾值。在實驗統計數據具有近似於常態分佈的機率分布之條件下，則約有 95% 數值分布在距離平均值有 2 個標準差之內的範圍，由於僅需考慮相似程度的底限，故只需加上 2 個標準差作為閾值即可。因此只要將速度估算之實際

相對速度 $v_{relative}$ 與建議閾值 $threshold_{Bha}$ 之間的關係建立模型即可，建立的模型如圖 3-19，並得到式 (3.32)。

$$threshold_{Bha} = 0.0165 \times v_{relative} + 0.1601 \quad (3.32)$$

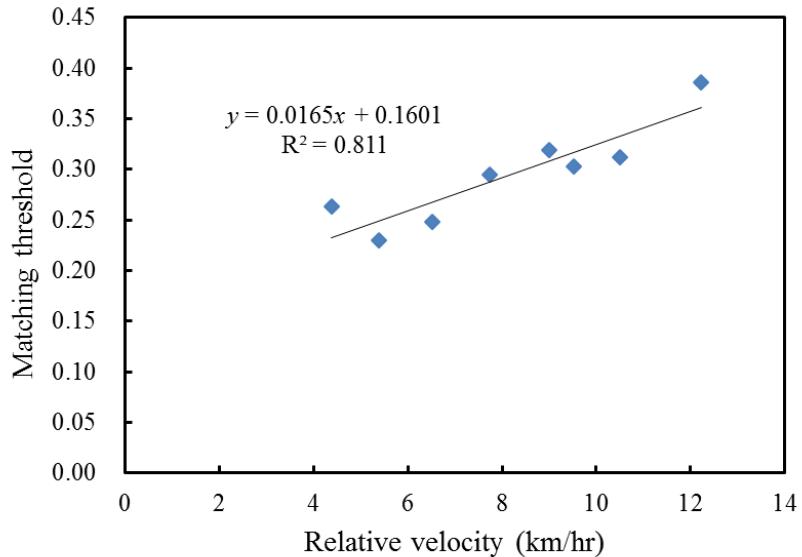


圖 3-19 速度與匹配閾值之關係

3.4.3 速度估算方法

即使得到速度與巴氏距離的模型，若不能在系統中估測障礙物的實際相對速度則無法發揮作用。根據立體視覺所得的三圍資訊可得到障礙物的真實座標，再加上立體視覺系統的影像框架速度 (fps) 為 15，可分別得到移動距離 $\Delta DisZ_{system}$ 及時間 $\Delta t = 1/15$ ，以此計算出障礙物速度。經過統計驗證後發現，這系統初估的速度與式 (3.31) 兩者之間有著線性關係。探究其原因發現，系統於離線進行分析時在硬碟讀取影片時以及錄製影片時寫入硬碟的動作皆會使原本預設的框架速度小於 15 fps。因此，系統估算障礙物實際速度需進行校正，如下式 (3.33)

$$v_{ob} = 0.3323 \times \frac{\Delta DisZ_{system}}{\Delta t} - 0.4254 \quad (3.33)$$



3.5 避障模式

本節主要敘述立體視覺架設於車輛時欲進行避障的方法，根據本章前面幾節的方法描述，已經能取得前方環境的大部分資訊包含偵測與追蹤障礙物，並估算出障礙物與攝影機的相對速度，藉以提供駕駛人了解車前障礙物之動態資訊。然而這些都是歷史資訊，若是能進一步知道障礙物未來的行進方向或甚至預測下一個時間點的位置，則更能使駕駛人做出正確的行車判斷，因此本節將依序提出演算法進行預測與路徑規劃，並建立一個能進行即時警示與導航的避障模式。



圖 3-20 避障模式示意圖

3.5.1 建立權重地圖

避障模式的最終目的是進行路徑規劃，在這之前我們必須建立權重地圖，最初透過立體視覺計算的上視圖以 30 公尺為偵測範圍可將其表示成 100×125 的二維矩陣，其中對應影像中經過計算所得到座標之像素個數，因此當作權重地圖的初始值是合理的。透過 5×5 的遮罩運算進行高斯平滑化處理後，可以擴大障礙物的影響範圍，其意義為障礙物附近在一定的範圍內也是危險的（高斯遮罩係

數如下圖 3-22)，這種作法與第二章所述的勢力場有相同的概念。

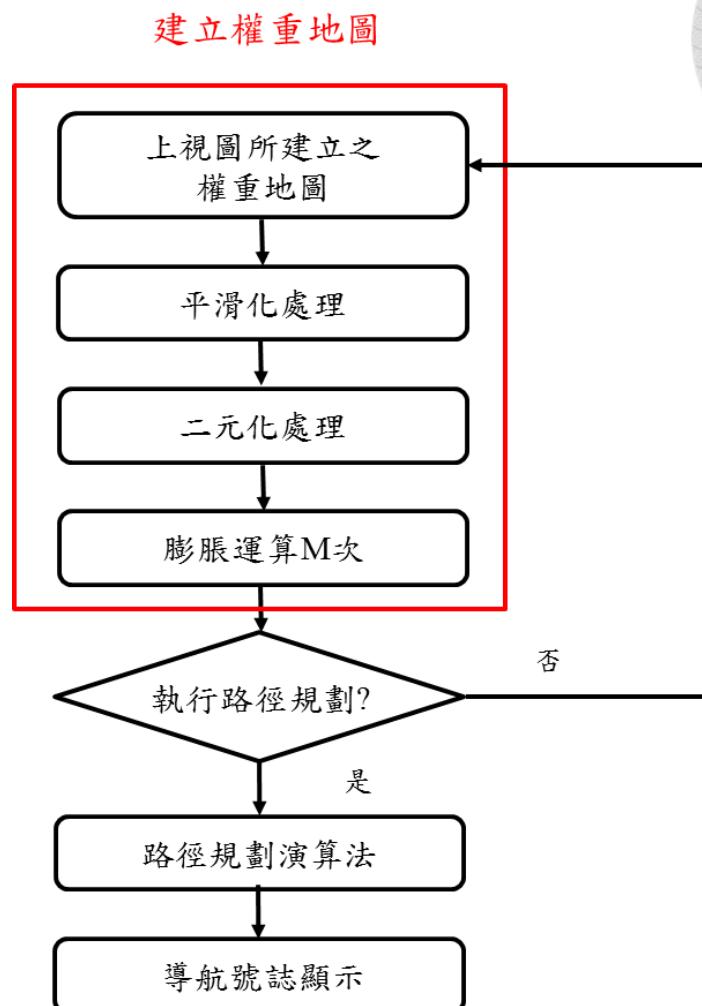


圖 3-21 避障模式流程圖

A 5x5 matrix representing Gaussian smoothing coefficients. The matrix is symmetric and centered around a value of 16. The values are as follows:

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

1
273

圖 3-22 高斯平滑化係數

進行平滑化之後，權重地圖上的值為 0 到 1000 不等，由於路徑規劃演算法需耗費較多的時間迭代，若是將權重地圖簡化將可減低運算時間，因此我們進行二值化，其中定義二維矩陣中的值大於或等於 100 即為障礙物，小於 100 則可視為安全區。此外，由於系統裝設在車輛上，必須將車身大小也考慮進去，所以利用形態學的膨脹 (Dilation) 對權重地圖進行運算。定義符號 M_{smooth} 和 $M_{Dilation}$ 分別為平滑化後與膨脹運算後的二維格子地圖。

$$M_{Dilation} = \begin{cases} 1 & \text{if } M_{Smooth}(i, j) \geq 100 \\ 0 & \text{otherwise} \end{cases} \quad (3.32)$$

膨脹為影像處理中用來將二元化之影像進行擴張的運算方法，主要利用 OR 的邏輯運算，當考慮到空間中的兩個集合 A 和 B 若 A 被 B 膨脹時可表示如下

$$A \oplus B = \bigcup_z \{[(\bar{B})_z \cap A] \subseteq A\} \quad (3.33)$$

其中 B 是一個結構元素 (Structuring Element)，而 A 是被膨脹的集合，只要將 B 這個結構元素不斷的位移 Z 單位， B 集合與 A 集合有重疊至少一個元素，這樣所形成的新集合即為 A 被 B 所膨脹的集合。膨脹運算在實作時結構元素的決定很重要，而 100×125 的權重地圖根據 3.1 節的立體視覺理論，計算得到其代表偵測範圍為深度 30 公尺，水平方向為 10.8 公尺的前方資訊，為了方便計算假設上視圖之橫向與縱向皆為線性關係，每個單位所代表的實際距離便可計算而得，以一台車身長 4540 mm、寬 1760 mm 的小型轎車為例，將長寬各乘以 1.2 倍作為安全係數，可算出 19×23 大小的結構元素，但實際上可以利用 3×3 的結構元素重覆運算 11 次達到相近的效果 (如下圖 3-23)，所以只要建立車身與膨脹運算次數的關係便能讓使用者依據自身的車子大小更彈性使用。



$$M = \arg \max \{m_{width}, m_{length} \mid m \in N\}$$

$$m_{width} = \left(\left\lceil C_{SF} \times \frac{w}{a} \right\rceil - 1 \right) / 2$$

$$m_{length} = \left(\left\lceil C_{SF} \times \frac{l}{a} \right\rceil - 1 \right) / 2 \quad (3.34)$$

w：車身寬度

l：車身長度

a：網格地圖所代表之實際寬度，單位：mm

b：網格地圖所代表之實際長度，單位：mm

C_{SF} ：安全係數

M：膨脹運算次數

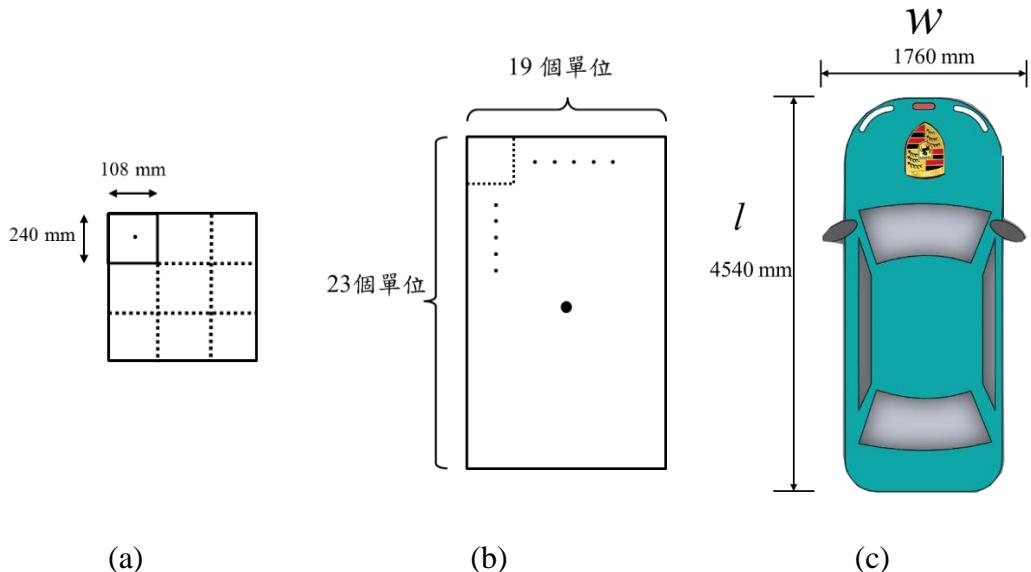


圖 3-23 膨脹運算使用的結構元素示意圖 (a) 3×3 結構元素之實際大小，
(b) 19×23 結構元素，(c) 車身大小示意圖

3.5.2 卡爾曼濾波器

卡爾曼濾波器 (Kalman, 1960) 為一種自迴歸濾波器，主要有兩類應用，最早於控制系統工程中用於濾波 (Filtering 或稱 Smoothing)，另一類是可對該系統做出預測。特別是後者被廣泛應用在需要預測與修正的場合如機器人導航，控制，

追蹤等，可稱做最佳化自迴歸處理演算法。因此本研究將以此方法進行障礙物位置預測。

首先，假設系統為線性如下式，其中狀態矩陣 $[x, y, dx, dy]$ 表示座標位置以及位移量

$$\begin{bmatrix} x' \\ y' \\ dx' \\ dy' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ dx \\ dy \end{bmatrix} \quad (3.32)$$

在進行卡曼濾波器演算法運算之前必須先建立運動模型，而障礙物的位置與速度以線性狀態空間描述，即

$$x(k) = \begin{bmatrix} x \\ v \end{bmatrix} \quad (3.33)$$

其中 x 為位置， v 為位置對時間的導數，假設障礙物的加速度符合均值為零。在此我們將引入一個離散控制過程的系統，該系統可用線性隨機微分方程來描述，並且根據牛頓運動定律可得線性系統狀態方程式為

$$x(k) = Ax(k-1) + BU(k) + W(k) \quad (3.34)$$

其中

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \Delta t^2 \\ 2 \end{bmatrix} \quad (3.35)$$

再加上系統的測量值為

$$z(k) = Hx(k) + V(k) \quad H = [1 \ 0] \quad (3.36)$$

上兩式子中， $x(k)$ 是 k 時刻的系統狀態， $U(k)$ 是 k 時刻對系統的控制量。 A 和 B

是系統參數。 $z(k)$ 是 k 時刻的測量值， H 是測量系統的參數。 $W(k)$ 和 $V(k)$ 分別表示過程和測量的噪聲 (Noise)，分別被假設成高斯白噪聲 (White Gaussian Noise)，其共變異數 (Covariance) 分別是 Q ， R 並假設不隨系統狀態而變化。接著將進行滿足線性隨機微分系統，過程和測量都是高斯白噪聲的兩個條件之下的卡爾曼濾波器運算。根據上述，我們利用系統的過程模型來預測下一狀態的系統，即

$$x(k|k-1) = Ax(k-1|k-1) + BU(k) \quad (3.37)$$

上式中 $x(k|k-1)$ 是利用上一狀態預測的結果， $x(k-1|k-1)$ 是上一狀態最佳的結果， $U(k)$ 為現在狀態的控制量，可為 0。此時系統結果已更新，但是對應於 $x(k|k-1)$ 的共變異數尚未更新，以 P 表示共變異數，即

$$P(k|k-1) = AP(k-1|k-1)A' + Q \quad (3.38)$$

$P(k|k-1)$ 是 $x(k|k-1)$ 對應的共變異數， $P(k-1|k-1)$ 是 $x(k-1|k-1)$ 對應的 covariance， A' 表示 A 的轉置矩陣， Q 是系統過程的 covariance。以上兩式就是卡爾曼濾波器五個主要公式當中的前兩個，也就是對系統的預測 (Prediction)。一旦有了現在狀態的預測結果，並取得現在狀態的測量值做結合便可得到狀態 (k) 的最佳化估算值 $x(k|k)$

$$x(k|k) = x(k|k-1) + Kg(k)(z(k) - Hx(k|k-1)) \quad (3.39)$$

其中 Kg 為卡爾曼增益值 (Kalman Gain)

$$Kg(k) = P(k|k-1)H' / (HP(k|k-1)H' + R) \quad (3.40)$$

至此以計算 k 狀態下之最佳估算值 $x(k|k)$ ，為了使卡爾曼濾波器不斷運行直到系統過程結束，將更新 k 狀態下的共變異數，即



$$P(k|k) = (I - Kg(k)H)P(k|k-1) \quad (3.41)$$

其中 I 為單位矩陣，當系統進入下一個時間時， $P(k|k)$ 就是卡爾曼第二公式的， $P(k-1|k-1)$ 以上三式是對系統的更正 (Correction)，如此一來就可以自迴歸的運算至系統結束。

表 3-4 卡爾曼濾波器演算法

Step 1: Setting initial guess and model

Set_model as 1D array

$$Pos(t) = (x_{system}(t), y_{system}(t))$$

$$Pos_{initial}(t) = Pos_{tmp}(t) + v_{relative}(t) \times t$$

Step 2: Kalman prediction

Kalman_Predict();

$$Pos_{tmp}(t) = Pos(t)$$

Step 3: Update measurement

$$measure(t) = Pos(t)$$

Step 4: Correct

$$Kalman_correct(measure(t))$$

實作過程如表 3-4，只要宣告 OpenCV 內建函式庫之 KalmanFilter 類別即可進行運算， $Pos(t)$ 代表該障礙物在時間 t 時的座標位置，需特別注意初始值 $Pos_{initial}$ 是經過前一節速度估算而得，也是本研究於後一章節所探討的問題。最後呼叫 *Kalman_Predict* 與 *Kalman_correct* 函式預測座標位置和更新量測值與估測值之間的關係，以此反覆運作。實際上，位置預測後的結果便會依據圖 3-23 (a) 上每個單位的實際大小進行換算，找出在權重地圖上相對應的位置並給予 3

$\times 3$ 大小的適當權重，並視為障礙物。

3.5.3 路徑規劃



A^* 演算法是在 1968 年由 Hart *et al.* 提出，以最佳解優先搜尋方式 (Best First Search) 搜尋地圖節點之路徑最佳方法。在此搜尋演算法中，只要起點與終點之間有路徑存在，則保證搜尋出最少花費成本之路徑，並且導入了起點與終點之間的啟發式 (Heuristic) 算法減少運算量，相較於傳統的 Dijkstra's algorithm 有搜尋範圍較小以及時間複雜度較低的優點。 A^* 演算法較常用於機器人導航之用途，其概念是利用預先建立之網格地圖 (Grid Map) 上的節點，產生從起點到終點之路徑， A^* 路徑搜尋演算法中，我們透過從起始點開始檢查相鄰方格的方式，向外擴展到找到終點為止。在這中間過程中，運用了 *OPEN* 列表跟 *CLOSE* 列表，前者代表待檢查的列表，目前被搜尋到的節點會與此列表中的每個節點計算出一個成本，並找出最小成本的節點，將該點更新為下次搜尋的節點，然而前一個節點便會放到 *CLOSE* 列表，如此反覆搜尋。*CLOSE* 列表裡則儲存所有不需要再次檢查的節點。此外在更新節點時必須將前一個我們稱為 *Parent* 的節點記錄下來，這將有助於之後畫出路徑時的重要資訊。

1	2	3
4		5
6	7	8

圖 3-24 節點八相鄰示意圖

我們將以下圖的例子來說明 A^* 演算法的計算過程，圖中的每一個格子均代表一個節點，紅色代表預先設定的終點，綠色代表起點，中間藍色區塊假設是一面不可穿越的牆或是障礙物。

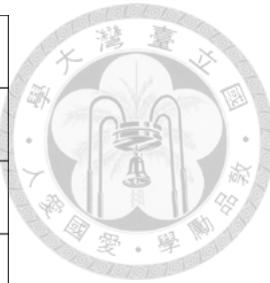
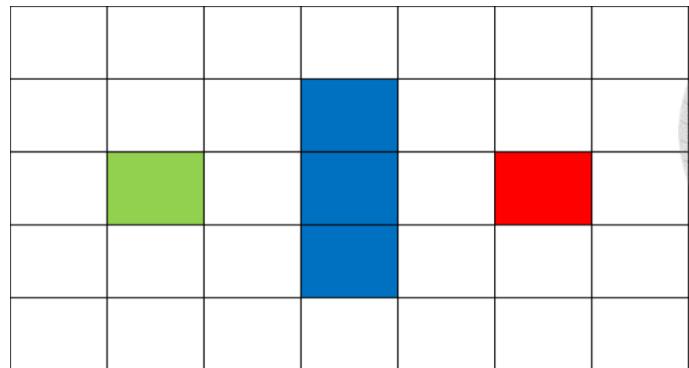


圖 3-25 進行路徑規劃的網格狀地圖

接著我們將重複前面敘述的步驟，將 *OPEN* 列表中的候選節點進行成本計算，計算公式如下

$$F(n) = G(n) + H(n) \quad (3.42)$$

$F(n)$ 代表目前節點到終點所花費的成本， $G(n)$ 代表從起始點到目前節點所花費的成本，計算 $G(n)$ 的成本我們不需要繞過中間間隔的障礙物，以本範例我們可以看出綠色格子右邊方格距離紅色方格有三格的距離，則定義 $G(n)$ 為 30，其餘的以此類推， $H(n)$ 稱作啟發式評估函數，用來估測移動成本的一個方法，在這裡令水平或垂直移動的成本為 10，對角線為根號 2 倍約 14.14，為了方便計算假設為 14，計算出的值如下圖，因此下一回合將由這回合選出最小成本的方格進行處理。

		F H G				
74 14 60	60 10 50	54 14 40				
60 10 50		40 10 30				
54 14 60	60 10 50	54 14 40				

圖 3-26 該節點經過其八相鄰到終點之成本

下一回合便以綠色格子右邊進行成本計算。八相鄰之中有三格是障礙物，一開始

就已經被放入 *CLOSE* 的列表中，而綠色格子因為計算過成本了也同樣被放入 *CLOSE* 裡，而剩下的四格則是在上一回合就被放入 *OPEN* 列表中。檢查 *H* 值來判定是否有更好的路徑，發現從目前的格子（對照圖 3-24 的 5）移動到下方（對照圖 3-23 的 8）則 *H* 值將從原本的 10 累積成 20，20 大於 14 成本增加，代表與其向右走再向下走不如直接走對角線。

當對 *OPEN* 列表中的 4 個格子重複了上述的計算後，發現沒有一條路徑可以透過使用目前的格子得到改善，所以不做任何改變，此時 *OPEN* 列表中有 7 個格子（對照圖 3-24 的 1234678），而用藍色框框代表進入 *CLOSE* 列表中的格子。此時發現最小成本有兩個格子是 54，但這時候會偏向選擇新加入 *OPEN* 列表中者，這並不影響最後計算出的成本大小，但可能會計算出相同成本但是不同的路徑。

下一回合在不能穿越牆角的情形之下，只有兩個方格可以新增至 *OPEN* 的列表中並把當前（對照圖 3-24 的 8）的格子指定為其 *Parent*，其計算成本的方式一樣，結果如下圖：

		<i>F</i> <i>H</i> <i>G</i>				
74 14 60	60 10 50	54 14 40				
60 10 50		40 10 30			Red	
54 14 60	60 10 50	54 14 40				
	88 28 60	74 24 50				

圖 3-27 路徑規劃過程圖

如此重複這樣的過程直到終點格子被放入 *CLOSE* 列表中，如下圖所示



108 28 80	94 24 70	80 20 60	74 24 50				
94 24 70	74 14 60	60 10 50	54 14 40				
80 20 60	60 10 50		40 10 30		82 72 10	68 68 00	82 72 10
94 24 70	74 14 60	60 10 50	54 14 40		74 54 20	68 58 10	88 68 20
108 28 80	94 24 70	80 20 60	74 24 50	74 34 40	74 44 30	74 54 20	102 72 30
		108 38 70	94 34 60	88 38 50	88 48 40	88 58 30	

圖 3-28 路徑規劃過程圖

在這個過程中，*Parent* 會因為計算成本的大小而被更新，因此許多格子的成本會有所變動，而在每回合開始搜尋的節點都會記錄前一回合的節點做為其 *Parent*，只要最後由終點開始往回指向 *Parent* 就可以回到起點，路徑也因此規劃出來（紅色圓點）。因此根據上述方法，實作時可依偵測到之不同障礙物及動態行為設計不同大小的成本來進行演算法運算達到路徑規劃的功能。

表 3-5 路徑規劃演算法

Step 1: Map creation

If *DisZ* is greater than 10 m and less than 20 m

$$M = M_{top}$$

Step 2: Smoothing

$$M_{smooth} = Gaussian_smooth(M, mask_size)$$

Step 3: Dilation

$$M_{Dilation} = Dilation(M_{smooth}, times)$$

Step 4: Path finding

$$path = Astar(P_{start}, P_{end})$$

End If

實作過程如表 3-5，系統在進行路徑規劃前會先確定障礙物是否在 10~20 m 之間，平滑化運算需輸入遮罩大小，接著系統會根據輸入的車身大小進行膨脹次數 *times* 的估算並呼叫其函式。主要核心路徑規劃演算法已設定成僅需輸入地圖上的起點 P_{start} 與終點 P_{end} 兩個引數，即可得到路徑規劃後所有的座標點集合 $path$ 。

3.5.4 避障標誌辨識方法

經過路徑規劃後雖然可得到完整的建議行走路線，但實際上若要將整個地圖呈現給使用者我們認為資訊是過多的，也會造成使用者解讀上的不便，如此一來便可能造成駕駛人分心而增加發生事故的機率。從原始動機來看，導航的目的是用以輔助駕駛，利用圖騰或是文字的方式來給予指引會方便許多，因此我們只要根據路徑規劃後的地圖定義其代表的意義並賦予號誌，便能達到避障目的，我們將其分成三類：「直行」、「靠右閃避」、「靠左閃避」（如下圖 3-29 所示）。

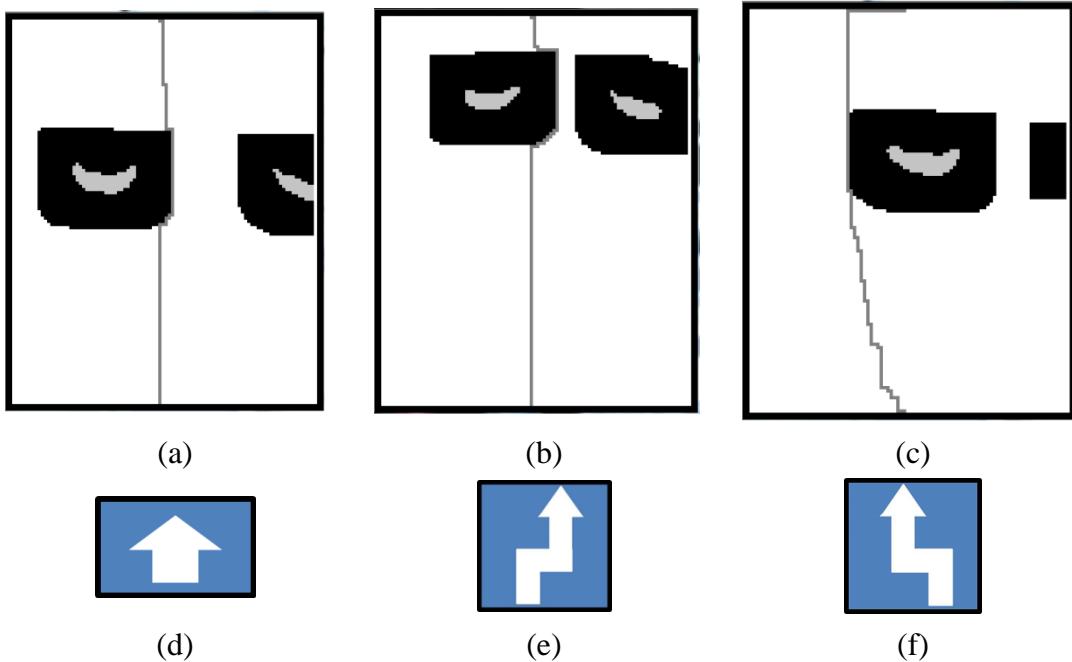


圖 3-29 路徑規劃分類及其代表的標誌 (a) 直行 (b) 靠右閃避 (c) 靠左閃避
(d) 直行標誌 (e) 靠右閃避標誌 (f) 靠左閃避標誌

類別數決定之後，便要在每次進行路徑規劃後進行辨識。利用決策樹加上兩次分類即可將此三個類別有效區分，區分的策略如下圖 3-30 所示。路徑規劃的結果發現我們所定義的「直行」其路徑長為小於一個閾值（預設為 132），再加上二階導數找出凹口的方向就能完成。使用二階導數來辨識僅須找出起始點 P_{start} 、反曲點 P_{check} 、終點 P_{end} 之三個點並連續做兩次微分，在此我們定義二階導數結果若是正數則辨識結果為「靠右閃避」、負數則為「靠左閃避」，如下表 3-6。

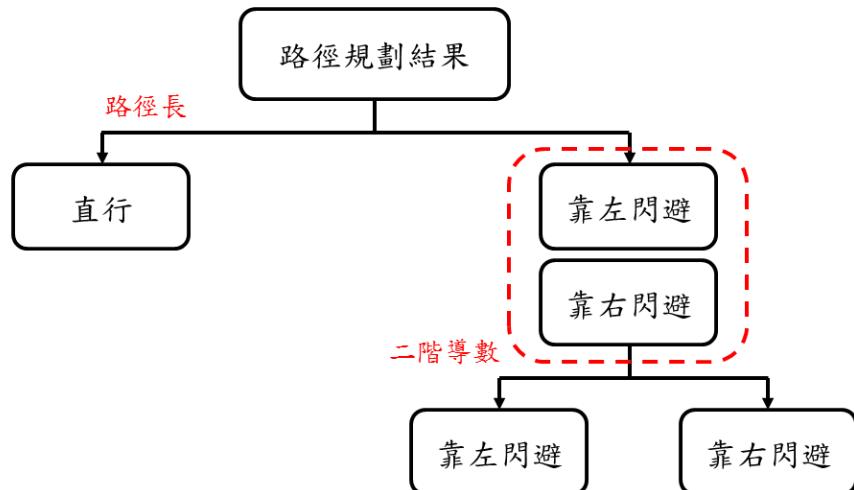


圖 3-30 避障標誌辨識流程圖

表 3-6 路徑規劃演算法

Navigation_State(path)

```

For  $i=2$  to size of path
    path_length += dis(path[i], path[i - 1])
End For
For  $j=1$  to size of path
    If path[j - 1]x is less than mid_point and path[j - 1]x is less than  $P_{check}$ 

```

$P_{check} = path$



```

End If

If path[j-1]x is greater than mid_point and path[j-1]x is less than Pcheck
    Pcheck = path

End If
End For

A = Diff_1(Pend, Pcheck)

B = Diff_1(Pcheck, Pstart)

C = Diff_2(A, B)

If path_length is not greater than thresholdState
    Return Straight

Else If path_length is greater than thresholdState and C is positive
    Return Turn Left
Else
    Return Turn Right
End If

```

3.5.5 障礙物動態行為分析

當障礙物可藉由障礙物偵測與追蹤方法在影像序列上進行連續的偵測與標定，以及時間上連續的位置關係後，便可進一步計算障礙物之行進狀態，並提供駕駛人了解車前障礙物之動態資訊，以供駕駛人提早避開可能發生的意外狀況。在本研究中，障礙物狀態分別以紅色箭頭表示「逼近」、綠色箭頭表示「遠離」，並且依據障礙物的歷史資訊計算出固定時間內之位移量來表示速度大小，並以箭頭長短表示，如下圖 3-31。

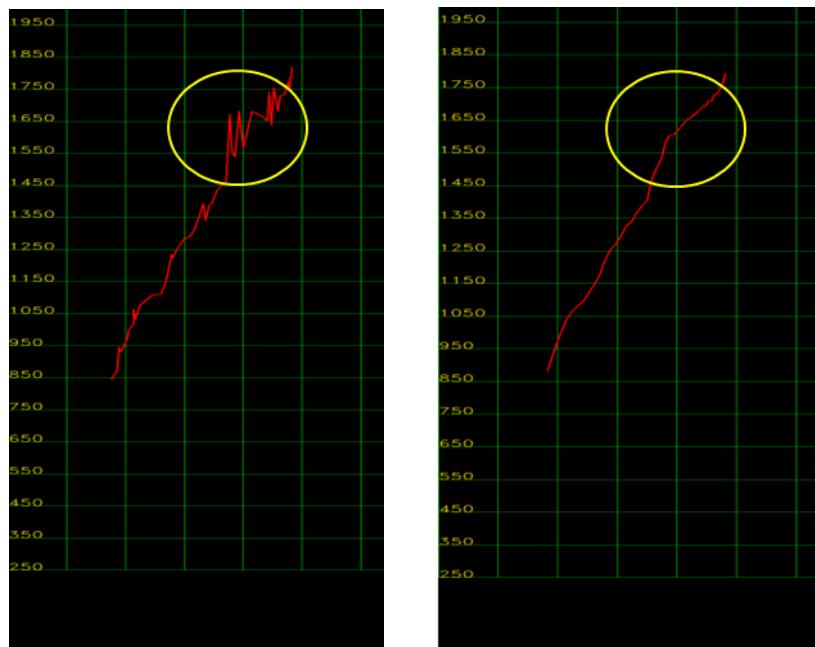


(a) 障礙物逼近 (b) 障礙物遠離

圖 3-31 障礙物狀態示意圖

由於立體視覺計算像差影像之誤差而造成距離估測的錯誤以導致障礙物狀態之誤判，我們在紀錄障礙物每一個時間的位置前先進行一次 $L=5$ 之空間平滑化，如下式表示。以實際一個持續逼近的障礙物為例子，畫出處理前後的軌跡如下圖 3-32，顯示經過平滑化後的結果較符合實際障礙物行走路線，如此可獲得更準確判斷障礙物狀態。符號說明： x 和 z 分別為障礙物於影像中之 x 座標以及平均深度 z 。

$$f'(x, z) = \frac{1}{L} \sum_{i=-\frac{L-1}{2}}^{\frac{L-1}{2}} f(x-i, z-i) \quad (3.43)$$



(a) 平滑化前 (b) 平滑化後

圖 3-32 障礙物空間平滑化前後軌跡



3.5.6 警示系統與導航

抬頭顯示器 (Head up display) 簡稱 HUD，是目前普遍應用在航空的輔助儀器，最早運用在軍事上，其目的是希望駕駛員或操作員可不需低頭就能看到所需要的資訊，也因降低低頭查看的頻率，避免注意力因中斷而造成駕駛上的危險。因為 HUD 的便利性以及能夠提升安全，隨後汽車開發廠也陸續將類似的功能應用在自家開發的車輛上。以汽車而言，最早使用抬頭顯示器是在 1988 年的美國通用汽車 (General motor) 公司，並以彩色顯示，德國的寶馬 (BMW) 是歐洲第一個提供 HUD 顯示裝置的汽車製造業者，這些汽車公司通常運用符號、圖示或數字顯示車子導航系統或車速。

若要將 HUD 的概念引入本系統，勢必要先定義圖形或符號代表的意義，以及需要顯示的資訊，以下將利用列舉的方式說明：

- **障礙物偵測**

障礙物偵測以簡單幾何形狀表示，障礙物顏色分別以紅、黃、綠表示。

- **障礙物速度**

障礙物速度為代表與攝影機之實際相對速度表示，單位為公里/時速表示，顏色的顯示與偵測之障礙物相對應。

- **動態行為**

動態行為可分為指向上或指向下，各代表與攝影機之相對遠離或逼近，顏色與速度和偵測之障礙物相對應，此外依據障礙物在實際影像中的橫向位置進行三等分並進行投影，如下表，可看出黃色箭頭代表障礙物逼近，且位於攝影機的前方靠右處。

- **警示系統**

警示系統包含文字與燈號，根據偵測出最靠近之障礙物的距離進行分級，距離由近至遠依序顯示「BRAKE」、「ALARM」、

「ALERT」、「CLEAR」。

- 導航系統

導航系統為避障的主要目的，以不碰撞到最靠近之障礙物進行導引，

可分為三個狀況，「直行」、「靠左閃避」、「靠右閃避」。

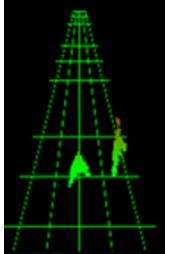
- 上視圖

透過上視圖是最能夠了解前方資訊的方法之一，然而駕駛員僅需要了

解前方的概況即可，解析度不需太高，因此以解析度 31×25 顯示。



表 3-6 HUD 避障模式資訊對照表

障礙物偵測			
障礙物速度		$V_1 = 3.54 \text{ km/hr}$ $V_2 = 5.86 \text{ km/hr}$	
動態行為	指向上 指向下	遠離 逼近	
警示系統	BRAKE	$D \leq 10 \text{ m}$	
	ALARM	$10 \text{ m} \leq D \leq 15 \text{ m}$	
	ALERT	$15 \text{ m} \leq D \leq 20 \text{ m}$	
	CLEAR	$D \geq 20 \text{ m}$	
導航系統	直行		
	靠右閃避		
	靠左閃避		
上視圖	解析度 31×25		



3.6 系統驗證與實驗方法

為了證明前述方法之可行性，以下將敘述系統進行驗證的項目以及實驗設計：

- 立體視覺系統

立體視覺的驗證主要是計算系統對距離估測的準確度，實驗時先固定攝影機於高度 110 公分，並分別在攝影機前方 5、10、15、20 公尺各放置一個高度約 68 公分的交通錐，藉由立體視覺理論計算得到交通錐的距離並計算與實際位置的誤差百分比。

- 障礙物速度模型的建立

由於系統偵測到行人的最遠距離約為 15 公尺，因此利用碼表量測行人於偵測範圍內直線等速度移動的時間，並利用碼表紀錄讀數換算成速度當作 Ground truth，並同時利用立體視覺估算出起始點與終點的距離除以框架速度得到系統的估算值，最後以線性迴歸建立模型。

- 障礙物偵測、障礙物追蹤、路徑規劃與避障

實驗場景主要有 3 個，依據環境複雜度由低至高依序是台大農場、台大校園、市區道路。障礙物數量為 1~4 個不等，並以三到五次的連續影像序列進行實驗結果的統計，其中台大校園、市區道路皆是將雙眼立體視覺系統實際裝設於車艙內的擋風玻璃上進行（如下圖 3-33），裝設的高度皆為 110 公分。偵測的距離範圍由於校園內多為行人或腳踏車，在障礙物設定條件裡曾提到障礙物在影像中所佔的像素需大於 300 的 pixel，因此偵測範圍只有設定到 20 公尺，而市區的車輛在影像中所占的面積較大（Pixel 數多），因此可以延伸偵測範圍到 30 公尺。

表 3-5 實驗設計相關參數

	場景一	場景二	場景三
實驗地點	台大農場	台大校園	台北市市區道路
攝影機高度	110 公分	110 公分	110 公分
偵測範圍	20 公尺	20 公尺	30 公尺
障礙物數量	1~3	1~4	1~3
障礙物種類	人、曳引機 、中耕機	腳踏車、行人	機踏車、小客 車、公車



圖 3-33 立體視覺架設於車艙內之示意圖

第四章 結果與討論



4.1 立體視覺

4.1.1 攝影機校正

進行攝影機校正時，為了使左右攝影機取得左右影像角點的資訊，利用二維校正用的方格版，長 120 公分，寬 90 公分並貼上印製角點數為 9×6 的棋盤方格作為校正物，如圖 4-1，方格大小為長寬各 11.7 公分，立體視覺各自拍攝 50 個不同角度的方格版影像進行校正。



圖 4-1 校正用的棋盤方格影像，擷取自 (a) 左攝影機 (b) 右攝影機

有了校正版影像之後，可以進一步利用角點偵測函式找出格子點的角點，並可計算出單科攝影機的內參數與畸變參數，此外藉由攝影機校正，亦可得出攝影機在畫面 x 、 y 方向的中心位置。由表 4-1 為校正之參數，根據賴 (2012) 的描述， x 、 y 方向的焦距計算出來的值差不多，因此實作時以等值帶入計算深度對實驗結果的誤差影響極小。此外，解析度為 640×480 的攝影機，再經過校正後的中心位置，也就是影像座標 $(320, 240)$ 處所指向的物體高度應與攝影機高度相

同，但是由於每次車輛應用實驗時，立體視覺裝設在車輛上會因車輛震動而造成攝影機鏡頭有些許的鬆動，因此實驗前進行一次手動校正，如圖 4-2 所示，假設攝影機高度為 110 公分，則障礙物不論在 5 公尺處或是 10 公尺處所指向的 110 公分高度應同在影像畫面的中心，若不符合則些微調整攝影機 x 方向的旋轉角度即可，經過手動微調鏡頭來校正後 x、y 方向的畫面中心位置則可以信賴，如此一來可省去每次實驗都需拍攝二維校正的麻煩。

表 4-1 攝影機內參數

攝影機位置	f_x (pixel)	f_y (pixel)	c_x (pixel)	c_y (pixel)
左攝影機	3545.0	3545.0	手動校正	手動校正
右攝影機	3545.0	3545.0	手動校正	手動校正

f_x 、 f_y ：x、y 方向的焦距， c_x 、 c_y ：x、y 方向的畫面中心

獲得每台攝影機的畸變參數就能消除各個攝影機的影像扭曲變形問題，如圖 4-3，校正前後的左右影像皆從原本的矩形畫面變成些微的曲面，畫面中看不出明顯變化是由於選用的鏡頭焦距為 16 mm，若使用更短焦距的鏡頭如魚眼，則校正的效果會更加明顯。而透過校正後的像差影像，可看出資訊豐富了許多，表示各像素點均能在左右影像找到其匹配點以計算深度，像差影像越偏紅色區域表示其距離攝影機越遠，越偏藍色則表示越近。

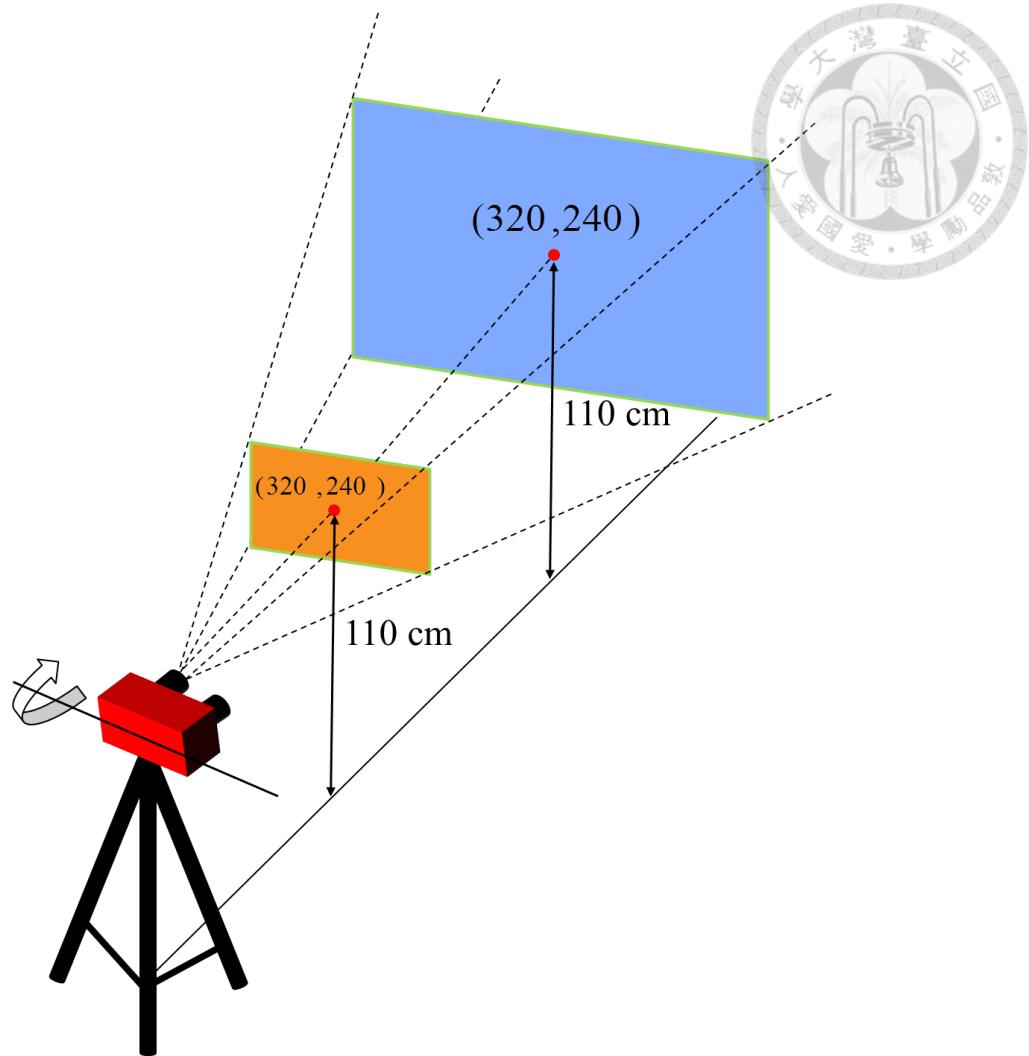


圖 4-2 攝影機畫面中心校正示意圖，

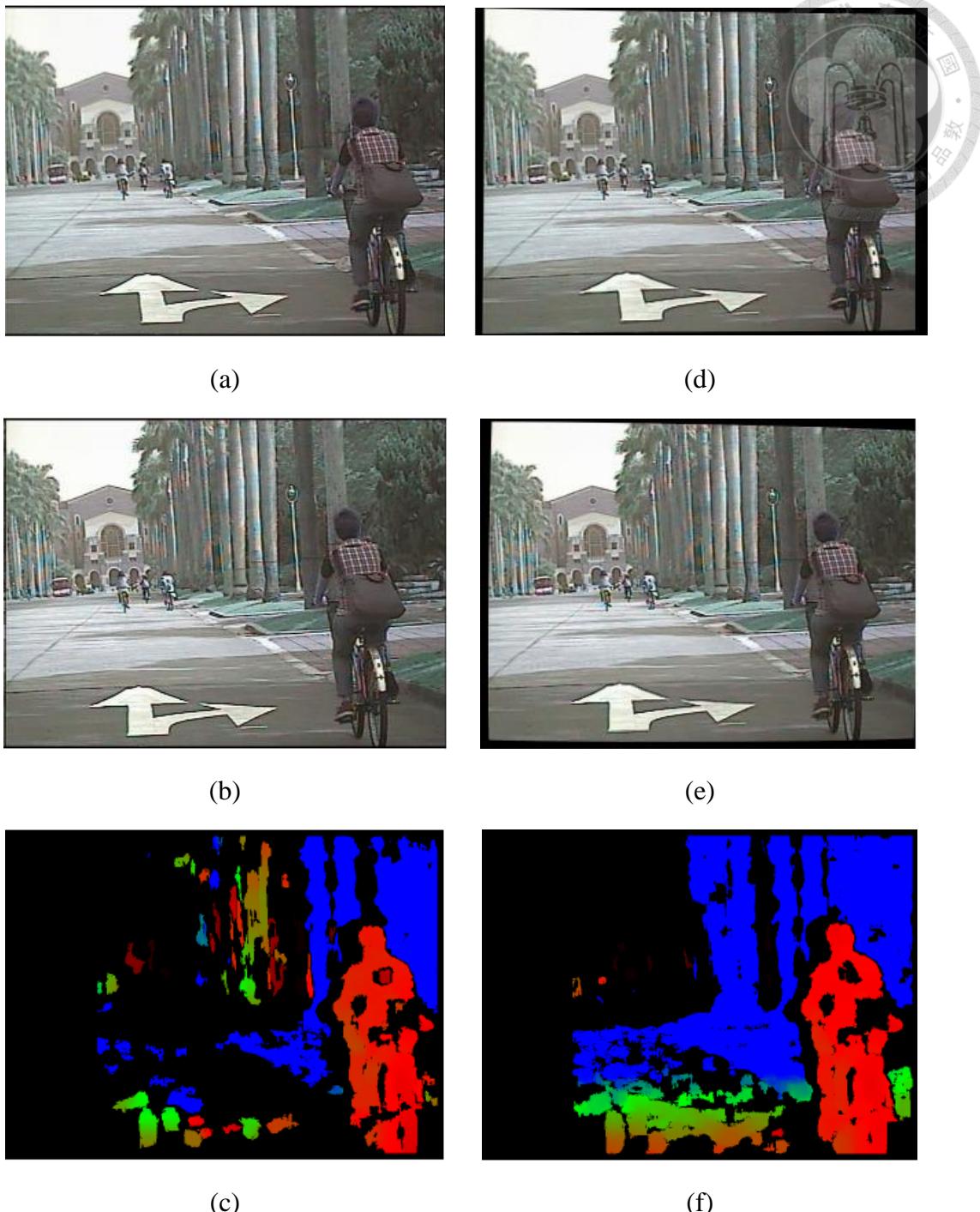


圖 4-3 校正前後比較圖，校正前 (a) 左 (b) 右 (c) 像差影像；
校正後 (d) 左 (e) 右 (f) 像差影像



4.1.2 像差影像

經過完整的校正後，立體視覺系統所拍攝的左右攝影機應滿足同軸的幾何限制，因此可進一步進算像差影像，根據第三章提出的半區域式區塊匹配 (Semi-Global Block Matching, SGBM) 方法便可計算每個像素的三維資訊，然而為了更容易觀察影像中距離的分布情形，利用 Pseudo color 將計算所得到偵測範圍切割成 256 等分，距離攝影機由近到遠分別以紅色到藍色表示，其中若某些像素因左右影像找不到對應點而無法計算深度或是超出偵測距離的範圍則以黑色表示，如圖 4-4。圖 4-4 (b) 像差影像的左側之所以是黑色，乃因左右影像之中必須有重疊才可找出對應點進行計算深度。

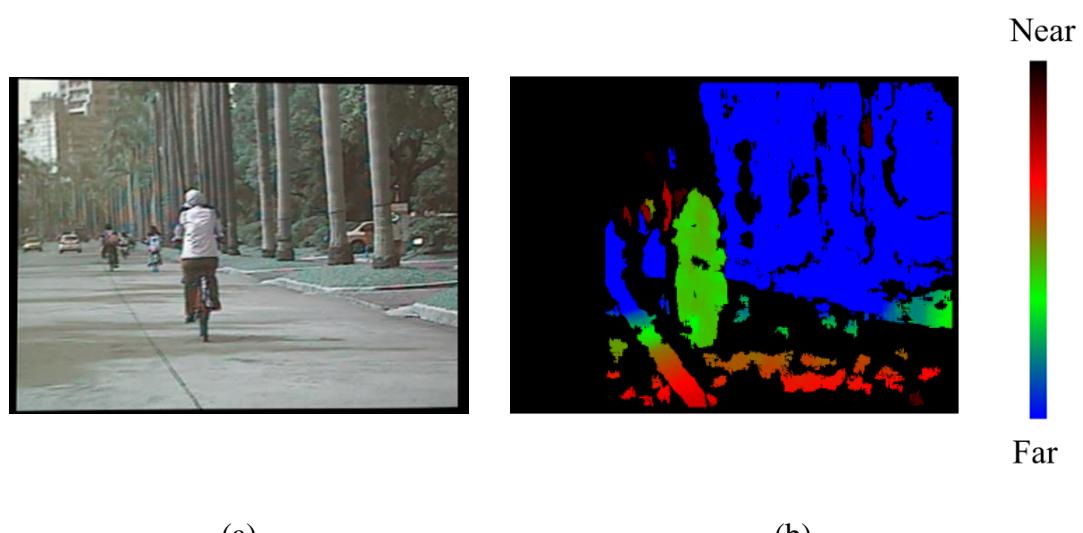


圖 4-4 像差影像結果圖 (a) 原始影像 (b) SGBM



4.1.3 距離估測

在經過攝影機校正、圖像比對等步驟後計算像差影像後，必須進一步探討與驗證其估測之三維資訊的準確性。我們以固定攝影機高度 110 公分進行前方 20 公尺以內的障礙物偵測，障礙物以交通錐（高度 68 公分）表示，分別擺設於距離立體視覺攝影機 5、10、15、20 公尺處，並進行立體視覺的深度計算。經過距離的估算誤差統計後如圖 4-5 所示，結果顯示距離攝影機越遠時，距離估測誤差率有越高的趨勢，探究其原因較可能為攝影機影像解析度不夠高，因而造成距離較遠時在影像上的像素差較不明顯，因此計算像差值的精準度下降，此結果是可預期的，但整體誤差率仍低於 4%，顯示距離估測的準確度仍相當高。

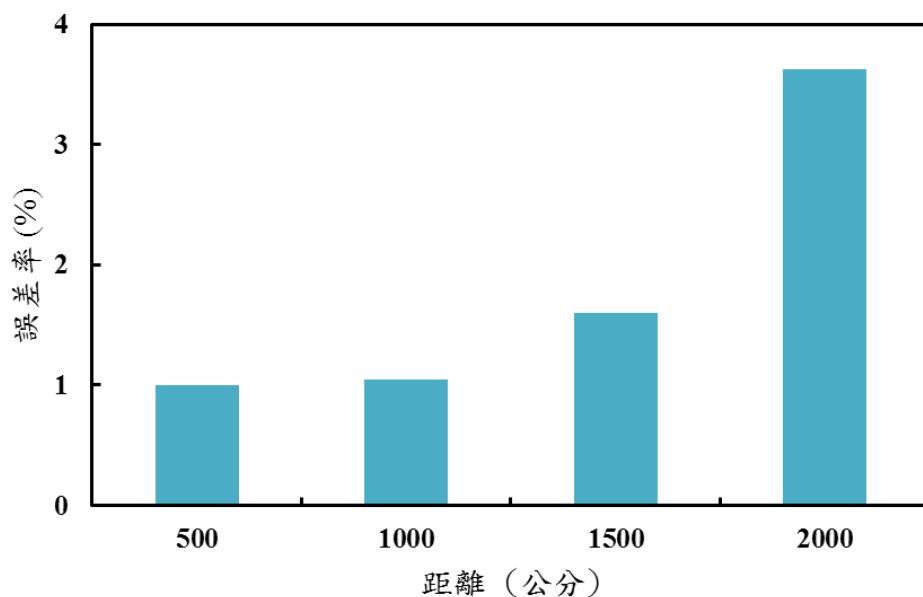


圖 4-5 立體視覺之距離估測誤差



4.2 障礙物偵測結果

回顧第三章所提出的障礙物偵測方法，是先將像差影像投影至上視圖，並利用團塊法將障礙物由深度影像中計算且標示出來，並配合特定障礙物的特徵與幾何條件的限制將障礙物從影像中找出，實作結果如下圖 4-6。在其間必須決定障礙物的表現方式，文獻中提出了幾種表現方式，其中包含輪廓以極簡單幾何形狀，在本研究中所開發的系統可以輕易從像差影像大約取出障礙物的形狀，請參考圖 4-7 (b)，然而這種表現方法較接近文獻中的輪廓，但是卻容易因為有些像素無法找到匹配點而不完整。因此，為了得到較美觀的顯示介面，我們將已標定的障礙物中，利用屬於該障礙物的像素點之最左上角與右下角的點計算長與寬進行框選，並對應到原始影像來顯示。

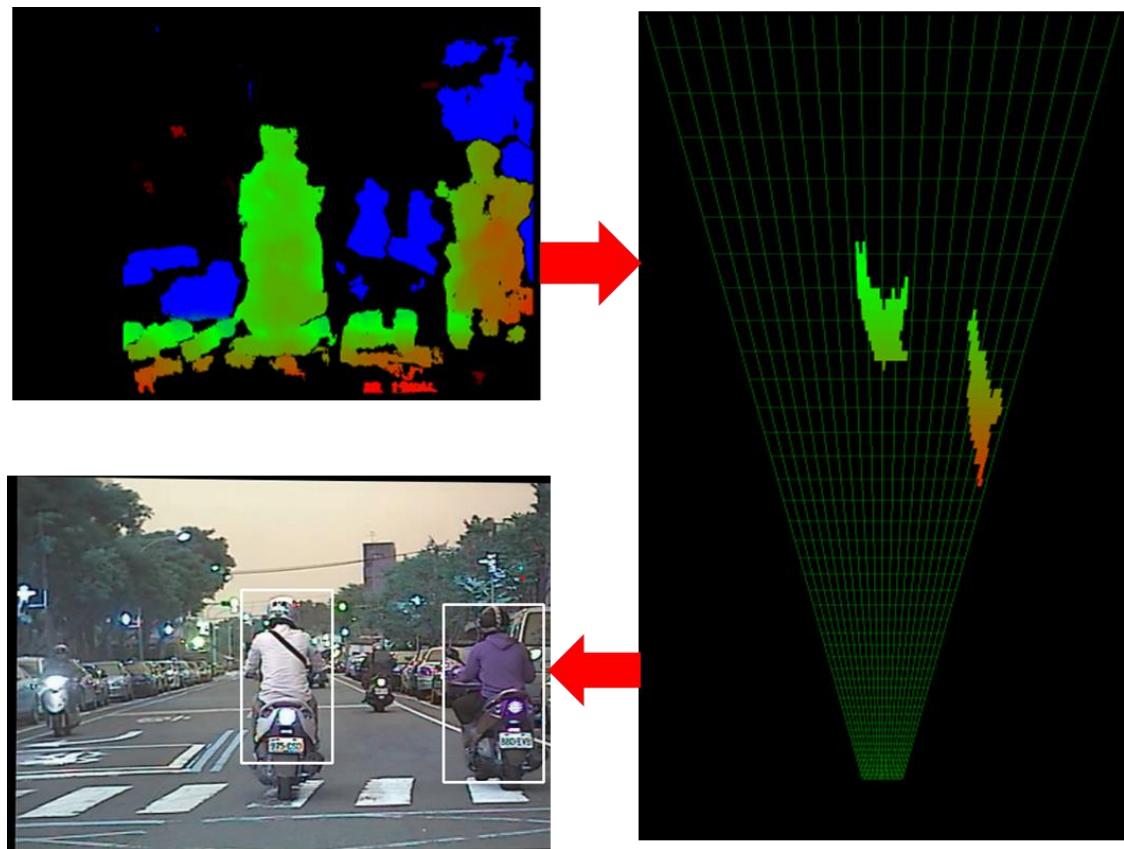


圖 4-6 障礙物偵測之結果與顯示方法



(a)



(b)

圖 4-7 障礙物表現示意圖 (a) 簡單幾何形狀 (b) 輪廓

經由前述的方法，為了驗證系統的可行性，將針對農業與車輛的應用進行實驗設計，共可分為台大農場、台大校園、市區道路三個場景，並各自以三到五次的連續影像序列進行實驗結果的統計，其中台大校園、市區道路階是將雙眼立體視覺系統實際裝設於車艙內的擋風玻璃上進行，裝設的高度皆為 110 公分。偵測的距離範圍由於校園內多為行人或腳踏車，在障礙物設定條件裡曾提到障礙物在影像中所佔的像素需大於 300 的 pixel，因此偵測範圍只有設定到 20 公尺，而市區的車輛在影像中所占的面積較大 (Pixel 數多)，因此可以延伸偵測範圍到 30 公尺。

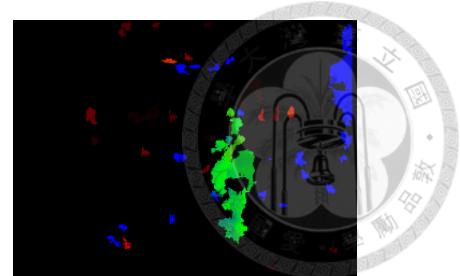
實驗結果如表 4-2，我們針對偵測並標定的物體是否為障礙物的統計。農場實驗中 (如圖 4-8)，主要障礙物為行人、曳引機和中耕機偵測的農場長 28 公尺寬 20 公尺，由於實驗環境複雜度較低，障礙物偵測的結果較理想；在校園中 (如圖 4-9)，我們所設定的幾何限制條件是希望能夠偵測到行人或腳踏車，該實驗中

非障礙物而被視為障礙物的情形多為樹木或草叢，行人在影像中所占像素較少，因此同一張影像中障礙物數量最多可達 3 個，根據上表結果統計出該場景的障礙物偵測準確率可達 90% 以上；市區障礙物偵測實驗中（如圖 4-10），我們希望能偵測到前方的公車、汽車或是機踏車並依此為判定偵測障礙物是否正確的準則，實驗結果發現偵測錯誤的物體多為電線桿或是路邊草叢，而因硬體在視角上的限制以及車輛安全距離上的顧慮，影像中同時偵測到的障礙物最多僅有 2 個，而結果統計市區障礙物偵測實驗的偵測準確率可達 95% 以上。

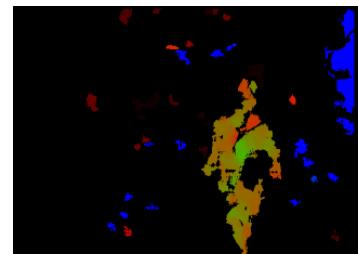
表 4-2 各場景中障礙物偵測的結果

Scenario	Complexity	Range of detection distance	Maximally detected class of vehicle	Result
Farm	Low	7~30 m	2	Frames 610
				Total detected obstacle 1020
				Incorrect detection 0
				Correct detection 1020
Campus	Medium	7~20 m	3	Frames 353
				Total detected obstacles 418
				Incorrect detection 33
				Correct detection 385
Urban	High	7~30 m	3	Frames 409
				Total detected obstacles 552
				Incorrect detection 20
				Correct detection 532

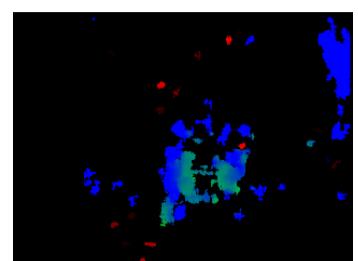
#80



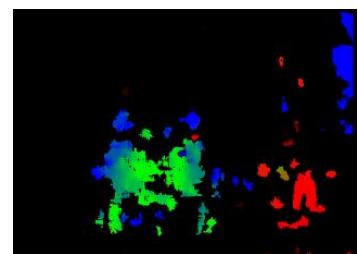
#150



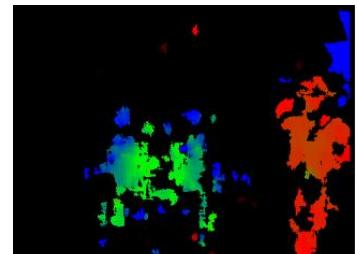
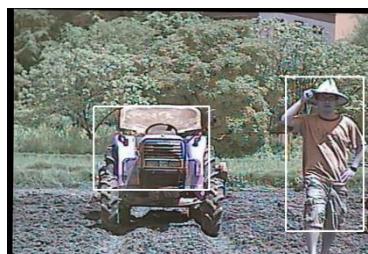
#200



#500



#700



#950

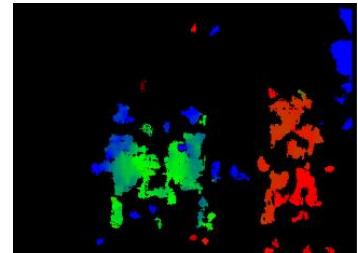


圖 4-8 立體視覺之影像序列於農場中的障礙物偵測結果

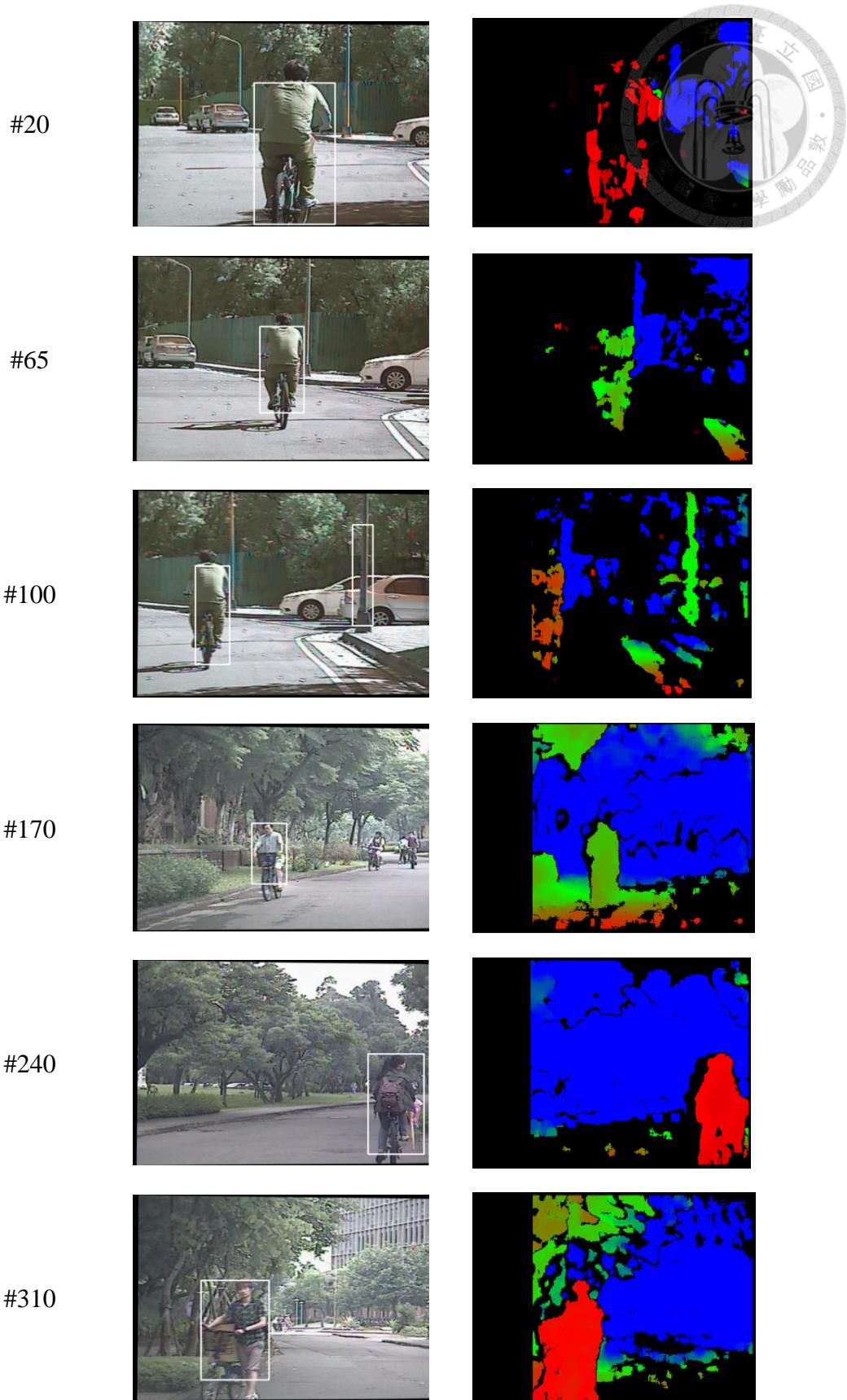


圖 4-9 立體視覺之影像序列於校園中的障礙物偵測結果

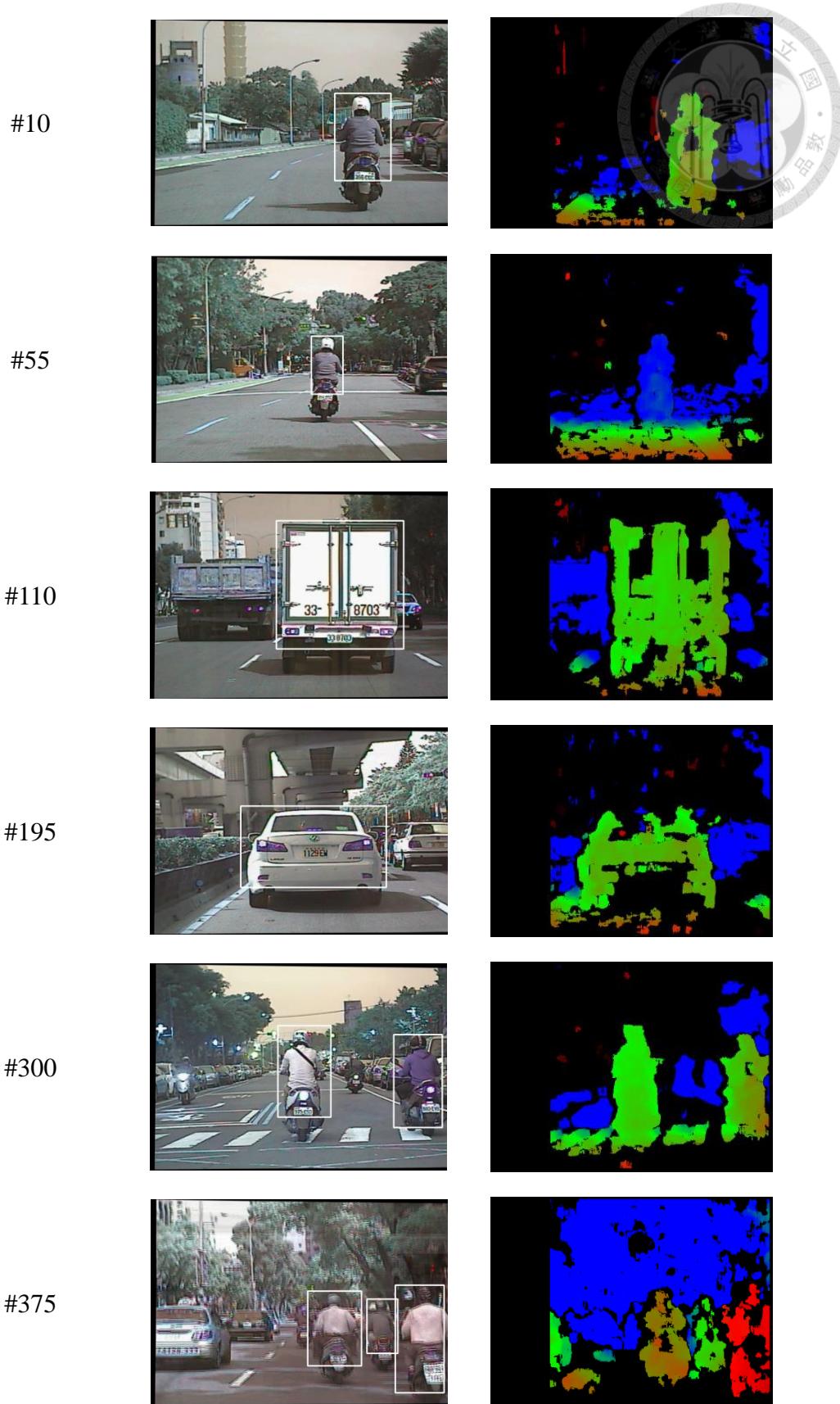


圖 4-10 立體視覺之影像序列於市區的障礙物偵測結果



4.3 追蹤方法探討

4.3.1 不同特徵之追蹤效果

本文提出之方法為結合距離與色彩資訊進行追蹤，為此我們進行兩次追蹤成功率的實驗進行估算，該影像序列中的障礙物數目為 1 到 3 個不等，偵測範圍於 20 公尺之內且在無複雜環境之場景，障礙物為大小曳引機與行人為主。實驗所偵測的障礙物數量，以及進行單一特徵與結合兩特徵之比較結果，統計後見下表 4-10。實驗結果可看出僅使用距離資訊的追蹤率相對低，探究其原因為多個障礙物並排之情形較為常見，易導致匹配錯誤，其上視圖與匹配錯誤例子如下圖 4-11。另一種造成匹配錯誤的情形為色彩資訊相似度太高，如下圖 4-12 可看出兩曳引機的統計直方圖相近，因此計算出接近的巴式距離造成匹配錯誤。因此我們提出之結合距離與色彩資訊能有效改善以上兩者匹配錯誤的情形，結果如下圖 4-13。

表 4-3 不同特徵之成功追蹤率

Farne 數	偵測到障礙物數	方法一			方法二	方法三
		(距離)	(色彩)	(色彩+距離)		
實驗 1	300	541	92.4%	97.0%	98.2%	
實驗 2	400	660	90.5%	97.1%	97.5%	

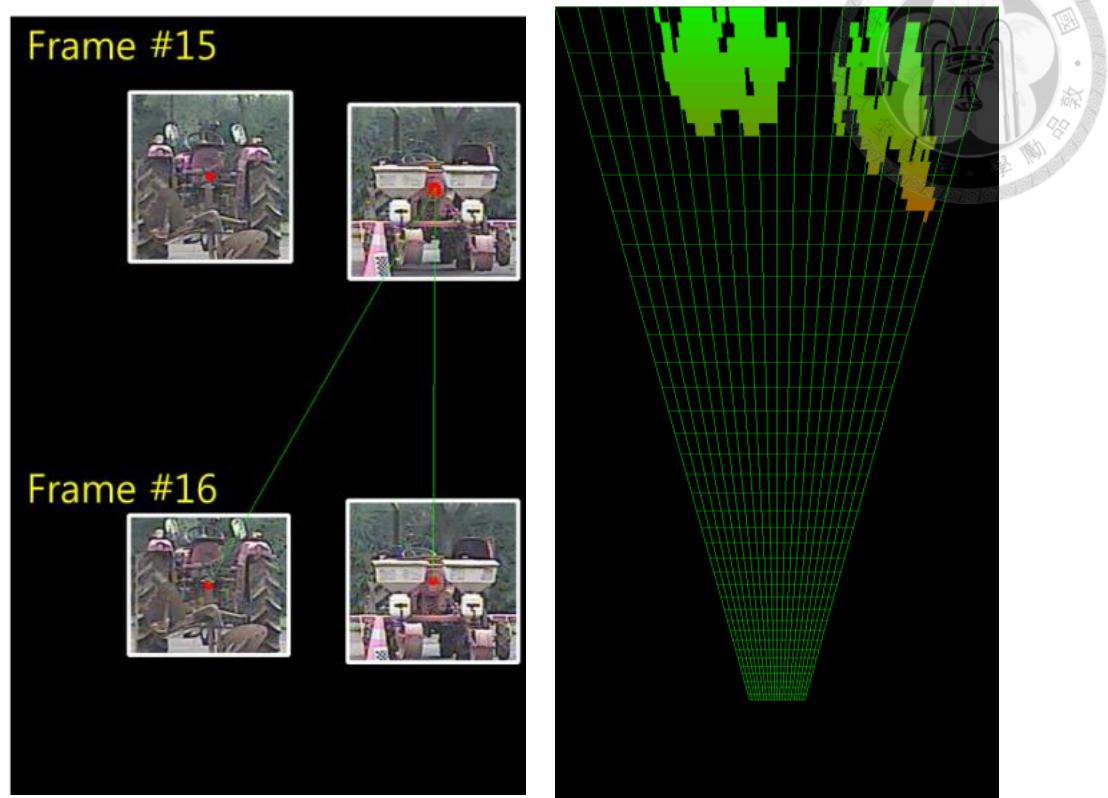


圖 4-11 因距離相同匹配錯誤之結果

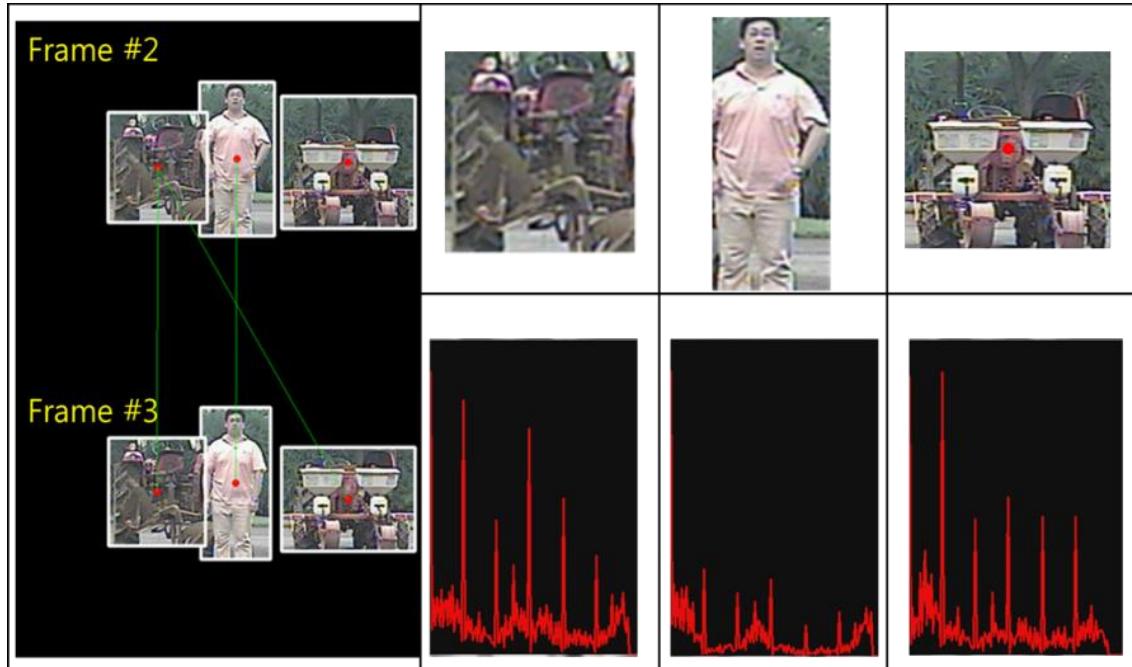


圖 4-12 因色彩資訊相似匹配錯誤之結果

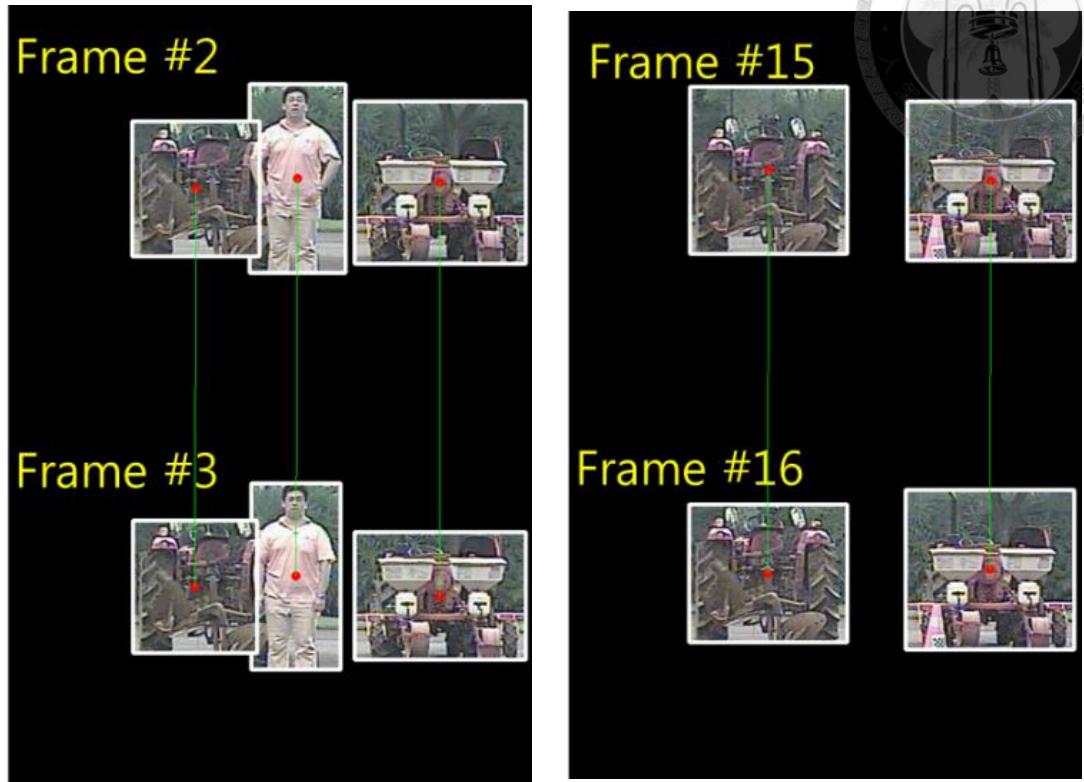


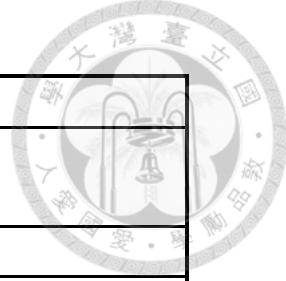
圖 4-13 結合距離與色彩資訊之結果

為了探討增加色彩資訊是否對整體追蹤效果有顯著提升，因此隨機選取 5 個實驗影片，並進行以 $\alpha = 0.05$ 檢定結合兩特徵之平均效果是否高於距離特徵之 ANOVA 統計分析，追蹤成功率如下表 4-4。根據表 4-5 之統計報表顯示，檢定統計之 F 值為 10.55，具顯著差異，故拒絕虛無假設（方法一的平均效果小於或等於方法二），即結合距離與色彩特徵有顯著提升追蹤成功率。

表 4-4 不同影片之追蹤成功率

方法一 (距離+色彩)	98.6%	97.8%	94.8%	98.9%	94.5%
方法二 (距離)	93.2%	95.5%	84.5%	87.4%	87.6%

表 4-5 單因子變異數分析之報表



單因子變異數分析						
摘要						
組	個數	總和	平均	變異數		
方法一	5	484.6	96.92	4.467		
方法二	5	448.2	89.64	20.653		

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	132.496	1	132.496	10.54904	0.011737	7.570882
組內	100.48	8	12.56			
總和	232.976	9				

4.3.2 背景影響

經過上一小節之實驗結果顯示結合距離與色彩特徵之追蹤成功率約 97.2 %。然而，在使用色彩資訊時，是利用框選出的障礙物進行計算統計直方圖，如果距離較遠或在左右影像較細小的物體容易匹配錯誤而造成深度計算上的誤差，進而影響投影至上視圖進行團塊處理時的結果，如此一來所框選出的障礙物影像會出現跳動，為了解決此問題，根據 3.4.1 節提出利用無背景之影像且不計算灰階值為 0 (黑色) 的統計直方圖取代原本含背景之障礙物影像進行實驗。實驗之影像序列中的障礙物數目為 1 到 3 個不等，偵測範圍於 20 公尺之內且在無複雜環境之場景，障礙物為大小曳引機與行人為主，偵測的障礙物數量與結果見下表 4-6，其顯示在單純環境下去除背景能克服一些系統因環境不同造成匹配錯誤的情形。

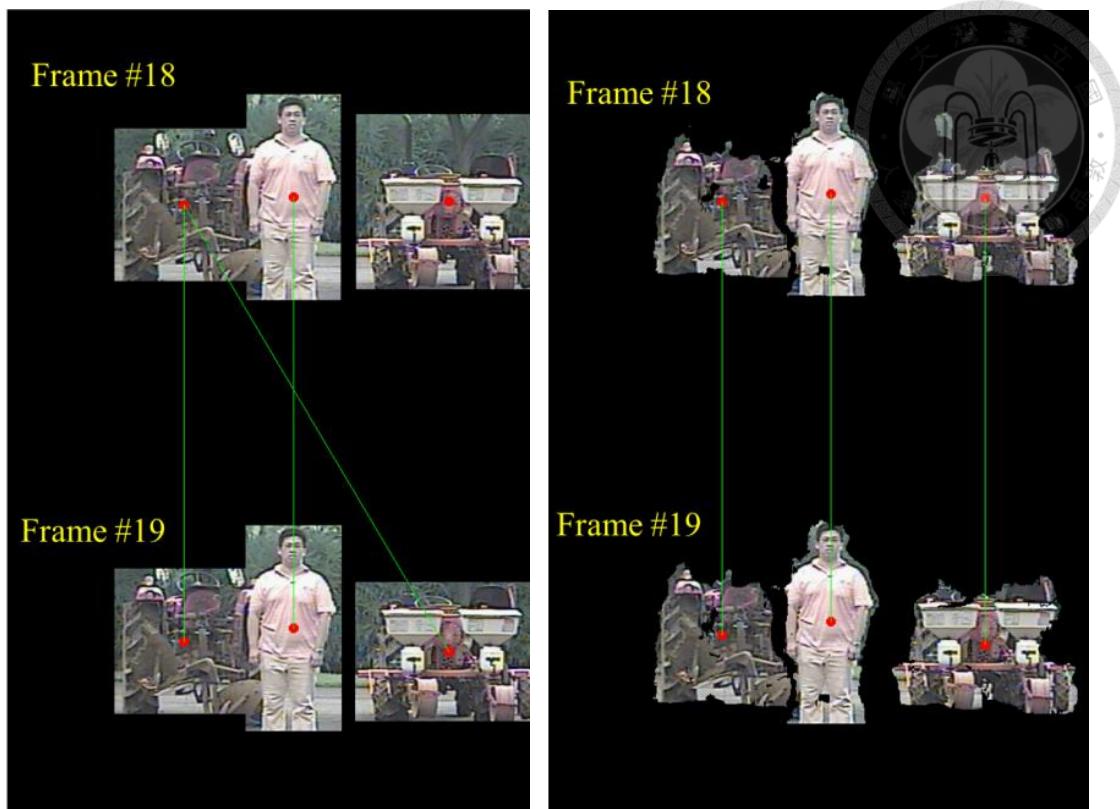


圖 4-14 背景去除前後之結果

表 4-6 去除與保留背景之追蹤成功率

Frame 數	偵測到障礙物數	方法三	
		(去除背景)	(保留背景)
實驗 3	200	612	98.1% 96.1%
實驗 4	150	555	99.0% 98.3%

4.3.3 障礙物速度估算

要能使適應性巴氏距離順利進行，最重要就是系統需估算偵測到障礙物的相對速度。根據 3.4.3 裡提出的方法，以行人進行直線的等速度移動並利用碼表紀錄讀數換算成速度當作 Ground truth，在利用立體視覺系統估算出的距離與框架速度得到系統的估算值，並利用線性關係進行模型的建立，結果見下圖 4-15。其結果顯示若偵測到的障礙物相對速度介於 4~12 (km/hr) 之間，可利用建立的模

型進行系統與實際相對速度的轉換。

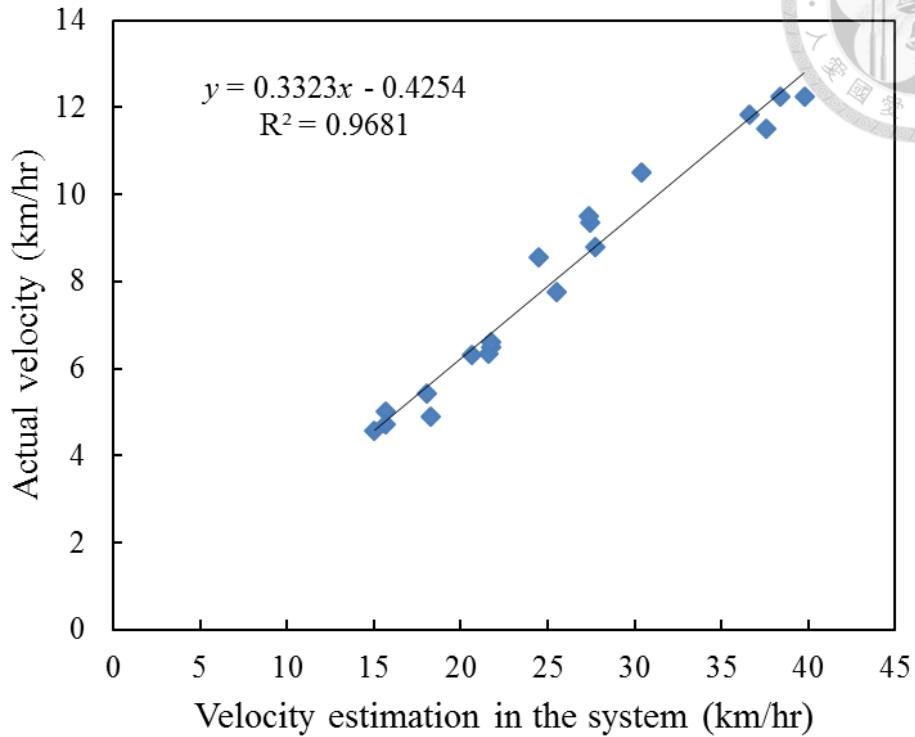


圖 4-15 系統與實際相對速度之關係

此外，為了證明所建立的模型為有效，必須進行驗證實驗。實驗同樣以行人進行直線的等速度移動並利用碼表紀錄讀數換算成速度當作 Ground truth，每次實驗影會在最後估計出一個速度值，最後計算出誤差率，結果見下圖 4-7。橫軸為實驗編號共 11 次，縱軸為實際平均速度與估計的速度分別以藍色和紅色表示，副軸表示誤差率。需注意的是，實驗結果是假設目標物為等速度的情況下所計算出的誤差率（因為實作時會在每一個 Frame 估算一個瞬時速度，而誤差是用全部來算所以必須這樣假設）。

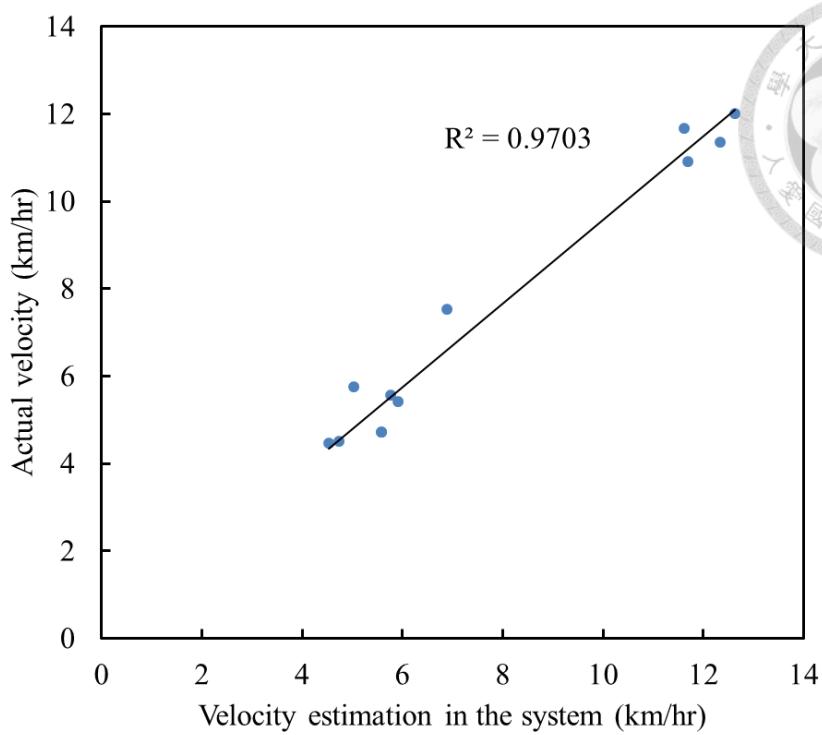
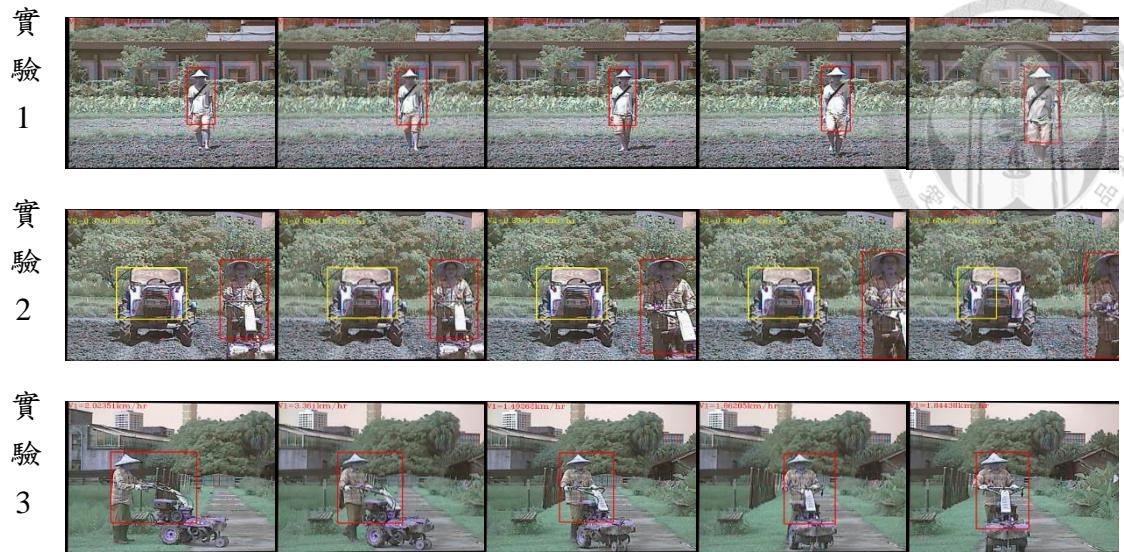


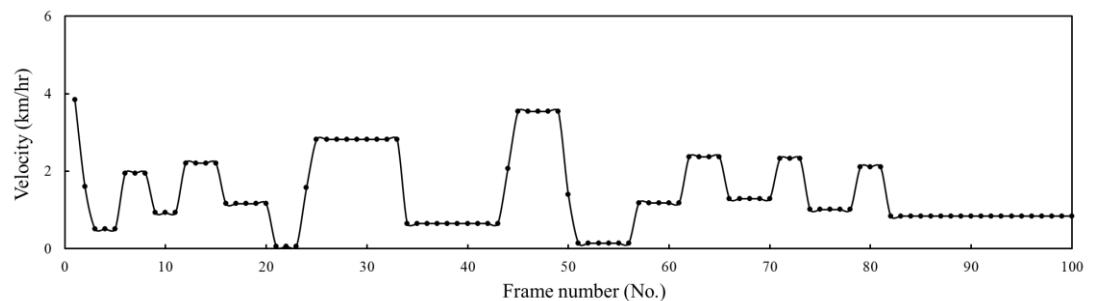
圖 4-16 速度估算驗證結果

圖 4-16 為經過驗證速度模型的結果，以線性回歸計算 R^2 約為 0.97，觀察後發現速度大小跟誤差百分比並沒有直接的相關，計算後得到的誤差百分比約為 10 %，與理論上速度大小只與障礙物距離呈線性正相關一致，且為一個誤差範圍內的跳動。

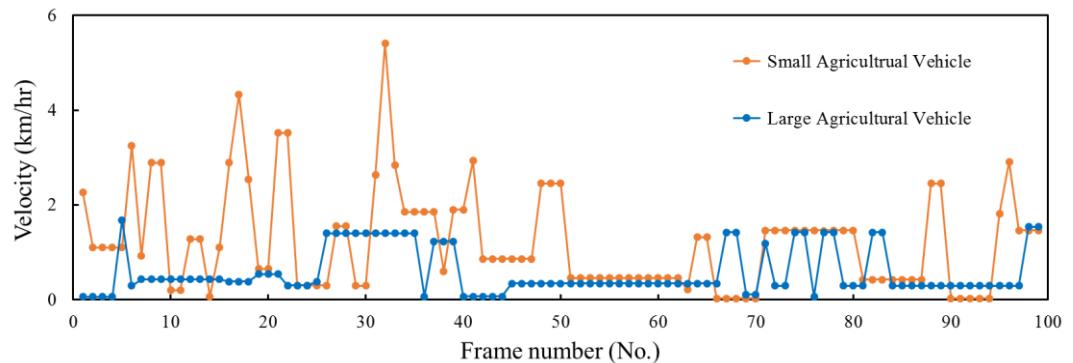
障礙物速度估算實驗共進行三次，實驗場景為台大農業試驗場，攝影機高度 110 公分，障礙物為曳引機、中耕機和農人，如下圖 4-17 所示。實驗時三種障礙物在農場中移動並記錄於影像序列中，實驗過程中相同障礙物被追蹤到會以同一顏色框選表示如下圖 (a)，若追蹤成功系統會同時記錄其位置並計算實際速度，實驗結果見圖 4-17 (b)~(d)。



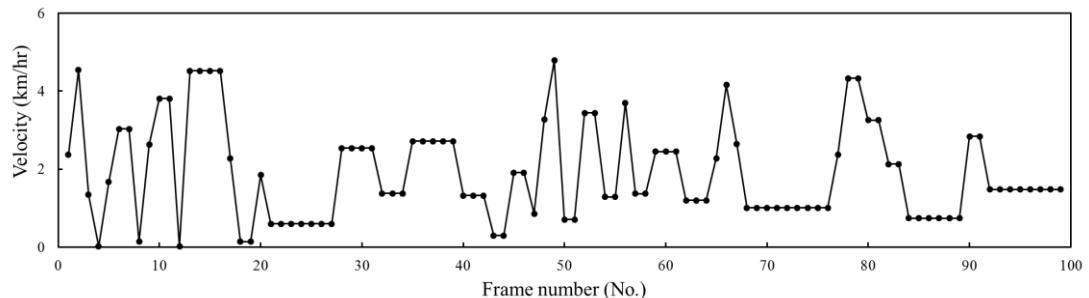
(a)



(b)



(c)



(d)

圖 4-17 障礙物速度估算實驗結果 (a) 3 個實驗之障礙物追蹤結果，(b)~(d) 實驗 1~3 障礙物速度估算結果



4.3.4 適應性追蹤方法

適應性巴氏距離閾值主要目的是為了補強障礙物偵測時因計算像差影像的誤差，所造成的障礙物影像大小跳動而追蹤失誤的情況，如同下圖 4-17 的其中一個實驗例子，已知系統偵測前方的障礙物為同一個行人，在第 11 張影像跟第 12 張影像所偵測到的障礙物雖是同一個人，但卻因為像差影像的誤差而 miss。起初定義追蹤成功建立在計算巴氏距離需在設定的閾值 0.3 以下才視為前後影像為相同障礙物，經由前述分析方法，我們可以大約計算出障礙物的相對移動速度，並且適應性的改變閾值來克服此問題。

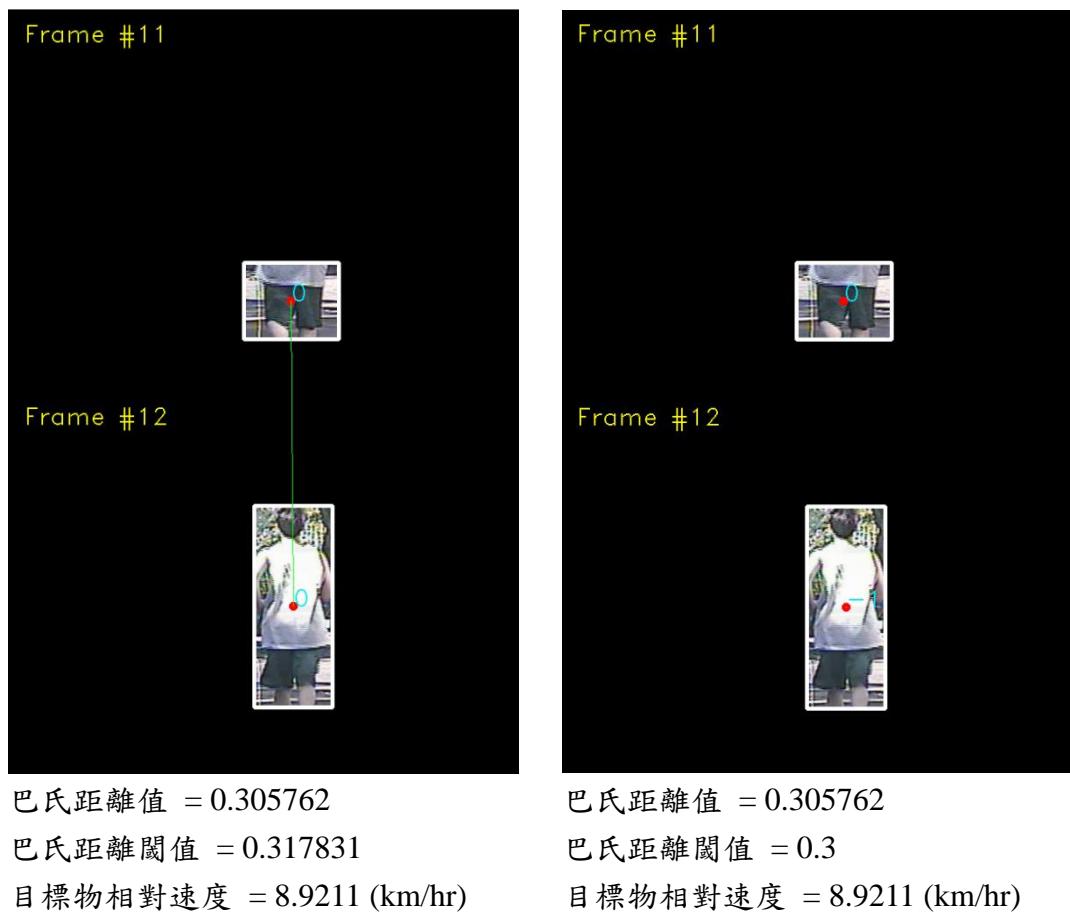


圖 4-18 適應性巴氏距離改良的結果示意圖



4.3.5 障礙物追蹤結果

為證明上述提出的方法能有效應用在農業與車輛上，依環境複雜程度的高低分成三個主要場景進行驗證，由低到高依序為台大農場、台大校園、市區道路，實驗各為 3~5 個連續影像序列來進行如下表 4-7。台大校園場景如圖 4-19，如同障礙物偵測實驗一樣，障礙物主要為行人以及腳踏車，實驗中觀察得知障礙物所佔的 Pixel 數較少對障礙物的偵測較不利，因此在相對速度較小時進行追蹤能獲得較好的連續追蹤效果，而相對速度太大則會造成偵測上的跳動，追蹤時會因為未達所設定的閾值而間斷性的追蹤，追蹤到障礙物於幾個 frame 之後便可進一步得到其瞬時相對速度，計算後會顯示在影像的左上角。市區道路實驗如圖 4-20 車輛與車輛之間相較於行人與行人之間的變化較明顯，且車輛的安全距離較大，故影像中的障礙物通常為 1~2 個，因此追蹤到的機會較大，追蹤到的車輛相對瞬時速度一樣可計算獲得。經過計算後算得追蹤演算法之系統執行運算時間平均為 11 fps。

表 4-7 障礙物追蹤於各場景相關資訊

Scenario	Complexity	Maximally detected class of vehicle	Frame
Farm	Low	2	700
Campus	Medium	3	320
Urban	High	2	272

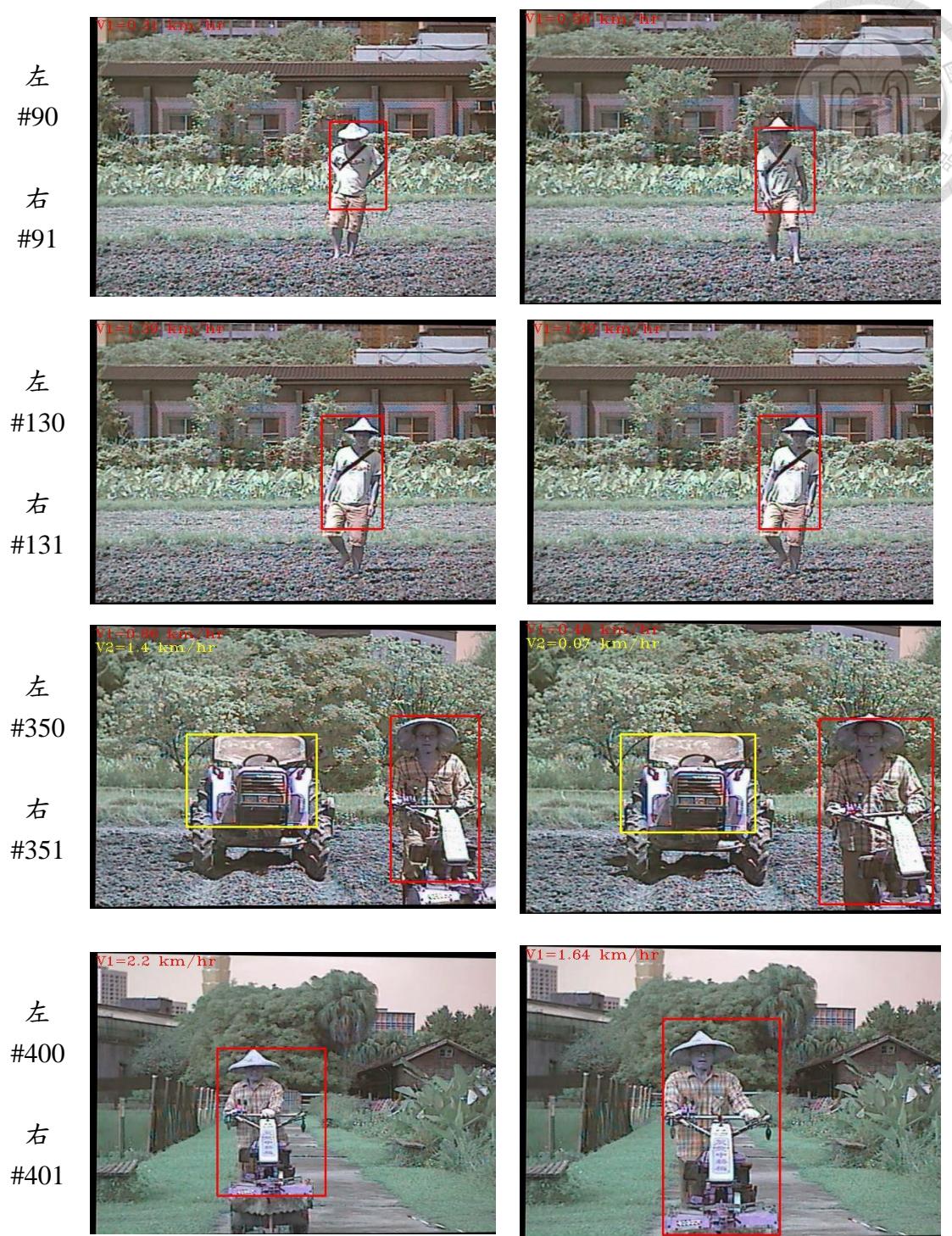


圖 4-19 立體視覺之影像序列於校園中的障礙物追蹤結果

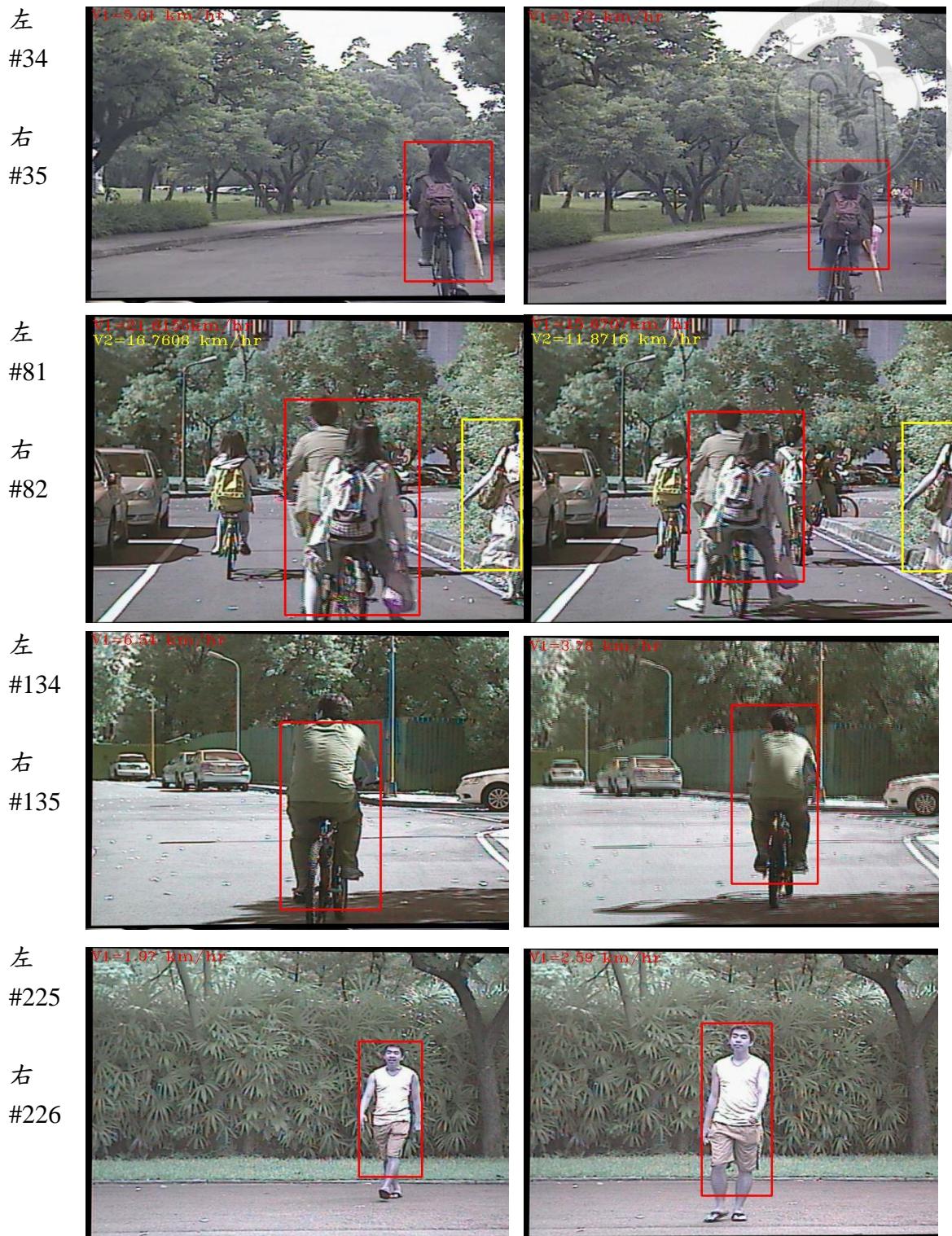


圖 4-20 立體視覺之影像序列於校園中的障礙物追蹤結果

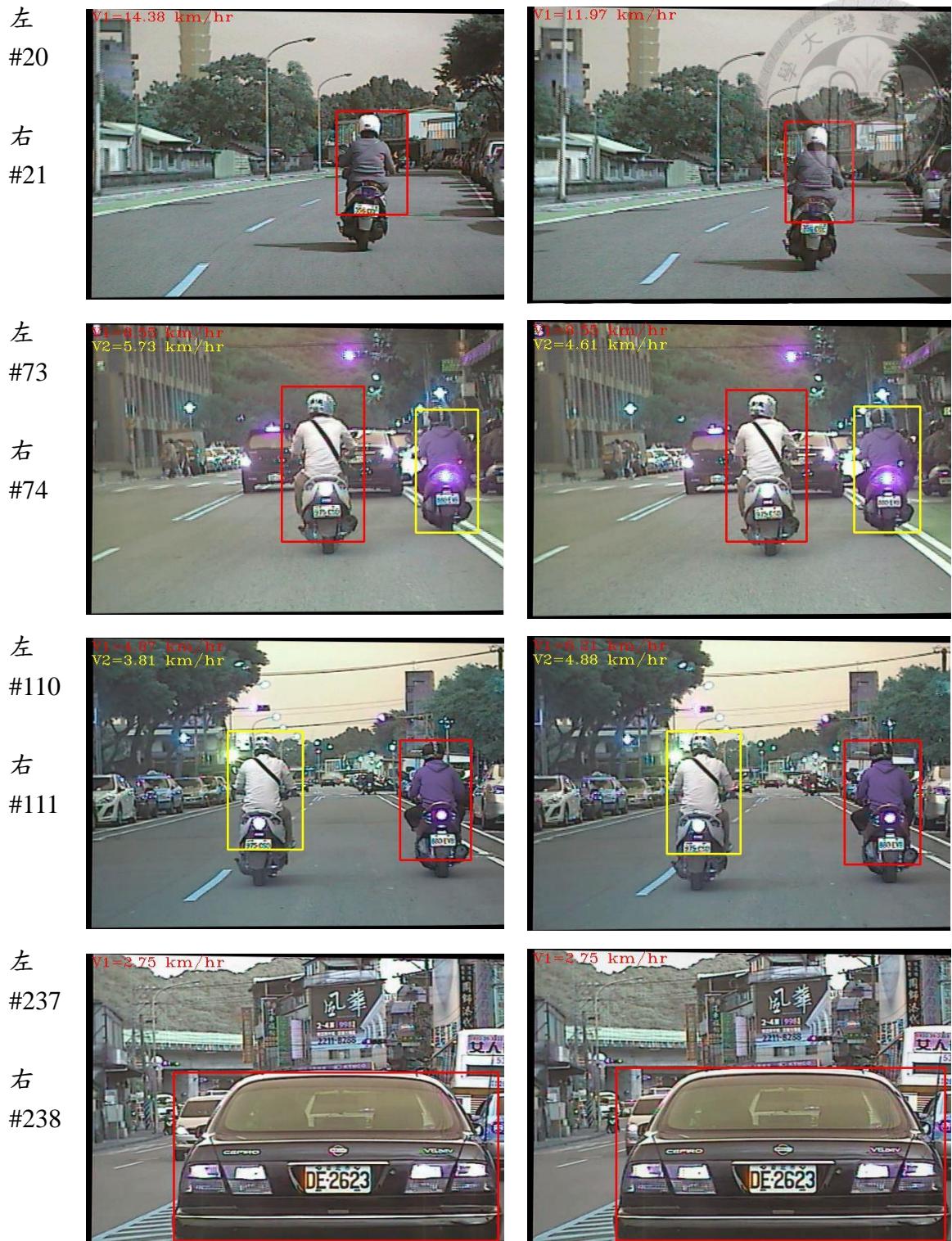


圖 4-21 立體視覺之影像序列於市區中的障礙物追蹤結果



4.3.6 橫向移動與追蹤效果之探討

前面幾節所探討的障礙物多半是針對縱向位移來進行偵測與追蹤驗證，而在現實生活中的緊急狀況多半是行人或車輛從駕駛員的左右兩側突然出現導致，因此本節將以行人為障礙物探討系統對側向移動之障礙物的追蹤效果。

在進行障礙物橫向移動實驗之前，必須先找出攝影機拍攝到之各深度的水平偵測範圍。根據式 3.1，可得到 15、13、10、7、5 公尺的水平偵測結果分別為 5.03、4.34、3.30、2.27、1.57 公尺，但實際上測得結果如下圖所示，這兩者之間的差異來自於障礙物本身的寬度，因此比理論值低是合理的。

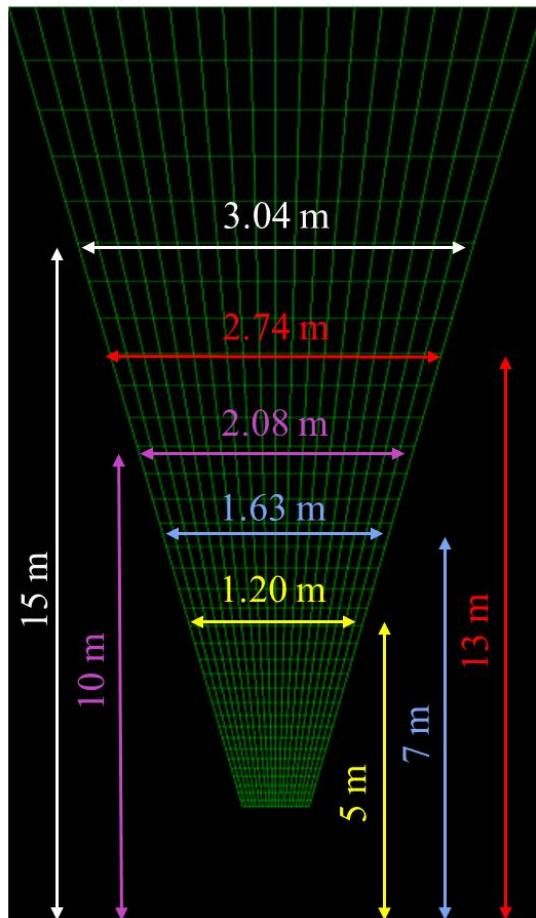


圖 4-22 立體視覺在不同距離之水平偵測範圍

經過初步測試後得到不同距離所對應到之水平偵測範圍，接著將模擬行人過馬路之情景分別在不同距離 (5、7、10、13、15 公尺) 橫向行走並追蹤，實驗場

景示意圖如下圖 4-23、統計後的實驗結果列於下表 4-8。結果顯示障礙物在距離 13 公尺以內的追蹤有較佳的表現，距離 15 公尺時則有急遽下降的趨勢，此因為障礙物偵側跳動較為明顯所致。

實驗 4



圖 4-23 障礙物橫向移動實驗示意圖

表 4-8 橫向移動於不同距離之追蹤結果

Distance of obstacle	5 m	7 m	10 m	13 m	15 m
Number of detected obstacles	231	224	586	587	255
Succeed	231	224	585	571	224
Fail	0	0	1	16	31
Success Rates	100 %	100 %	99.83 %	97.27 %	87.84 %

4.4 避障模式

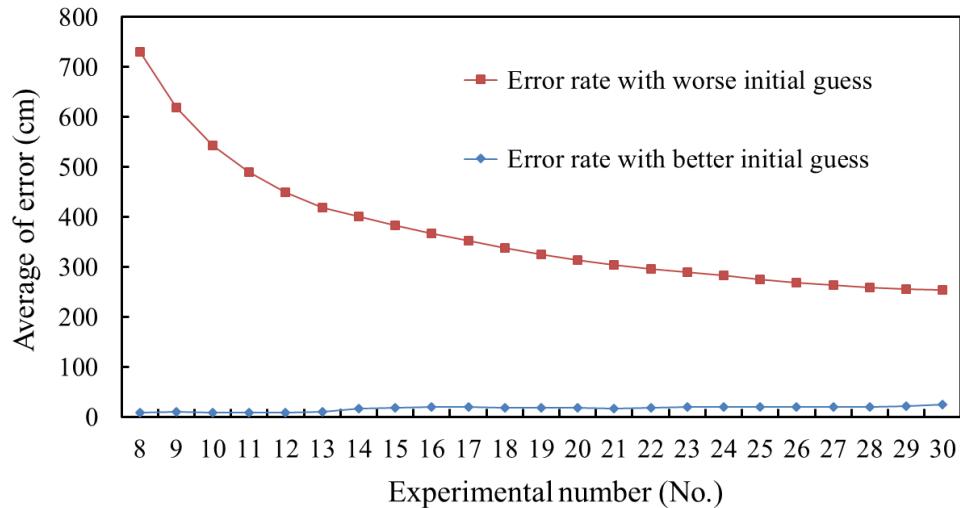
4.4.1 卡爾曼濾波器之探討

經過速度模型的建立後，我們已經約略可估算出障礙物速度了，這也表示我們可利用速度模型計算下一個時間障礙物的概略位置，所使用的卡爾曼濾波器其運動模型適合用在等速度直線運動上，在時間夠長的情況下會收斂，但是就應用上來說障礙物的位置預測能越早收斂越好，且追蹤到的障礙物因為偵測距離上的限制，所累積的 frames 數目不一定足夠多到讓系統收斂。為了解決此問題我們提出一個改進 initial guess 的方法，將利用現在已知的速度模型分別計算 x 跟 z 方向的速度大小進而得到下一個時間的位置，給予卡爾曼濾波器 initial guess 跟狀態位置的估算，如此一來便可在較短時間內收斂並得到較低的誤差，最後經過測

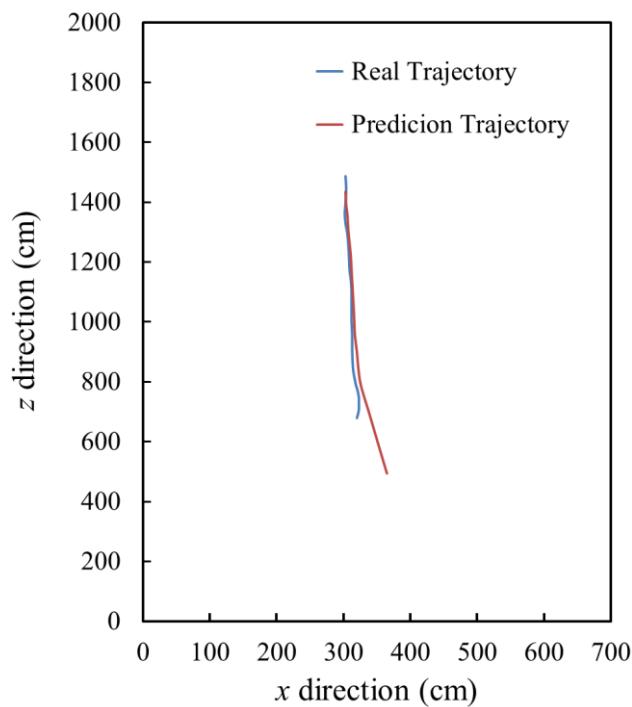
試得到的結果如下圖 4-24 其中預測位置與實際目標物位置之誤差計算方式採用 RMSE (root-mean-square error)的計算方式，如下式

$$RMSE = \sqrt{\frac{\sum_{t=1}^n [(x_{real,t} - x_{predict,t})^2 + (z_{real,t} - z_{predict,t})^2]}{n}} \quad (4.1)$$

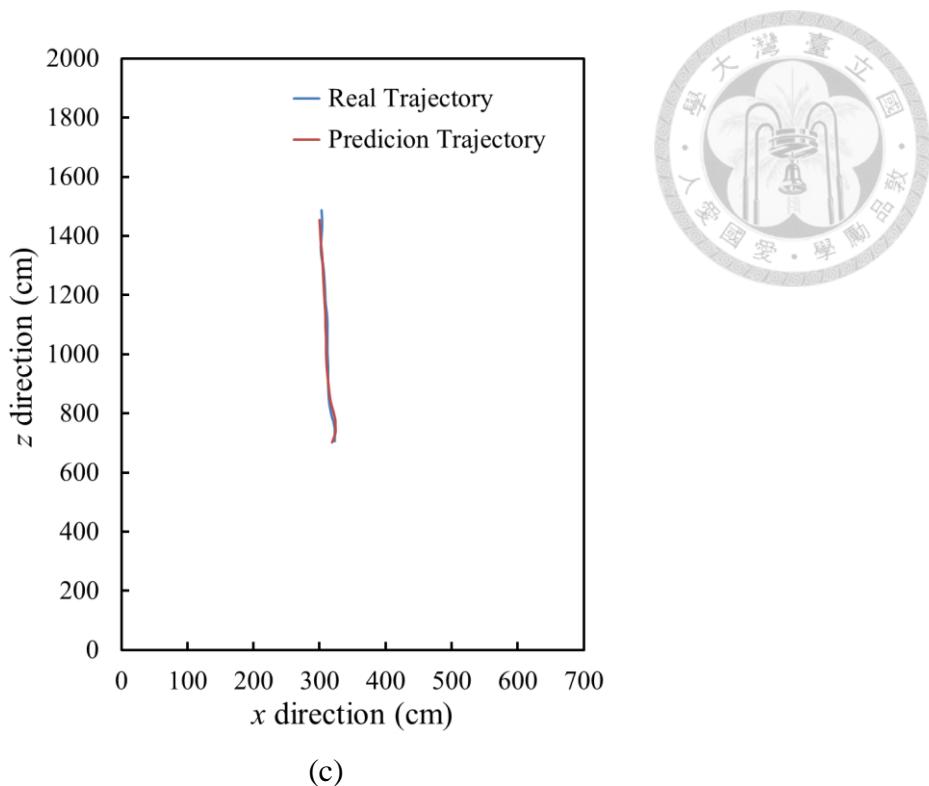
n 代表預測次數



(a)



(b)



(c)

圖 4-24 卡爾曼濾波器探討 (a) 不同 Initial guess 的誤差比較 (b) 較差的 Initial guess 之軌跡圖 (c) 較佳的 Initial guess 之軌跡圖

實驗結果如上圖 4-23 設攝影機位置在座標 (350,0) 處，我們以上視圖觀察被追蹤之障礙物移動過程，圖中藍色為雙眼立體視覺偵測到並透過像差影像計算出的障礙物座標軌跡，紅色則是預測的座標軌跡。圖 4-24 (a) 是利用速度模型輸入 initial guess 紿卡爾曼濾波器進行預測的軌跡，經過計算後預測的 RMSE 約為 25 公分，由於偵測範圍為距離雙眼立體視覺攝影機 20 公尺誤差百分比約為 1.25 %。圖 4-24 (b) 則是利用隨機產生的 initial guess 紿卡爾曼濾波器進行預測的 RMSE 約為 120 公分，誤差百分比約為 6%。比較結果後可看出雖然理論上利用隨機產生的 initial guess 進行預測最後會因為收斂而預測趨近實際的軌跡，但實際上此實驗中在尚未收斂障礙物便已經遠離偵測範圍。因此若 initial guess 紿予卡爾曼濾波器較接近的預測位置時則可以快速收斂並有較低的 RMSD，對於本系統偵測距離為 20 公尺可確保在障礙物偵測到後利用較短時間內達到較低的預測誤差有實際的成效。

探討後可知一開始的量測誤差會決定一個系統收斂與預測的能力，因此以利體視覺計算出的三維座標，加上可信任之速度估算結果決定出較佳的 initial guess 進行位置預測軌跡實驗。實驗場景於台大農業試驗場，實驗過程將以一個農人推動中耕機來進行，攝影機高度 110 公分，偵測範圍 20 公尺。實驗時系統會偵測並追蹤移動中的障礙物同時預測其移動軌跡，實驗結果如下圖 4-25 示。圖 4-25 (a) 可同時比較系統預測的軌跡與實際行走的軌跡，經過計算平均誤差約 11 公分（約 0.5 %）。

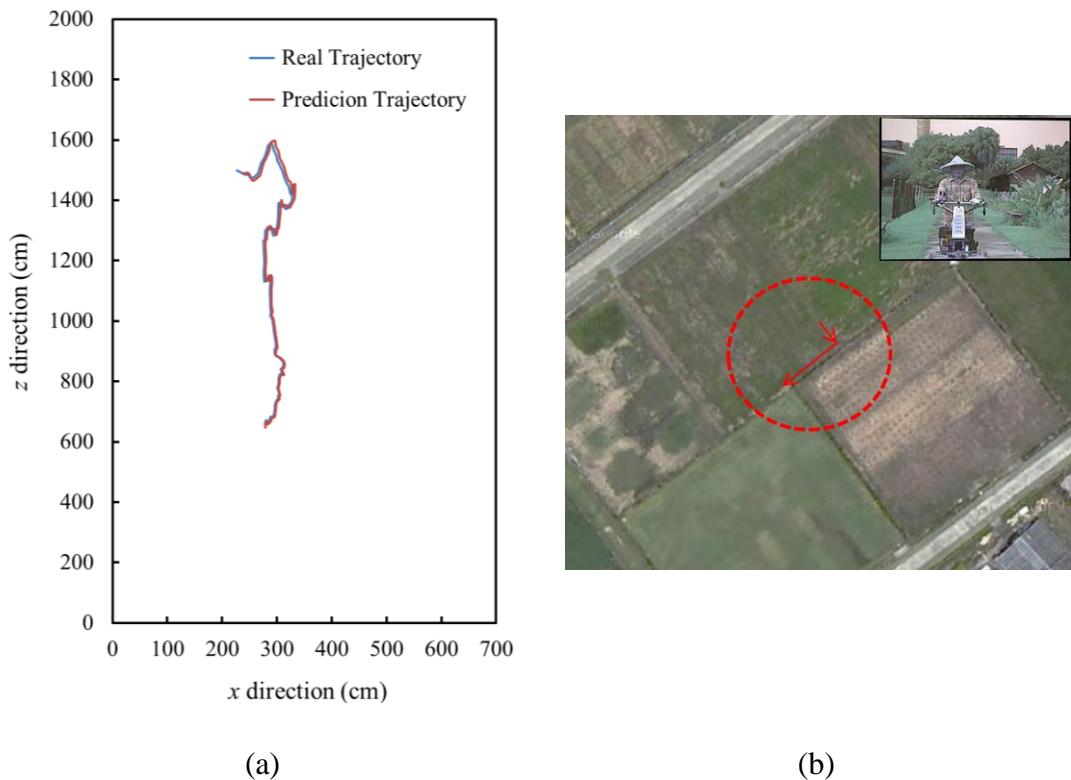


圖 4-25 軌跡預測與實際軌跡的比較

(a) 軌跡記錄於上視圖的結果，(b) 中耕機移動的路線圖。

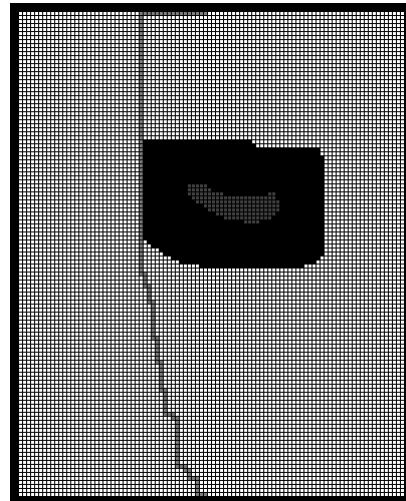
4.4.2 路徑規劃

利用權重地圖進行路徑規劃並不是每個時間都進行，配合著我們已經分成四

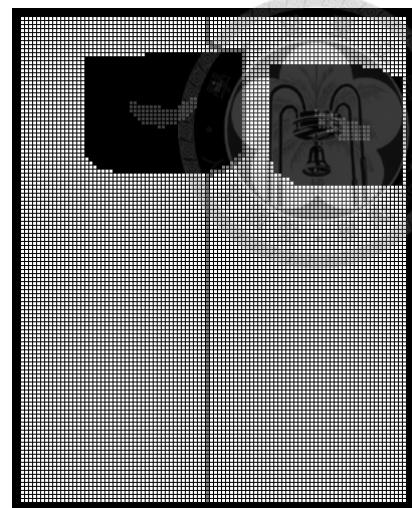
類的警示訊息，只有在「ALARM」、「ALERT」兩個狀態下才執行演算法，原因是當障礙物距離過近也就是「BRAKE」的狀態下，會建議使用者直些剎車而不建議採取閃避的策略，「CLEAR」則表示前方 20 公尺以內皆為 free space 的安全區域，因此也沒又必要做導航。

路徑規劃實驗於市區道路進行，結果如下圖 4-26 所示，期間共進行約 90 次的路徑規劃，權重地圖解析度為 100×125 ，圖中可看出團塊狀中間的深灰色代表障礙物在地圖上所顯示的原始大小，周圍環繞的黑色則為經過膨脹運算後的結果，代表著考慮車身大小後障礙物的影響範圍，淺灰色則代表路徑規劃後的結果。

30



50



80

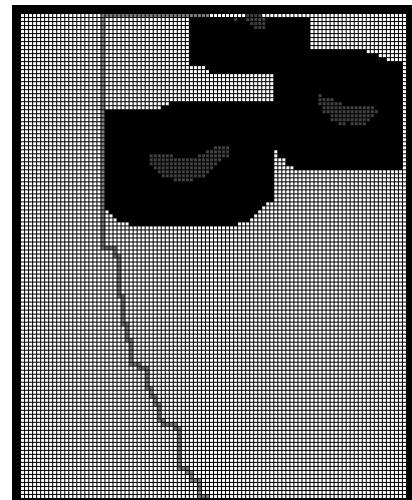


圖 4-26 左影像與路徑規劃結果

在 3.5.3 時曾提到過 A* 演算法是一個區域式搜尋的演算法，較適合用在即時系統上，因此將設計一實驗來探討地圖搜尋的範圍。實驗場景包含農場、校園以及市區道路，並將路徑規劃之地圖搜尋範圍百分比記錄下來，實驗數據共 253 筆，經統計後算出地圖平均被搜尋範圍約為 49.1 %，如下圖所示。系統執行路徑規劃時的平均速度約為 12 fps。

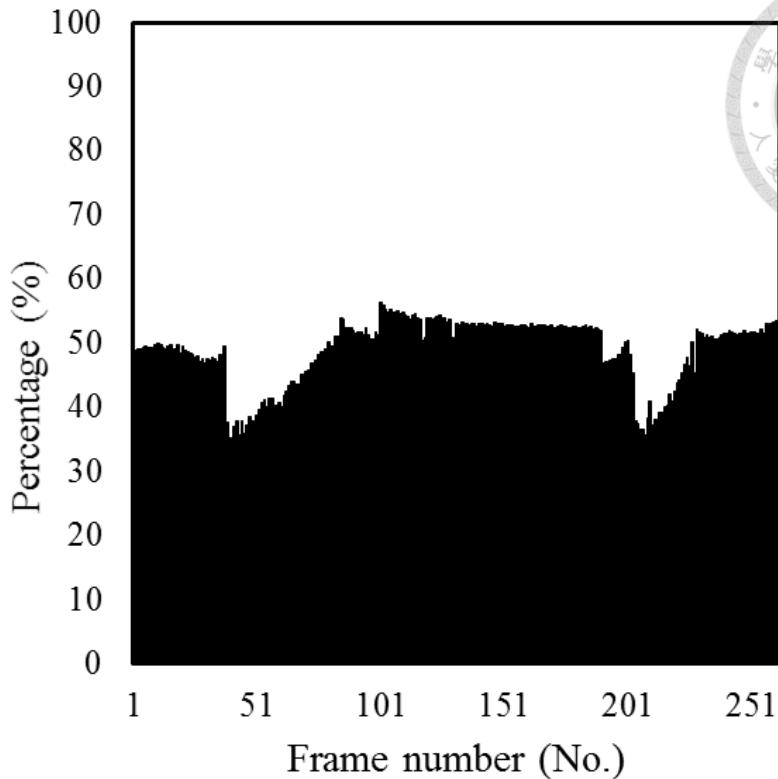
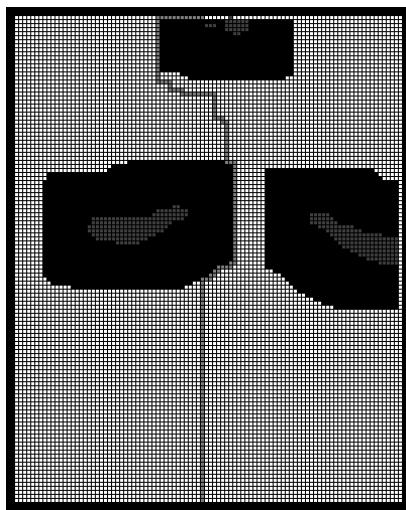


圖 4-27 路徑規劃之地圖搜尋百分比

4.4.3 避障標誌辨識結果

在 3.5.4 所提出的避障標誌辨識方法包含兩種有效特徵（總路徑長、二階導數）以及決策樹，實驗是延續上一節的路徑規劃結果來進行辨識，辨識的正確與否並無考慮到對向車道或攝影機視角以外的範圍，僅以權重地圖上障礙物分布與辨識結果來進行判斷，經過統計後得到結果如表 4-9，在總路徑長不大於 132 歸類為直行之下的辨識結果顯示「直行」的辨識效果良好，而其他類別有少數 false alarm 的情形，探究其原因多為前後兩個障礙物在前視圖影像中重疊性過高導致路徑規劃結果呈現 S 型，如此一來演算法找尋到的反曲點可能不正確，以下圖 4-28 為例，我們期待的辨識結果應為「直行」，但在整個地圖的路徑規劃中因後方有障礙物而必須繞過去造成判斷錯誤。根據提出的辨識方法進行標誌判斷可得辨識率約 91.3 %。



(a)



(b)

圖 4-28 避障標誌辨識錯誤的例子 (a) 路徑規劃結果 (b) 辨識結果

表 4-9 避障標誌辨識結果

Recognized sign	Straight	Turn right	Turn left	Total	Recognition rate
Straight	89	1	0	90	98.8 %
Actual sign	Turn right	0	66	10	76
	Turn left	0	11	76	87.4 %
Average recognition rate					91.3 %

4.4.4 導航與警示系統

根據本研究所提出的方法與結果，透過抬頭顯示器的概念結合了障礙物偵測、追蹤、速度估算、動態行為以及最後的路徑規劃結果做出一個顯示介面來進行避障導引，並給予車輛駕駛人警示訊息以協助做出正確的行車判斷。顯示介面如下圖 4-29，圖中的同種顏色皆為同一個障礙物所擁有的資訊，包含障礙物為何，狀態為逼近或遠離以及相對速度大小是多少，以該圖黃色框框的機車騎士

為例，目前相對速度約為 1.67 km/hr，狀態為遠離。根據上視圖可看出紅色框框的機車騎士是距離架設立體視覺系統的車輛最近，警示訊息為「ALERT」代表最近的車輛距離駕駛的車輛約 15 m ~ 20 m 之間。

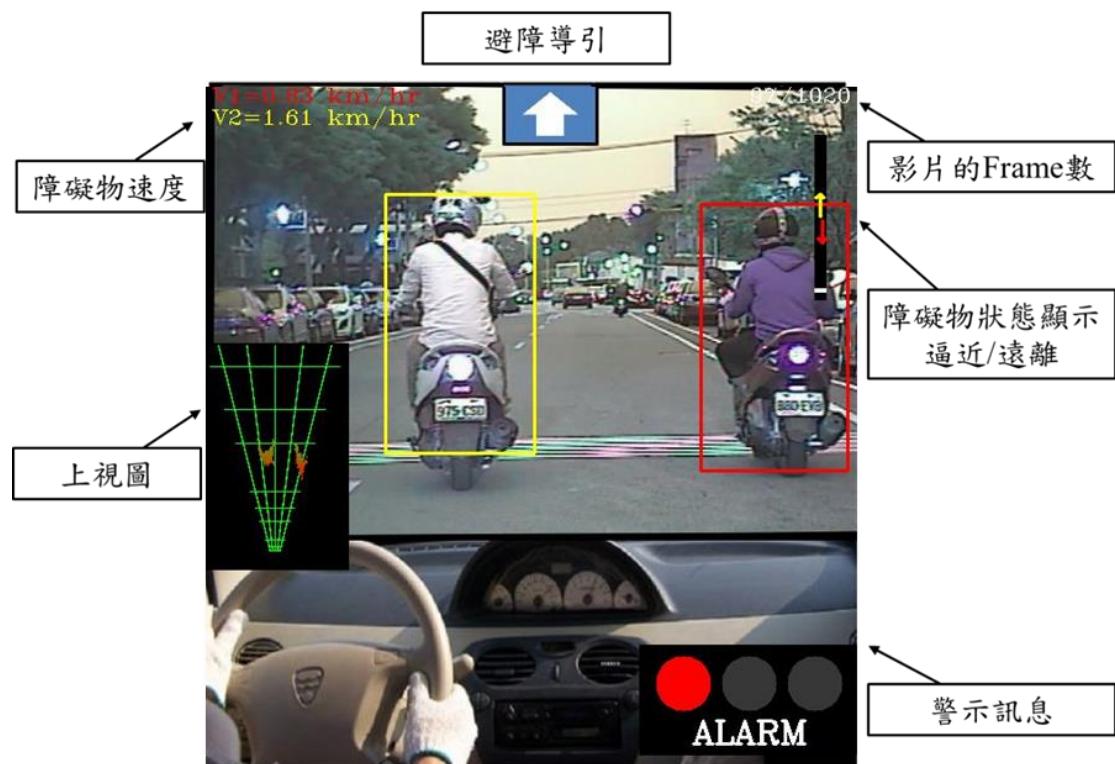


圖 4-29 避障與導航顯示介面

第五章 結論與建議



5.1 結論

本研究成功建立一套雙眼立體視覺系統。此系統利用兩顆 CCD 攝影機為核心，同時具備色彩影像與深度資訊的優點，可實際應用於校園，農場，市區道路上之障礙物避障與導航，並兼具成本低廉的優勢。根據所提出之方法與實驗結果，可整理出以下幾點結論：

1. 加裝鏡頭固定架

使用鏡頭固定架除了可增加系統實車使用時的穩定性外，並可減少每次實驗皆須進行攝影機校正的程序，因此僅需在使用第一次時校正完後並在每次架設立體視覺於車輛時手動調整使影像中心所指的高度與攝影機高度相同即可。

2. 立體視覺誤差

立體視覺在計算每個像素的深度後必須了解與實際的深度誤差，經過實驗後得知在偵測範圍為 20 公尺之內的深度計算誤差均低於 4%，因此證明本系的三維資訊是可信賴的。

3. 障礙物偵測方法

使用 SGBM 圖像匹配演算法計算像差影像的結果具豐富的深度資訊，再投影至上視圖進行障礙物偵測，由於解析度從 640×480 降低至 100×125 ，因此可有效減少系統運算時間。

4. 障礙物追蹤方法

根據本文提出的巴氏距離已成功並有效結合距離與色彩資訊並利用交叉比對法進行匹配，更提出背景的消除能有效克服環境。然而，障礙物的速度會影像系統偵測的穩定度，便發展適應性巴氏距離使追蹤效果更穩定。系統運

算之平均追蹤速度約 11 fps。



5. 速度估測

為了使適應性巴氏距離能有效的發揮作用，已成功建立系統對障礙物估測的相對移動速度與實際移動之相對速度的模型，速度估算經過驗證後誤差約為 10 %。

6. 卡爾曼濾波器

卡爾曼濾波器能對所偵測到的障礙物進行下一個時間點的位置預測，initial guess 如果給的好則較快收斂，因此以障礙物速度的計算估測下個時間的位置作為 initial guess 的好處除了能較正確預測位置外，並且針對障礙物追蹤時障礙物已超出偵測範圍仍還未收斂的這個問題能有效解決，且經過計算後得到偵測範圍為 20 公尺之內誤差約 1.25 %。

7. 路徑規劃與導航標誌辨識

本研究成功發展 A* 路徑規劃演算法，地圖搜尋為區域性，地圖搜尋百分比約 49 %，且運算時間平均約 11 fps，故可用於即時運算。根據路徑規劃結果利用路徑總長度與二階導數已能成功將其分成三個賦予標誌的類別。

8. 視覺化使用者介面

設計出具視覺效果之使用者介面如圖 4-20，將前方資訊以上視圖呈現於介面。介面中還提供的資訊包含以不同顏色框選追蹤之障礙物的位置以及其速度大小，遠離或逼近，並有警訊訊息以及避障建議供使用者參考。

9. 立體視覺於車輛與農業之應用

本文所進行的道路與市區測試實驗皆實際安裝系統於車艙進行；農場實驗則是以可移動式的載具於田野中測試，此外障礙物類型皆符合實際之應用，並經過驗證可實際應用於該場景中。



5.2 建議

雖然系統的開發已具有一定的穩定性，但仍有一些空間可進行改善，以下用列舉的方式說明：

1. 未來若想針對側向移動之障礙物進行追蹤，可考慮更換視角較廣的攝影機，而其他距離感測器也可結合使用使距離估測的準確度更高。
2. 障礙物的動態軌跡是利用線性模型進行預測，對於障礙物的運動方式有較大限制，未來若發展非線性運動模型將能有機會適應環境的突發狀況。

參考文獻



徐嘉鴻。2011。大尺度虛擬實境場景接合與修補演算法之研究。碩士論文。臺北：國立臺灣大學生物產業機電工程所。

賴宗誠。2012。應用多組雙眼攝影機系統進行車前三維環境模型重建。碩士論文。臺北：國立臺灣大學生物產業機電工程所。

Andrews, J. R. and N. Hogan. 1983. Impedance control as a framework for implementing obstacle avoidance in a manipulator. *Control of Manufacturing Processes and Robotic Systems*. 243-251.

Barth, A. and U. Frank. 2009. Estimating the driving state of oncoming vehicles from a moving platform using stereo vision. *IEEE Transactions on Intelligent Transportation Systems*. 10(4): 560-571.

Barth, A. and U. Frank. 2010. Tracking oncoming and turning vehicles at intersections. *International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 861-868.

Bay, H., T. Tuytelaars and L. Van Gool. 2006. Surf: Speeded up robust features. In 9th European Conference on Computer Vision. 404-417

Boykov, Y. Y. and M. P. Jolly. 2001. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. *Proceedings of International Conference on Computer Vision*. 105-112.

Bradski, G. and A. Kaehler. 2008. *Learning OpenCV: Computer vision with the OpenCV library*. America: O'Reilly Media.

Billingsley, J. 2005. Vision Application in Agriculture. *The 9th International Conference on Mechatronics Technology*. 1-6.

Chang, C. C. and C. J. Lin, LIBSVM : a library for support vector machines, 2001.

Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

- Chapron, M. 1993. Visualization of Corn Acquired From Stereovision. Conference Proceedings of Systems Engineering in the Service of Humans. 334-338.
- Cohn, J. F., A. J. Zlochower, J. J. Lien and T. Kanade. 1998. Feature-Point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expression. International Conference on Face and Gesture Recognition. 396-399.
- Comaniciu, D. and P. Meer. 2002. Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24 (5): 603–619.
- Diaz Alonso, J., E. Ros Vidal, A. Rotter and M. Muhlenberg. 2008. Lane-change decision aid system based on motion-driven vehicle tracking. IEEE Transactions on Vehicular Technology. 57(5): 2736-2746.
- Dijkstra, E. W. 1959. A note on two problems in connection with graphs. Numerische Mathematik 1: 269-271.
- Elfes, A. 1987. Sonar-Based Real-World Mapping and Navigation. IEEE Journal of Robotics and Automation. 3(3): 249-265.
- Ferguson, D., M. Likhachev and A. Stentz. 2005. A guide to heuristic-based path planning. Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS). 9-18.
- Fischler, M. A. and R. C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM. 24(6): 381-395.
- Franke, U., C. Rabe, H. Badino and S. Gehrig. 2005. 6d-vision: Fusion of stereo and motion for robust environment perception. Springer Berlin Heidelberg In Pattern Recognition. 216-223.

Freund, Y. and R. E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. *Computational learning theory*. Springer Berlin Heidelberg. 23-37.

Garcia, F., P. Cerri, A. Broggi, J. M. Armindo and A. D. Escalera. 2009. Vehicle Detection Based on Laser Radar. *EUROCAST*. 391-397.

Guivant, J., E. Nebot and S. Baiker. 2000. Autonomous Navigation and Map building Using Laser Range Sensors in Outdoor Applications. *Journal of Robotics Systems*. 17(10): 565-583.

Harris, C. and M. Stephens. 1988. A Combined Corner and Edge Detector. *Proceedings of the 4th Alvey Vision Conference*. 147-151

Hartley, I. R. 1997. Self-calibration of stationary cameras. *International Journal of Computer Vision*. 22(1): 5-23.

Hirschmuller, H. 2008. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 30(2): 328-341.

Hwang, Y. K. and N. Ahuja. 1992. Gross motion planning: A survey. *Computing Surveys*. 24(3): 219-291.

Jian, G., L. Li and W. Chen. 1998. Fast recursive algorithms for two-dimensional thresholding. *Pattern Recognition*. 31(3): 295-300.

Kasper, D., G. Weidl, T. Dang, G. Breuel, A. Tamke and W. Rosenstiel. 2011. Object-oriented Bayesian networks for detection of lane change maneuvers. *IEEE Intelligent Vehicles Symposium (IV)*. 673-678.

Kise, M., Q. Zhang and F. Rovira Más. 2005. A stereovision-based crop row detection method for tractor-automated guidance. *Biosystems Engineering*. 90(4): 357-367.

Liu, W., C. Song, X. Z. Wen, H. Yuan and H. Zhao. 2007. A monocular-vision rear

vehicle detection algorithm. IEEE International Conference on Vehicular Electronics and Safety. 1-6.

Lowe, D., G. 2004. Distinctive image features from scale-invariant keypoints. *Journal of Computer Vision*. 60(2): 91-110.

Lv X. L., X. R. Lv and B. F. Lu. 2011. Identification and Location of Picking Tomatoes Based on Machine Vision. IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA). 101-107.

Mockel, S., F. Scherer and P. F. Schuster. 2003. Multi-Sensor Obstacle Detection on Railway Tracks. *Proceedings IEEE Intelligent Vehicles Symposium*. 42-46.

Ollis, M. and A. Stentz. 1996. First results in vision-based crop line tracking. *Proc. IEEE International Conference on Robotic and Automation*. 951-956.

Pilarski, T., M. Happold, H. Pangels. M. Ollis, K. Fitzpatrick and A. Stentz. 2002. The Demeter System for Automated Harvesting. *Proceedings of the 8th International Topical Meeting on Robotics and Remote Systems*.

Pocoll, C., S. Nedevschi and M. M. Meinecke. 2008. Obstacle Detection Based on Dense Stereovision for Urban ACC Systems. In *Proceedings of the 5th International Workshop on Intelligent Transportation*. 13-18.

Rovira-Más, F., Q. Zhang, J. F. Reid and J. D. Will. 2005. Hough-transform-based vision algorithm for crop row detection of an automated agricultural vehicle. *Proceedings of the Institution of Mechanical Engineers*. 219(8): 999-1010.

Rovira-Más, F., Q. Zhang and J. F. Reid. 2004. Automated agricultural equipment navigation using stereo disparity images.

Rowley, H. A., S. Baluja and T. Kanade. 1998. Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine intelligence*. 20(1): 23-38.

Safavian, S. R. and D. Landgrebe. A survey of decision tree classifier methodology.

IEEE Transactions on Systems, Man and Cybernetics. 21(3): 660-674.

Schofield, C. P., J. A. Marchant, R. P. White, N. Brandl and M. Wilson. 1999.

Monitoring Pig Growth using a Prototype Imaging System. Journal of Agricultural Engineering Research. 72: 205-210.

Sivaraman, S. and M. M. Trivedi. 2013. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. IEEE Transactions on Intelligent Transportation Systems. 1-23.

Sivaraman, S., B. T. Morris and M. M. Trivedi. 2011. Learning multi-lane trajectories using vehicle-based vision. IEEE International Conference on Computer Vision Workshops (ICCV Workshops). 2070-2076.

Sturm, P. F. and S. J. Maybank. 1999. On plane-based camera calibration: A general algorithm, singularities, applications. In Conference on Computer Vision and Pattern Recognition. 432-437.

Sun, Z., G. Bebis and R. Miller. 2006. Monocular precrash vehicle detection: features and classifiers. IEEE Transactions on Image Processing. 15(7): 2019-2034.

Sural, S., G. Qian and S. Pramanik. 2002. Segmentation and histogram generation using the HSV color space for image retrieval. International Conference on Image Processing. 589-592.

Tilneac, M., V. Dolga, S. Grigorescu and M. A. Bitea. 2012. 3D Stereo Vision Measurements for Weed-Crop Discrimination. Electronics and Electrical Engineering. 123(7): 9-12.

Trucco., E. and A. Verri. 1998. Introductory techniques for 3-D computer vision. New Jersey: Prentice Hall.

Tsai, R. Y. 1987. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. IEEE Journal of Robotics and Automation. 3(4): 323-343.

Welch, G. and G. Bishop. 1995. An introduction to the Kalman filter. University of North Carolina at Chapel Hill.

Wender, S. and K. Dietmayer. 2008. 3D vehicle detection using a laser scanner and a video camera. IET Intelligent Transport Systems. 2(2): 105-112.

Yang, J., J. Y. Yang, D. Zhang, and J. F. Lu. 2003. Feature fusion: parallel strategy vs. serial strategy. Pattern Recognition. 36(6): 1369-1381.

Yao, J., C. Lin, A. J. Wang and C. C. Hung. 2010. Path planning for virtual human motion using improved A* star algorithm. Proceedings of the IEEE International Conference on Information Technology. 1154-1158.

Yilmaz, A., O. Javed and M. Shah. 2006. Object Tracking: A Survey. ACM Computing Surveys. 38(45): 1-45.

Zhang, Z. 1999. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence. 22(11): 1330-1334.

Zhu, C. 2011. Video Object Tracking using SIFT and Mean Shift. Master Thesis. Gothenburg: Chalmers University of Technology, Department of Signals and Systems.