

國立交通大學

資訊科學與工程研究所

碩士論文

倒車輔助系統之影像分析與合成

Images Analysis and Synthesis of a Parking

Assistance System

研究生：吳佳昱

指導教授：莊仁輝 教授

中華民國九十六年六月

倒車輔助系統之影像分析與合成

Images Analysis and Synthesis of a Parking Assistance System

研究生：吳佳昱

Student : Chia-Yuh Wu

指導教授：莊仁輝

Advisor : Jen-Hui Chuang

國立交通大學

資訊科學與工程研究所



Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

倒車輔助系統之影像分析與合成

學生：吳佳昱

指導教授：莊仁輝 博士

國立交通大學

資訊科學與工程研究所

摘要

本論文中實作了一個即時的提供視訊畫面之倒車輔助系統，不同於其他相關的研究中使用一些特殊的裝置或設備擷取汽車額外資訊，我們沒有使用到任何的特定感應器或設備，而只使用一台裝在車後的攝影機拍攝車後影像，再利用平面投影轉換將影像校正成一個俯視畫面，使駕駛者可以直接由畫面中了解汽車與停車格間的關係。並配合特徵點的擷取與追蹤，將過去影像和目前影像進行合成接圖，以提供駕駛者一個完整且清楚的環境俯視圖，讓駕駛者在倒車停車時可以更為快速且容易。

Images Analysis and Synthesis of a Parking Assistance System

Student : Chia-Yuh Wu

Advisor : Dr. Jen-Hui Chuang

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

The logo of National Chiao Tung University is a circular emblem. It features a gear-like outer border. Inside the circle, there is a stylized building or structure. At the bottom of the emblem, the year '1896' is inscribed.

In this thesis, a real-time parking assistance system, which provides a top-view video for a driver to refer to, is investigated. Unlike other parking assistance systems that use special equipments or multiple sensors, the system only uses a rear-view camera to capture scene images behind a backing car. The rear-view images are then converted to top-view ones of the vehicle and its surrounding area using homography transformations. Thus, the driver can easily observe relative positions between the vehicle and the parking area. Moreover, a sequence of top-view images can be further synthesized to generate a single, size-extended one by feature point tracking and image registration. Hence, the system is able to display surrounding area outside the current camera view. By using the proposed system, the driver can have a more clear sense of distance and direction for easier and faster parking in a garage or on the street.

致謝

本論文得以完成，首先必須感謝我的指導教授莊仁輝老師的耐心教導，老師熱心的指導讓我在研究方面事半功倍；老師嚴謹的研究態度，讓我在研究跟生活上的態度上皆有所啟發；老師開明的個性，一直包容著我，讓我可以順利且快樂的渡過研究所的生涯。此外，更要感謝三位口試委員，陳玲慧教授、雷欽隆教授以及賴飛熊教授的意見及指教，讓本篇論文的內容得以更加充實且完善，在此衷心感謝他們。

我還要感謝智慧型系統實驗室的所有成員們，感謝高肇宏學長跟林泓宏學長在研究上的指導與幫助；感謝所有碩二的戰友們，因為有你們陪伴在我身邊，大家一起努力寫論文、拼畢業，一起相互扶持，真是一段寶貴的回憶；感謝學弟妹們的鼓勵與打氣，讓我在實驗室裡充滿了歡樂，生活中不再孤單；還要感謝從大學就一起跟我奮鬥到現在的資科 94 同學們，因為有你們，我在交大的生活才能如此多采多姿。兩年的碩士生活裡，因為有你們的陪伴，讓我過得很充實也很愉快，真的非常謝謝你們。

最後，感謝一直以來支持我、為我操心的家人，以及所有幫助過我的朋友們，僅以此篇表達我誠摯的謝意。

目錄

第一章 簡介.....	1
1.1 利用後視影像建立車子周圍的 3D 場景表示法.....	2
1.2 行徑軌跡指示之倒車輔助系統.....	3
1.3 提供俯視圖畫面(Top-view)的倒車輔助系統.....	4
1.4 系統流程.....	5
1.5 本論文各章節介紹.....	7
第二章 偏斜影像校正.....	8
2.1 齊次座標系表示法.....	8
2.2 平面投影轉換.....	9
2.3 投影轉換矩陣.....	9
2.4 使用平面投影轉換校正影像.....	12
2.5 校正影像的縮放功能.....	14
2.6 總結.....	17
第三章 影像的合成—環境接圖.....	18
3.1 邊線特徵點的偵測.....	19
3.2 擷取角點特徵點.....	21
3.2.1 Hough 轉換.....	21
3.2.2 掃瞄線配合 PCA 方法.....	22
3.2.3 擷取角點特徵點.....	32
3.3 連續影像中的角點特徵點追蹤.....	33
3.4 影像的合成.....	34
3.5 總結.....	42
第四章 其它擷取特徵點的方法.....	43
4.1 Harris 角點特徵點擷取演算法.....	43
4.2 利用 Harris 演算法擷取角點的實驗結果.....	46
4.3 總結.....	49
第五章 結論與未來工作.....	50

圖目錄

圖 1-1	實驗場景示意圖。	6
圖 1-2	簡略系統流程圖。	6
圖 2-1	利用投影轉換矩陣做平面間點對點的轉換。	10
圖 2-2	車後偏斜影像利用投影轉換矩陣校正圖。	13
圖 2-3	影像轉正後未作補點動作的結果，圖 (a) 為原始影像，圖 (b) 為原始影像經投影轉換矩陣轉正後的結果。	13
圖 2-4	影像轉正後利用最近相鄰內插法作補點動作後的結果，圖 (a) 為原始影像，圖 (b) 為原始影像經投影轉換矩陣轉正後補點完的結果。	14
圖 2-5	將校正影像縮放的方法。	15
圖 2-6	校正影像的縮放功能。	16
圖 3-1	影像合成的簡略流程圖。	19
圖 3-2	Sobel 遮罩的運算子：(a) Sobel 遮罩的 X 運算子，(b) Sobel 遮罩的 Y 運算子。	20
圖 3-3	原始影像使用 Sobel 遮罩後的結果，(a) 攝影機拍攝到的原始影像，(b) 原始影像使用 Sobel 遮罩得到的結果。	20
圖 3-4	利用 Hough 轉換找出停車標記的實驗結果。	22
圖 3-5	將影像分別從左上角與右上角為起點，做水平掃瞄及垂直掃瞄的圖示。	23
圖 3-6	由左上角開始水平掃瞄的結果示意圖。	24
圖 3-7	水平掃瞄的過濾雜點機制之簡單示意圖。	25
圖 3-8	水平掃瞄到的邊界點之分布情況。	25
圖 3-9	不同角度停車標記由左上角開始水平掃瞄和垂直掃瞄的結果。(a) 對影像中的停車標記做水平和垂直掃瞄後，最後各得到一條邊的結果，(b) 對影像中的停車標記使用水平和垂直掃瞄後，最後各得到二條邊的結果。	27
圖 3-10	篩選邊流程圖。	29
圖 3-11	選出最後二群線上點之流程圖。	30
圖 3-12	利用 PCA 得出 Principal Components 之示意圖。	32
圖 3-13	擷取角點特徵點的簡略流程圖。	33
圖 3-14	影像合成預期結果示意圖。	34
圖 3-15	找出角點校正後的座標值。	35
圖 3-16	合成方法簡略示意圖。	35
圖 3-17	合成影像過程之示意圖。	37
圖 3-18	擴大影像合成之實例。(a) ~ (f) 分別為實驗結果 video 中第 1、20、40、60、80、100 和 120 張 frame。	39
圖 4-1	矩形區域在影像中移動示意圖。(a) 為矩形區域在灰階變化平坦的影像	

區域中移動，(b) 為矩形區域在邊或線的影像區域中移動，(c) 為矩形區域在具有角點特徵點的影像區域中移動。.....44

圖 4-2 特徵值 α 、 β 與影像灰階變化關係圖。.....46

圖 4-3 使用 Harris 特徵點擷取演算法，特徵值超過臨界值 2000 的影像點。 .47

圖 4-4 主要角點篩選方法示意圖。.....48

圖 4-5 使用 5×5 block 做區域最大值篩選。.....48



第一章 簡介

在今日社會中，多數人都以汽車當作主要的交通工具，然而對許多駕駛者而言，停車是一件麻煩的事。汽車的倒車停車須倚賴駕駛者個人的技術和經驗，但即使是開車技術純熟的駕駛者也會有為了停好車，將車完全開進停車格而花上不少時間的麻煩。而且在停車倒車時常會擦撞到停在旁邊的車輛，駕駛者在專心停車時會忽略周圍的情況，因而發生一些意外悲劇。近年來，隨著科技的發展與交通意外的失事率增加，人們對於行車安全的需求日益重視，因此許多有關於汽車安全的輔助系統陸續被研發出來，而且也逐漸成為了新車的標準安全配備之一。其中，倒車輔助系統就是一項汽車安全系統中備受矚目的技術。

目前市面上最常見的倒車輔助系統分成二種，一種是使用可以認知環境的感應系統，如倒車雷達。其作法為在車子的後保險桿上裝一系列超音波感應器，當車子接近後方障礙物時，感應器會偵測到後方障礙物並發出聲響來提醒駕駛。另外一種是在車子的後方裝設一台後視的攝影機（rear-view camera），將車子後方的實際影像傳送到車內的小型螢幕顯示給駕駛者參考。配合上視覺化的畫面，讓駕駛者可以看到車子後方的環境，進而判斷車子跟停車格的位置及相對關係。利用這種方式，相信可以避免很多在停車時所遭遇到的擦撞事故。在缺乏視訊的畫面之情況下，駕駛者僅能靠著眼睛及經驗判斷，而有時更會因遮蔽物以及視角的關係而看不到車子後方的情況，若能以有提供視訊畫面的倒車系統輔佐之，不僅能看到車子與停車格的相對關係，甚至可以看到地面上有無障礙物或地面是否有傾斜的情況，必定對於駕駛者停車有很大的助益，也對駕駛者的安全有了更大的保障。

本篇論文所提出的倒車輔助系統就是利用在倒車時配合視訊畫面的構想，使用一台固定在車子後方的後視攝影機拍攝車後方的環境，讓駕駛者可以在倒車時觀察到車子與車子後方環境關係，並將所拍攝的車後環境的影像畫面轉成擬真的鉛直俯視角度的影像畫面，使車子及停車格的關係能夠以一個校正的畫面來呈

現，並提供給駕駛者觀察，讓駕駛者可以直接且清楚的看到目前車子及車後環境的相對位置關係，以增加倒車停車的便利性及安全性。此外，如果可以再利用特徵點的追蹤，將過去攝影機所拍攝到的影像和現在的影像合成，進行畫面的接圖，將讓使用者可以參考到較完整的場景環境圖，而不會受限於目前攝影機較小的視角範圍而造成情況誤判。在以下的小節裡，我們將會介紹一些同樣利用攝影機來拍攝車子及車子周圍環境，並提供視訊畫面以供參考的倒車輔助系統，以及其他利用拍攝到的影像來發展的相關應用系統。

1.1 利用後視影像建立車子周圍的 3D 場景表示法

使用後視攝影機為基礎架構的倒車輔助系統，雖然提供駕駛者有關視覺方面的輔助，但因為只在螢幕上顯示攝影機直接拍攝到的車後環境偏斜影像，駕駛者在參考時可能會因不確定車子跟車後環境較完整的相對關係或是因攝影機裝設的角度不妥而導致無法拍攝到的所需場景資訊等問題，而無法正確且迅速的倒車停車。因此，有許多系統都會將後視影像作一些更進階的處理，讓駕駛者可以更清楚更容易地根據所提供的影像畫面來倒車停車。

Fintzel et al.[1][2]實作了一個使用 3D 場景表示法的倒車輔助系統。這個系統是將後視攝影機拍攝到的車後環境影像經過一些影像處理後轉成 3D 的場景表示，希望透過 3D 的場景表示法可以將車子經過的整體環境都建構出來，讓駕駛者可以根據整體停車環境的 3D 架構更清楚了解車子周圍障礙物的位置以及車子和環境之間的位置關係。這個系統主要的輸入是利用架設在車後的一台後視 CCD camera、裝設在汽車左右後輪上的 ABS sensors 和 reverse gear 的資訊，將這些資訊整合後利用汽車運動軌跡方程式建立起對應的汽車行徑測距法 (odometry) 來估算出相機的外部校正參數，接著再重建出影像中的障礙物的點。在重建的過程中，首先先利用[3]的方法偵測到影像中的特徵點，再將拍攝到的影像分成不同解析度，分別對其做特徵點偵測，利用較差解析度的影像找到的特

徵點，將較細碎的路面上的雜點刪除，使找到的特徵點可以集中在環境中的較大障礙物上（如停放在路邊的汽車）。然後以 optical flow 的方式對影像中的特徵點進行追蹤(tracking)，最後使用立體視覺的原則(principle of stereo vision)重建出影像中的特徵點，並配合一些過濾雜點的機制，將一些不屬於場景中的錯誤 3D 重建點刪除，確保重建出來的 3D 影像點的正确性。之後，為了增加 3D 場景表示的可讀性，再將重建出的 3D 點建立其網狀結構(mesh)並貼圖。但是利用 3D 場景表示法是否真的可以提供駕駛者更容易辨視環境及障礙物呢？這個問題的答案是無法確定的，因為這個系統最後重建出的障礙物的點並不清晰，而且對於駕駛者而言，人工所建立出的 3D 場景，其可讀性與可理解性也許並不會比真實影像場景來得更好，所以提供 3D 場景表示法的倒車輔助系統也許不是那麼適用於現實生活中。



1.2 行徑軌跡指示之倒車輔助系統

除了提供視覺化輔助畫面外，現今有些倒車輔助系統還融入了智慧化的倒車導引指示輔助功能。即除了車後環境畫面外，還在車內螢幕上顯示駕駛者倒車的行徑路線及操作參考的指示線，以供駕駛者判斷參考，來提高倒車的便利性和安全性。如日本愛信精機(Aisin Seiki)所開發的停車輔助後視儀(parking assist rear view camera)[4]，以及吳瑞鴻及劉景富在[5]所提出的倒車輔助系統，皆具備這項功能。這種系統主要是當駕駛者排入倒車檔時，系統就會啓動倒車後視攝影機及方向盤轉角感測器，擷取出車後即時影像畫面和汽車方向盤轉動的角度訊號。而後，利用方向盤轉動的角度和車子本身內外部的資訊（包括車輛的後輪距離、前軸和後軸中心點的距離、前軸中心點的速度，車輛中心軸與水平方向的夾角，車子前軸和後軸中心點的位置，...等資訊），套用在汽車運動模型上，推導出車輛倒車之運動軌跡方程式[6]，就可以事先估算出汽車倒車的路徑軌跡以及相對距離的參考線。接著將這些軌跡資訊透過座標軸的轉換、透視投影與廣角鏡頭畫面

的扭曲現象，產生符合攝影機所拍攝到的倒車軌跡輔助指示線後，再將倒車軌跡輔助指示線和車後影像疊合處理，顯示在車內的螢幕中。最後，再加入距離參考線和車身延伸線來提供駕駛者判斷車輛方向位置和相對距離等重要資訊，讓倒車停車的動作更容易也更快速。這種方法雖然提供了倒車路徑讓駕駛者參考，但是對駕駛者而言，他們實際上如何依據畫面所顯示的輔助倒車路徑去倒車？是否可以承擔倒車系統規劃的路徑發生錯誤的風險？這些都是可能存在的問題。

1.3 提供俯視圖畫面(Top-view)的倒車輔助系統

目前倒車輔助系統多是將後視攝影機拍攝到的車後環境影像直接顯示在車內幕以供駕駛者參考，對於駕駛者而言，其實較難直接根據畫面上顯示之車後環境影像來判斷出汽車和停車格的關係。因此 DENSO 公司就發展出將拍攝到的影像轉成用俯視角度呈現的影像畫面，以提供俯視圖供駕駛者參考的 Top-View Parking Assist System[7]。這個系統覺得若能將拍攝到的影像轉成以俯視角度為主的俯視圖，會讓使用者更清楚車子和停車格間的位置關係。所以他們先使用座標轉換將後視攝影機的影像校正成俯視影像，再利用陀螺儀 (gyro)、汽車速度、移動距離等資訊擷取出影像中的運動向量(motion vector)，並使用運動向量將過去的影像(past frames)作合成，將車子經過的場景接成較完整的環境圖。另外，更配合超音波感應器(ultrasonic sensors)、及方向盤轉向資訊等得到影像中與四周障礙物的距離，建立起汽車的倒車路徑，整合成一個具有俯視功能及整體環境參考的倒車輔助系統。在這個系統中，希望呈現給使用者較完整的倒車環境與功能輔助，所以使用的額外特殊設備（如 gyro 和超音波感應器）也較多，但其實一般汽車是沒有裝設這些設備。

而 Suzuki et al.[8]中提出一種在停車格的後上方裝設一台向下斜拍的攝影機，再將拍攝到的影像利用無線網路傳送到車子內的螢幕中。因為攝影機裝設的位置是在停車格後上方，所以往下拍時可以明顯拍攝到車子和停車格，而不是只

有車後的環境，他們認為以這種方式拍攝到的影像可以較明確的顯示車子和停車格間的位置關係圖，可以更方便駕駛者參考。接著再將拍攝到的影像利用座標轉換校正成俯視圖。這裡需要注意的是，在將影像轉正成俯視圖時，影像中拍攝到的地面上有深度的物體(即正要倒車的汽車)，就會因平面轉換而產生扭曲變形。為了避免這些有深度的物體變形所導致的影像失真扭曲，而失去影像可讀性，這個系統另外再整合背景差異(background difference)和連續二個畫面間的差異(inter-frame difference)的資訊，擷取出影像中汽車的位置，再用一個和汽車大小呈相似比例的矩形來表示它。這樣的系統也許有一定的輔助程度及與可參考性，但是能呈現在影像畫面中的資訊有限，而且都只根據目前攝影機拍攝的畫面作轉換，無法整合過去的資訊。但若又要限制只能使用一台攝影機拍攝到的影像，卻又希望在螢幕中顯示更多的資訊給使用者參考，可能需要一些額外設備來輔助，如[7]中所使用的陀螺儀。由於考量一般汽車並不會裝設這些特殊設備及感應器等，所以這個系統在未來規劃中又提出使用多台攝影機架構的倒車輔助系統。希望利用三台攝影機(分別裝設在車子欲停入的目標停車格的左後上方、右後上方及車後)分別拍攝車子及周圍相關環境資訊，並整合成較完整的環境圖供駕駛者查看。

雖然[8]最後發展出來的多台攝影機架構之倒車輔助系統，並不需要使用特殊的設備來輔助，但光是在停車格需架設二台攝影機，以及汽車後需再架設一台，其架設的麻煩性與攝影機花費的成本都提高許多，而實用性與便利性則相對降低不少。在真實生活中的可行性是值得考量的。

1.4 系統流程

本論文中的倒車系統，是基於配合視訊畫面的想法下，使用一台固定在車尾位置、固定相同拍攝角度的攝影機，拍攝車後方的環境。而我們假設環境場景是停車格中間畫有一塊矩形的標記，而停車格的其餘地面上沒有複雜的花紋，如圖 1-1 所示。

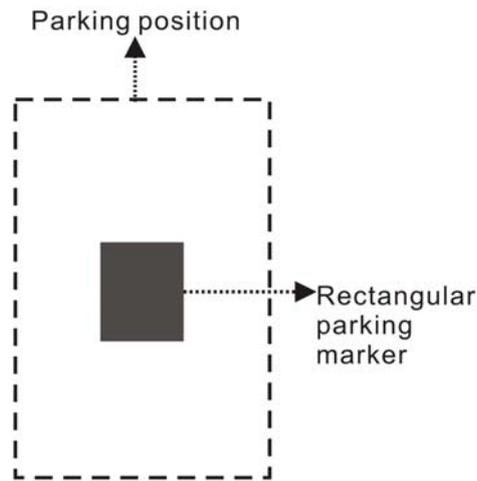


圖 1-1 實驗場景示意圖。

我們會將車後攝影機拍攝的連續二維影像，利用已知的空間資訊(如圖 1-1 中的停車標記的實際位置座標)轉正成擬真的鉛直俯視角度畫面，並配合影像中特徵點的擷取與追蹤完成整體環境的銜接，以供駕駛者倒車時參考，增加倒車的便利性與安全性，而且不需要額外使用任何特殊的裝置或設備，不會造成使用者額外負擔。系統概略的流程圖如圖 1-2 所示。

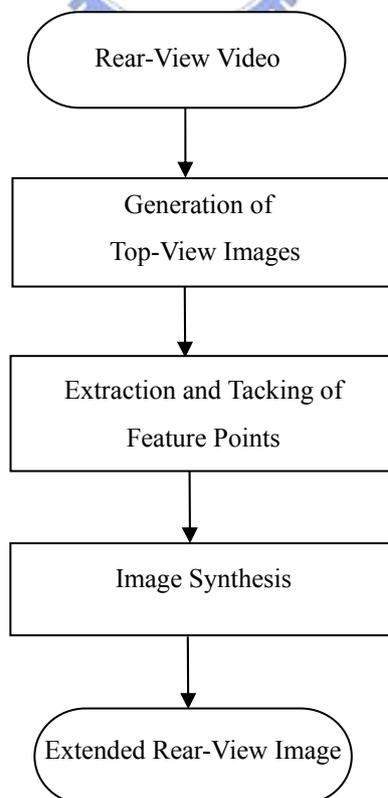


圖 1-2 簡略系統流程圖。

1.5 本論文各章節介紹

本篇論文全文共分為五章，除了本章為緒論外，第二章介紹將攝影機拍攝到的偏斜影像轉正成俯視影像的方法，內容包括一些基本的投影幾何概念與投影轉換矩陣的數學計算。第三章則是介紹特徵點的偵測與追蹤，並利用追蹤結果找出連續影像間的對應關係，將過去影像和目前影像銜接合成的詳細演算法。第四章介紹其它擷取特徵點的方法，詳述 Harris 特徵點擷取演算法及實驗結果。第五章為結論並說明本研究未來的一些研究方向。



第二章 偏斜影像校正

在我們的系統中，我們會使用到一些電腦視覺系統相關的觀念，將車後攝影機所拍攝的偏斜畫面作校正，把拍攝到的影像轉成一個擬真的鉛直俯視畫面，讓駕駛者可以藉由畫面很清楚看到車子與停車格及所處環境的相對位置關係，並作適當的對應處理與調整行車路徑。在本章裡，我們將針對系統中所採用到的每個觀念依序做簡單的介紹，其中包括了齊次座標表示法、平面投影轉換。由於電腦視覺方面的研究已經發展了相當長久的一段時間，因此在本章僅選出一些和本系統最直接相關的部份來做介紹，若想要深入瞭解或探討本章沒有提到的部份，請參考電腦視覺相關書籍[9][10]。



2.1 齊次座標系表示法

在代數學中，在平面上的任意一條直線我們可以用等式 $ax+by+c=0$ 來表示，而不同的 a 、 b 、 c 參數組合決定了不同的直線，所以我們可以使用向量的形式 $(a, b, c)^T$ 來代表平面上的一條直線。假設一個點 $p=(x_0, y_0)^T$ 在一條直線 $l=(a, b, c)^T$ 上，則將點座標代入直線方程式中，即可得到 $ax_0 + by_0 + c = 0$ 這個式子。為了簡化此式，可以使用向量內積的方式來表示它，我們改寫 p 點的座標成 $p = (x_0, y_0, 1)^T$ 的形式，則：

$$ax_0 + by_0 + c = (x_0, y_0, 1)(a, b, c)^T = p^T l = 0 \quad (2.1)$$

其中 $(x_0, y_0, 1)^T$ 和 $(a, b, c)^T$ 就是平面上點 p 和直線 l 的齊次座標表示法。

需要注意的是， $ax+by+c=0$ 和 $kax+kby+kc=0$ 代表的是相同的直線，因此在 $k \neq 0$ 情況下，寫成向量的形式 $(ka, kb, kc)^T$ 和 $(a, b, c)^T$ 也是表示同一條直線。同樣的道理， $ax_0 + by_0 + c = 0$ 和 $ka(x_0) + kb(y_0) + k(c) = 0$ 的意義也是相同的，所以在 $k \neq 0$ 的情況下， (kx_0, ky_0, k) 和 (x_0, y_0, k) 表示的是同一個點。也就是說，

任意一點 $(x_1, x_2, x_3)^T$ ，只要在 $x_3 \neq 0$ 的情況下，就可以對應到我們熟悉的歐幾里德平面上的 $(x_1/x_3, x_2/x_3)^T$ 這一點。

我們可以利用齊次座標系來化簡繁瑣的代數運算式，在電腦視覺或圖學的應用中，常使用齊次座標的概念來呈現，方便將空間的裡座標系的平移、縮放、旋轉的等複雜的方程式轉換成使用矩陣來記錄。而我們在下一節所要說明的：平面投影轉換 (homography)，就是引用了齊次座標系的概念來簡化計算式。

2.2 平面投影轉換

如果我們對同一個場景，在不同的視角拍攝兩張影像，則這兩張影像中在三維空間中共平面的點會存在某種投影轉換的關係。假設一群位於同一平面上的點集合 x_i 對應到一群另一平面上的點集合 x_i' ，若是使用齊次座標系的方式來表示 x_i 和 x_i' ，它們之間存在一個投影轉換矩陣 H 。透過 H 這個矩陣，可將 x_i 中的每個點轉換為 x_i' 中對應的點，我們稱這個投影矩陣為 homography 矩陣。而 x_i 和 x_i' 之間點的對應關係，則可用下式來表示之：

$$\lambda x_i' = H x_i \quad (2.2)$$

其中 H 是一個 3×3 的可逆 (non-singular) 矩陣， λ 為一個非零的比例常數。

2.3 投影轉換矩陣

投影轉換概念中，最重要的就是要求解投影轉換矩陣 H ，使得在平面上的一點可以對應到另一平面上的一點。如圖2-1所示。

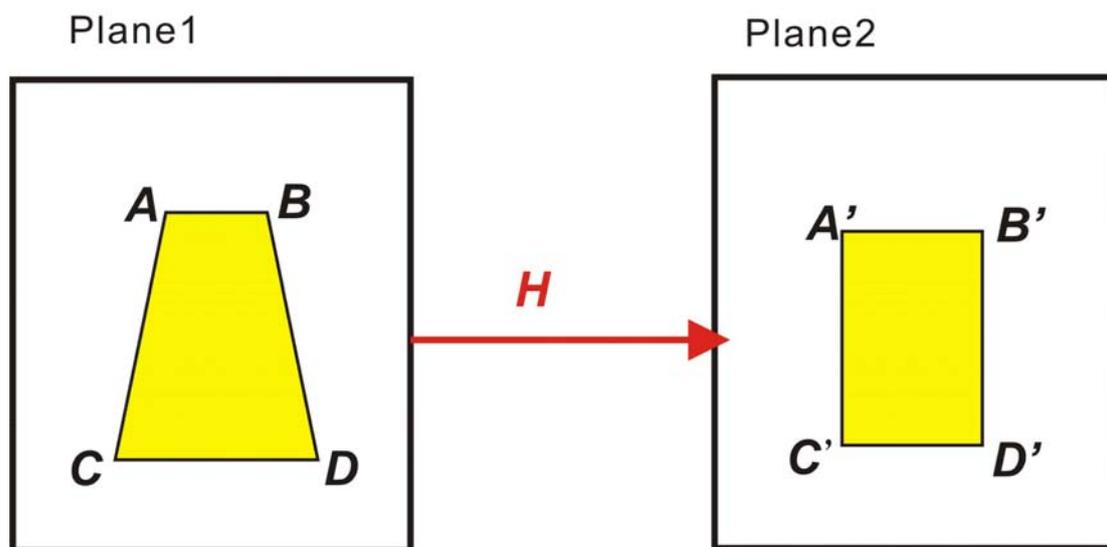


圖 2-1 利用投影轉換矩陣做平面間點對點的轉換。

我們觀察 H 可以發現， H 為一個 3×3 的可逆矩陣，矩陣中雖然有九個變數，但是因為和比例無關，因此有一個變數可以當作是縮放係數用來作正規化的，因此我們可以說，實際上一個投影轉換矩陣 H 有八個自由度。又 $2D$ 平面上的每個點都由二個變數組成（ x 座標和 y 座標），所以每一組對應的點，都會有2個限制數（number of constraint）。因此若是有 n 組對應點時，其對應關係可以用 $2n$ 個方程來表示它，其中每個方程式內有八個變數（因為 H 中的自由度為8）。由於有八個變數，故當 $n = 4$ 時我們可以求解出所有的變數。因此我們可以得知最少要有4組對應點才能求出 H 。

接著我們要利用齊次線性解（homogeneous solution）來解出 H ，此方法是使用SVD（singular value decomposition）求解。首先，先給定一組 $2D$ 平面上點的對應集合， $x_i \leftrightarrow x_i'$ ，在理想的情況下，我們期望能求出一個投影轉換矩陣 H ，滿足 $Hx_i = x_i'$ 。此外，若是允許 Hx_i 和 x_i' 向量大小不一定要完全相等，但其向量方向是相同的，它們之間就存在一個非零的比例常數 λ ，即 $Hx_i = \lambda x_i'$ 。若我們用外積的方式來表示此式，即可寫成

$$x_i' \times Hx_i = 0, \text{ with } H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (2.3)$$

將 H 的第 j 個列用 h^{jT} 來表示，我們可以得到：

$$Hx_i = \begin{bmatrix} h^{1T} x_i \\ h^{2T} x_i \\ h^{3T} x_i \end{bmatrix} \quad (2.4)$$

接著，我們利用第一節所介紹的齊次座標系來描述 x_i' ，則 $x_i' = (u_i', v_i', w_i')^T$ ，將

其代入(2.3)式的外積中的結果為：

$$x_i' \times Hx_i = \begin{bmatrix} v_i' h^{3T} x_i - w_i' h^{2T} x_i \\ w_i' h^{1T} x_i - u_i' h^{3T} x_i \\ u_i' h^{2T} x_i - v_i' h^{1T} x_i \end{bmatrix} \quad (2.5)$$

利用矩陣相乘的特性， $h^{jT} x_i = x_i^T h^j$ ， $1 \leq j \leq 3$ ，我們可以將(2.5)式改寫成下

列形式：

$$\begin{bmatrix} 0^T & -w_i' x_i^T & v_i' x_i^T \\ w_i' x_i^T & 0^T & -u_i' x_i^T \\ -v_i' x_i^T & u_i' x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (2.6)$$

最後，我們要來求解出 h^{jT} ， $1 \leq j \leq 3$ 。雖然(2.6)式中有三個方程式，但是只有前兩個是線性獨立的方程式，第三個方程式是由前兩個方程式所產生，所以我們只需要使用前兩個方程式，因此可以再一步化簡(2.6)式為：

$$\begin{bmatrix} 0^T & -w_i' x_i^T & v_i' x_i^T \\ w_i' x_i^T & 0^T & -u_i' x_i^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (2.7)$$

在本節前面的描述中，我們知道最少只需要四組對應點即可求出 H ，所以只要選取兩個對應點集合中的四組對應點 $x_i \leftrightarrow x_i'$ ($1 \leq i \leq 4$)，代入(2.7)式中，就會產生八個對應的線式方程式，進而求解出 H ，而求解 H 的演算法詳述如下：

給定 n 組對應點 $x_i \leftrightarrow x_i'$ ， $n \geq 4$ ，求出一個矩陣 H ，使得 $Hx_i = \lambda x_i'$ 。

(1) 對於 $x_i \leftrightarrow x_i'$ 中每組對應的點，從(2.6)式中可以得到一個 2×9 的矩陣 A_i 。

(2) 有 n 組對應點，就會得到 n 個 2×9 的矩陣 A_i ，將這些矩陣 A_i 組合成一個 $2n \times 9$ 的矩陣 A 。

(3) 將 A 做SVD分解，得到 $A = UDV^T$ ，其中 D 為 A 的singular value所構成的對角矩陣 (diagonal matrix)、 V 為singular vector所構成的正交矩陣 (orthogonal matrix)。而我們所要求的 H 就是最小的singular value所對應的singular vector，一般而言，SVD分解後的 D 會將singular value由大到小排列，也就是說 V 中第9行的九個元素即為構成 H 的元素。

(4) V 的第9行為： $[h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9]^T$ ，透過 (2.3) 式，可以從 V 的第9行得到 H 。

2.4 使用平面投影轉換校正影像

因為我們的倒車輔助系統是將攝影機架設在車後拍攝車後的影像，所拍攝到的影像畫面會因此有所變形扭曲，所以我們希望可以將拍攝到的影像校正成擬真的鉛直俯視畫面，讓駕駛者更方便且明白的了解車子和車後環境的關係。我們便是利用第2.2節和第2.3節所介紹的平面投影轉換，將拍攝到的影像校正。攝影機所拍攝出來的畫面，因為架設的角度和位置的關係，顯示出來的圖並不是一個鉛直俯視圖，但是以該角度所拍攝到的地平面，和現實的三維空間中我們以俯視角度拍攝到的地平面，存在著投影轉換的關係。因此，我們只要選取車後攝影機所拍攝到的影像中地平面和以俯視角度為基準的地平面上共同出現的四個標記點，就可以利用其算出這二種以不同角度拍攝出的地平面上的點之間的平面投影轉換矩陣。

在我們的系統中，攝影機所拍攝到的原始影像中的停車標記的四個頂點，是由人為手動去選取的。而俯視圖中的四個標記點位置，因為我們實際上並不是真的再另外使用一台攝影機以俯視的角度去拍攝地平面的影像，但是利用已知的環境條件，我們可以知道所拍攝到的車後影像中地平面上的那四個標記點，在以地平面為主座標軸的畫面中所對應的座標位置。如此由這四點在二個不同座標系分別所在的位置，我們就可以算出存在他們之間的平面投影轉換矩陣 H ，如圖2-2所

示。接著我們將攝影機所拍攝到的車後影像中地平面上的所有點，令其集合為 P ，乘上投影轉換矩陣 H ，即可得到 $HP = P'$ ， P' 則為鉛直俯視圖中地平面上對應的點。

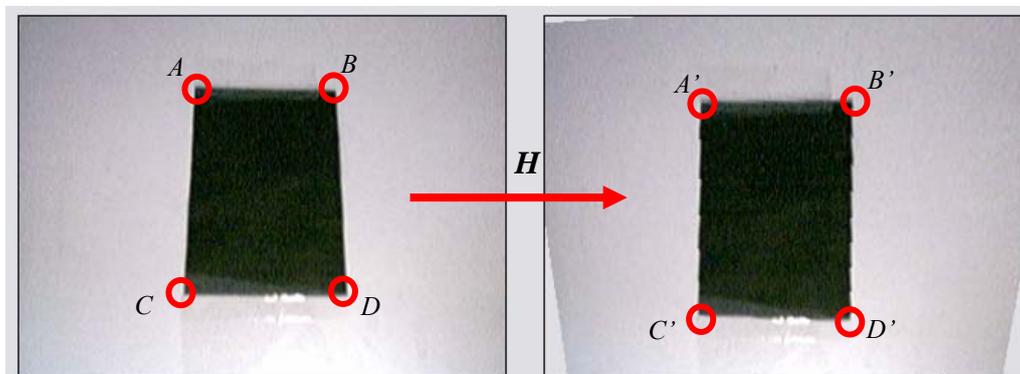


圖 2-2 車後偏斜影像利用投影轉換矩陣校正圖。

由於攝影機拍攝角度的關係，距離攝影機近的線段，其perspective projection出來的長度會較長，距離攝影機較遠的線段，其perspective projection出來的長度會比較短，所以拍攝出來的畫面中本來地面上矩形的停車格線，會被扭曲。此外，經過投影轉換矩陣轉正後，轉正的影像會出現有些座標是沒有值的情況發生，如圖 2-3 (b) 所示，所以我們還需要對轉正出來的影像中，沒有值的座標點進行補點的動作。

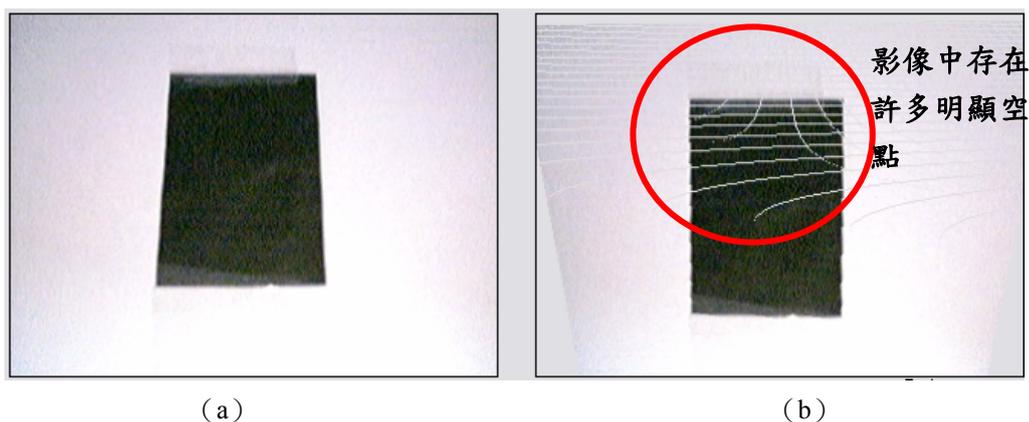


圖 2-3 影像轉正後未作補點動作的結果，
圖 (a) 為原始影像，圖 (b) 為原始影像經投影轉換矩陣轉正後的結果。

我們使用的方法是最近相鄰內插法 (nearest-neighbor interpolation)。使用此方法的原因最主要是因為速度的考量。我們考慮到本倒車系統是即時的，要將拍

攝到的影像立即迅速又正確的轉正並顯示給駕駛者參考，而最近相鄰內插法是所有內插方法中最簡單又最快速的方式。再者，使用最近相鄰內插法內插出來的結果除了邊線會有些模糊呈鋸齒狀外，整體畫面還是清楚且擬真的。因為 H 是可逆的 3×3 矩陣，所以存在 H 的反矩陣 H' 。又 $HP = P'$ ，則 $HH'P = H'P'$ ，利用 $HH' = I$ ，我們可以得到 $H'P' = P$ 這個式子。因此，我們先將 H 的反矩陣 H' 求解出來，再把轉正後影像中為空值的座標點，利用 $H'P' = P$ ，到拍攝到的影像中找到對應點的值，將該值補到轉正後的影像，這麼一來，就產生了對應的完整擬真鉛直俯視圖，如圖2-4所示。

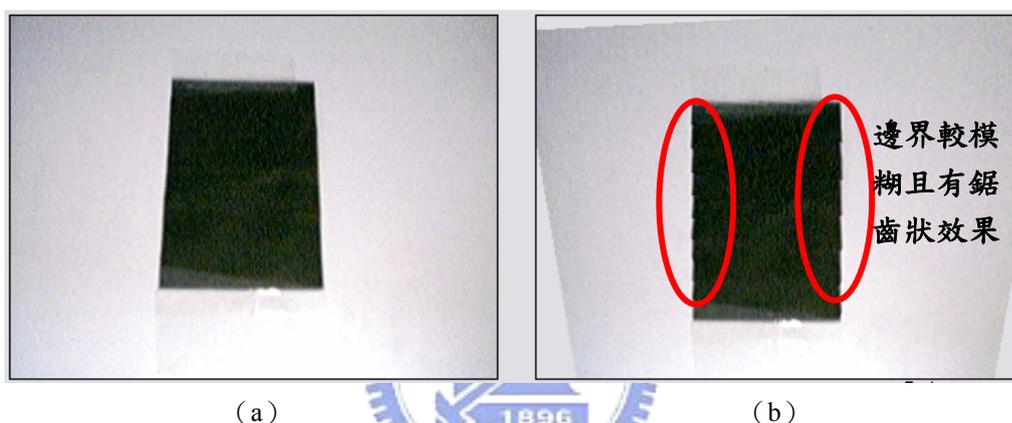


圖 2-4 影像轉正後利用最近相鄰內插法作補點動作後的結果，
圖 (a) 為原始影像，圖 (b) 為原始影像經投影轉換矩陣轉正後補點完的結果。

2.5 校正影像的縮放功能

當駕駛者使用本倒車影像系統進行倒車的動作時，有時可能會希望可以看到較細部的車後影像環境，如地面上可能有細小碎片或尖銳物；或是想看看車子與車後環境整體的關係圖，所以我們的系統提供了將校正後的影像放大縮小的功能，讓使用者可以依據自己目前的需求去調整影像畫面。因為我們已知攝影機所拍到的車後環境中地面上的四個標記點轉到擬真俯視圖中對應的四個座標點，所以我們使用的方法是將原始影像中的四個標記點所對應的俯視圖中四個座標點的範圍分別擴大和縮減，如圖2-5所示。

在圖2-5中左邊的圖形表示由攝影機拍攝到的扭曲影像，其中之四個標記點

為 A 、 B 、 C 、 D ，分別對應到圖2-5右邊俯視圖中的已知點 A' 、 B' 、 C' 、 D' 。我們分別將 A' 、 B' 、 C' 、 D' 四點間的距離拉大，使原始影像中的 A 、 B 、 C 、 D 四點對應到俯視圖中的5、6、7、8四點，根據這四點的對應，我們可以求出具有放大功能的投影轉換，利用此矩陣即可將校正影像作放大的效果。同理，我們也將 A' 、 B' 、 C' 、 D' 四點間的距離減少，使原始影像中的 A 、 B 、 C 、 D 四點對應到俯視圖中的1、2、3、4四點。而由這四點的對應，我們可以求出具有縮小功能的投影轉換矩陣，利用此矩陣就能將校正影像作縮小的功能。需要注意的是，當我們在對影像作放大功能時，影像中可能會出現更多的空點，這是因為標記點彼此間的距離被拉長了，因此我們需要作更多的內插來補點，使用的內插法仍然是最近相鄰內插法，所以呈現的放大影像會較模糊且有區塊狀況，而縮小功能的影像，因需內插的點可能幾乎沒有，所以呈現出來的影像則會較清晰，但變化複雜的區可能會有失真情形。

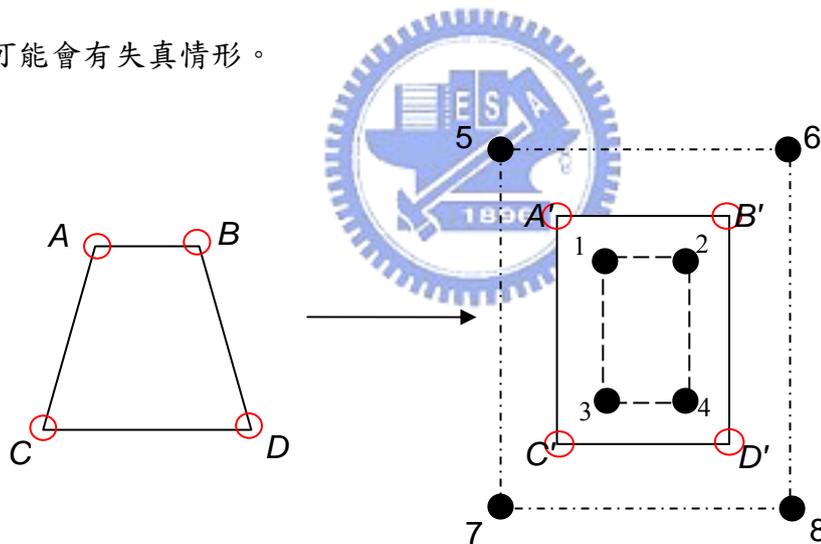


圖 2-5 將校正影像縮放的方法。

在我們的系統中，放大功能是将原始影像經過投影轉換矩陣校正後的長寬各作 1.5 倍的放大，如圖 2-6 之 (b) 所示。而縮小功能是将原始影像經過投影轉換矩陣校正後的長寬各作 1.5 倍的縮短。產生的結果如圖 2-6 之 (c) 所示。



(a)校正影像呈現原始尺寸。



(b) 校正影像放大效果。

圖 2-6 校正影像的縮放功能。



(c) 校正影像縮小效果。

圖 2-6 校正影像的縮放功能 (續)。

2.6 總結

在本章介紹了一些我們有使用到的電腦視覺系統的概念，並且利用投影轉換矩陣 (homography H) 將車後攝影像拍攝的影像轉成擬真的鉛直俯視角度的畫面，使車子及停車格的關係能夠以一個校正的畫面提供給駕駛者參考，讓駕駛者可以更清楚場景和車子的相對關係，而倒車的過程則可以更為快速且安全。

第三章 影像的合成—環境接圖

當我們將攝影機所拍攝到的車後環境影像轉成俯視圖後，為了避免其涵蓋範圍受限於目前攝影機的視角，而且又希望呈現車子周圍完整的環境圖供駕駛者參考，因此我們需要整合過去的影像，將車子倒車移動時所拍攝到的所有影像合成為一個完整的畫面。而實驗環境如第一章的圖 1-1，則我們可以利用停車格中間的矩形停車標記上的特徵點（如四個頂點），來對影像進行合成與接圖。我們使用的合成方法之主要步驟包括：

- (1) 偵測並擷取目前攝影機所拍攝到的影像中停車標記的邊線點(edge points)。
- (2) 根據(1)所找到的邊線點，找出停車標記邊線的交點(corner points)，並在連續影像中作這些交點的追蹤。
- (3) 根據追蹤的結果作影像的合成，接成一個較完整的車子周圍環境圖。

圖 3-1 為影像合成的簡略流程圖，我們將在接下來的各節中逐一介紹各個步驟的詳細作法與原理。

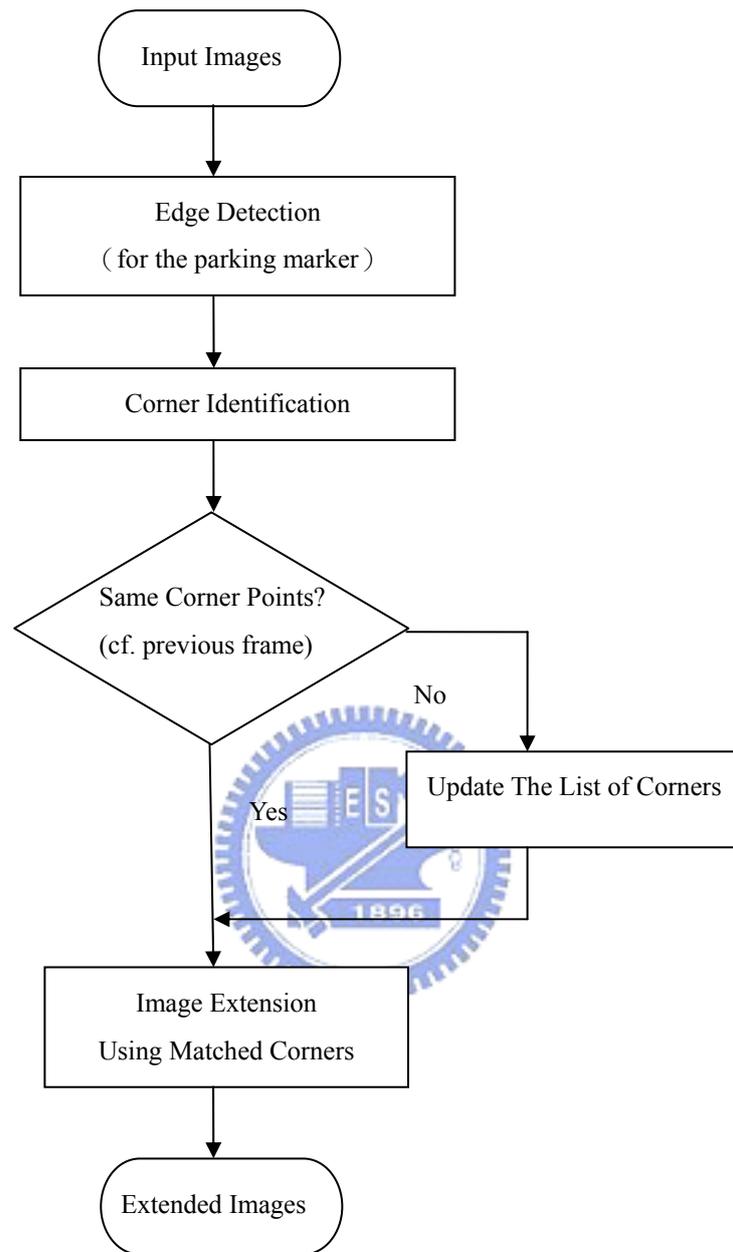


圖 3-1 影像合成的簡略流程圖。

3.1 邊線特徵點的偵測

在本系統中，我們偵測特徵點的方法是使用Sobel 遮罩，來偵測影像中的物體的邊緣點。這個運算包含兩個運算子 G_x 和 G_y ，其值如圖 3-2所示。

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

(a)

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

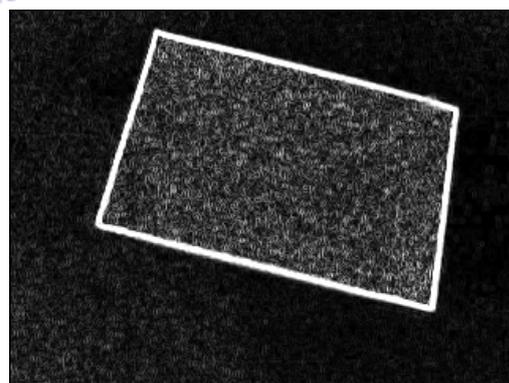
(b)

圖 3-2 Sobel 遮罩的運算子：(a) Sobel 遮罩的 X 運算子，(b) Sobel 遮罩的 Y 運算子。

利用 G_x 及 G_y 二個運算子可以分別偵測出影像中 X (水平) 與 Y (垂直) 方向的邊緣變化。因為 G_x 和 G_y 的遮罩係數總和為 0，所以如果一個點其上下左右方向相鄰的八個點都是相同的灰階值，表示這個點處於一個平緩的區域，沒有邊緣的變化，故使用 G_x 和 G_y 運算子算出來的 Sobel 值會为零。但若是一個點處於垂直邊緣或是水平邊緣，則利用 G_x 和 G_y 運算子算出來的 Sobel 值，其絕對值相對的就會很大。所以使用 Sobel 運算，我們可以偵測出影像中物體的邊緣。圖 3-3 是利用 Sobel 遮罩得到的結果，圖 3-3 (a) 為攝影機拍攝到的原始影像，圖 3-3 (b) 是將原始影像先轉成灰階影像，再使用 Sobel 遮罩得到的結果，可以明確地呈現出影像中的邊。



(a)



(b)

圖 3-3 原始影像使用 Sobel 遮罩後的結果。(a) 攝影機拍攝到的原始影像，
(b) 原始影像使用 Sobel 遮罩得到的結果。

影像經過 Sobel 運算後，為了避免受到雜訊的影響，或是擷取到一些不需要的特徵點 (如背景的花紋等)，故我們設定每點計算出來的 Sobel 值 S 必須大於一臨界值 T ，即如 3.1 式：

$$|S| \geq T \quad (3.1)$$

在我們的實驗中，我們會先將拍攝到的原始彩色影像轉成灰階影像（即將影像中 R、G、B 相加後平均），再對每點去計算 Sobel 值，臨界值 T 則設為 240。這麼一來，我們就可以過濾掉影像中一些不必要的雜訊，而只保留下來一些較明顯的影像邊緣點。

3.2 擷取角點特徵點

在上一章節中，我們使用 Sobel 運算將影像中的主要邊緣點偵測出來後，因為假設的環境場景是停車格中間畫有一塊矩形的標記，所以我們希望利用這塊矩形標記的四個頂點來作影像的接圖。因此，我們只需要先找到此矩形標記的四條邊線，接著再找出四條邊線的交點即可。找出影像中直線的方法有許多種，而目前最常用的方法就是 Hough 轉換（Hough transform），所以我們就實作 Hough 轉換以找出影像中停車標示的四條邊線。



3.2.1 Hough 轉換

首先，我們利用上一節中的 (3.1) 式，將 Sobel 值大於臨界值的影像點擷取出來，再使用這些點去作 Hough 轉換，進一步得到影像中所有直線的直線方程式。在實作 Hough 轉換的過程中，我們必須設定一些參數，包括指定 $\rho\theta$ 平面的分割，以及 $\rho\theta$ 平面的累加器單元（accumulator cells）的計數臨界值。若是 $\rho\theta$ 平面分割愈細，則最後計算出來的直線參數表示會愈精確，但是計算量也相對變大許多，但若分割太大，計算量雖然較小，但是直線的參數表示就會有較大的誤差。因此如何分割 $\rho\theta$ 平面以取得較適當的平衡，是值得我們深入探討的。在我們的實驗中，我們是把 $\rho\theta$ 各細分成 100 個增量，而累加器單元的計數臨界值是設定為 250，而實作出的結果如圖 3-4 所示。在圖 3-4 中，紅色的線是我們利用 Hough 轉換找出的停車標記的四條邊線，很明顯地，跟影像中的停車標記有些誤差（如

圖中藍圈所框範圍)，這是因為受到 $\rho\theta$ 分割大小的影響。若我們將 $\rho\theta$ 分割更細，則可以得到較精確的結果，但計算量相對較大。在我們的實驗中， $\rho\theta$ 平面分割成 100×100 ，其計算一張影像平均要花 0.9~1 秒的時間，對於一個即時系統其執行速度並不是很理想，若是再將其更一步的細分，則時間耗費更多，執行速度更為緩慢，結果就不是我們樂意看到的。因此在此我們決定使用別的方法來找出影像中的線，也就是用掃瞄線(scan line)配合PCA的方法，找出影像中停車標記的四條邊線。

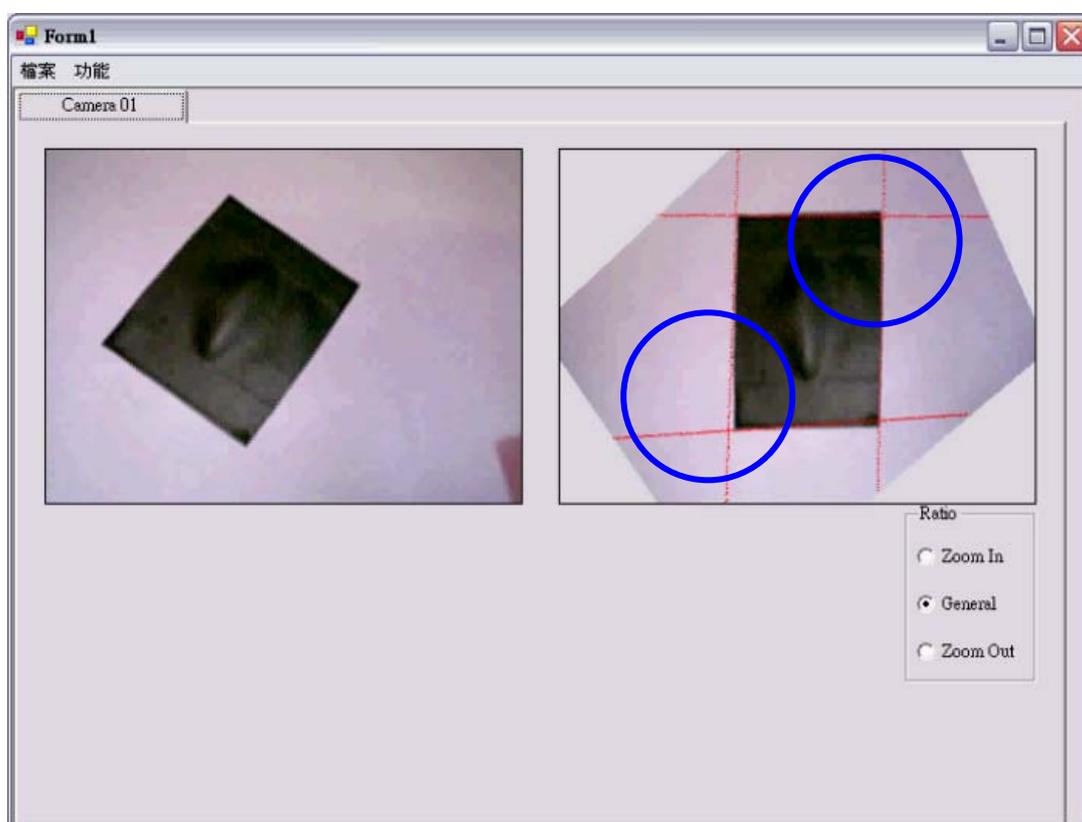


圖 3-4 利用 Hough 轉換找出停車標記的實驗結果。

3.2.2 掃瞄線配合 PCA 方法

在本方法中，我們先利用掃瞄線的方式將停車標記的邊線點找出並分群，再將分屬不同群（即不同直線）的影像點做 PCA，找出每群點的直線方程式。首先我們利用(3.2)式將原始影像轉成一張只有明顯物體邊緣的二值化影像(binary image)。

$$\begin{cases} I(x, y) = 1 & \text{if } |S_{x,y}| \geq T \\ I(x, y) = 0 & \text{if } |S_{x,y}| < T \end{cases} \quad (3.2)$$

其中 $S_{x,y}$ 是指影像座標 (x, y) 的 Sobel 值，而 $I(x, y)$ 是指對應的二值化影像中座標 (x, y) 的值。接著，我們分別從剛剛得到的二值化影像的四個頂點出發，做垂直與水平方向的掃描，如圖 3-5 所示（僅顯示由左上角與右上角作掃描之方式）。

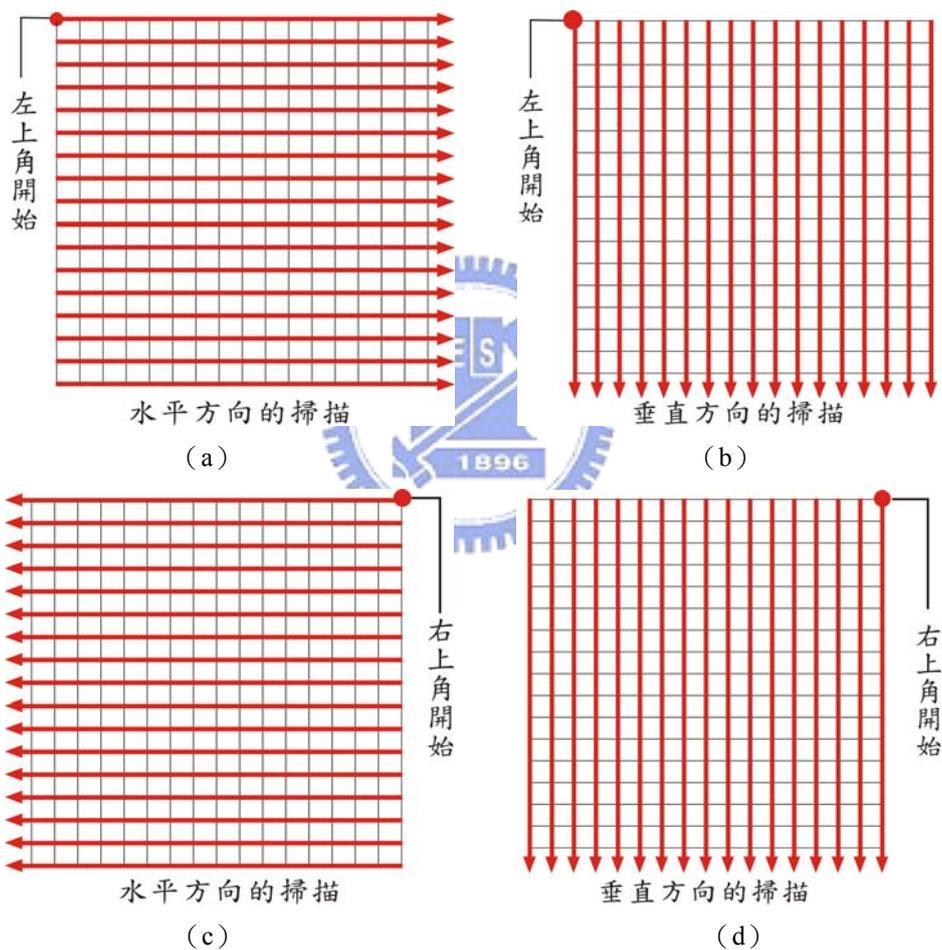


圖 3-5 將影像分別從左上角與右上角為起點，做水平掃描及垂直掃描的圖示。

在掃描的過程中，以左上角開始水平掃描為例（即圖 3-5 (a)），是先由影像左上角為起點，各列由左往右掃描，再由各行自上而下掃描。因為我們假設實驗的環境中，只有一塊矩形的停車標記（如第一章的圖 1-1 所示），而沒有其它複雜的花紋，所以只要由左往右掃描到的第一個點，我們會視該點為矩形標記的邊

線，因此就會將該點保留下來，接著就直接跳到下一列，再由左往右掃瞄，不斷重覆這些步驟，直到掃完整張影像為止。圖 3-6 為由左上角為起點水平掃瞄的最後結果之示意圖，圖中黑色和灰色的點皆為物體的邊界，但我們掃瞄完後只會保留左側的外圍邊界點，即黑色點的区域。

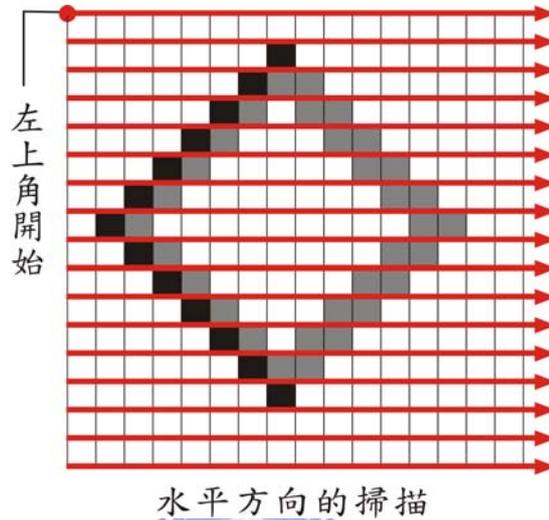


圖 3-6 由左上角開始水平掃瞄的結果示意圖。

當我們分別完成由四個頂點開始的水平掃瞄和垂直掃瞄，即可得到停車格中間標記的外圍邊界點。但因為影像中可能會混雜著些許雜訊，所以我們要對找到的邊界點進行篩選，過濾掉一些雜訊的點。我們使用的過濾機制是檢查線段上的每點位置的差異值，如果是從左上角開始水平掃瞄的方式，那麼每兩列間的邊界點的 x 座標值差異（如圖 3-7 中的 Δx ）就必須要很小，在此我們設定兩列間的 Δx 值最多只能差 1，即設定 $|\Delta x| < 2$ 這個條件，符合這個條件的邊界點才保留下來，否則就視其為一個雜訊點。圖 3-7 為簡單的示意圖，黑色區域為我們掃瞄出來的邊界點，每列旁的 Δx 說明了每兩列間各點的水平距離，圖 3-7 中藍色圓框圈的點，其 $|\Delta x|=3$ ，所以會被視為是雜訊點。因為如果是一條連續的線段，那麼該線段上的每點，其列與列之間的水平距離都是相近的，所以使用 Δx 來判斷是合理的。

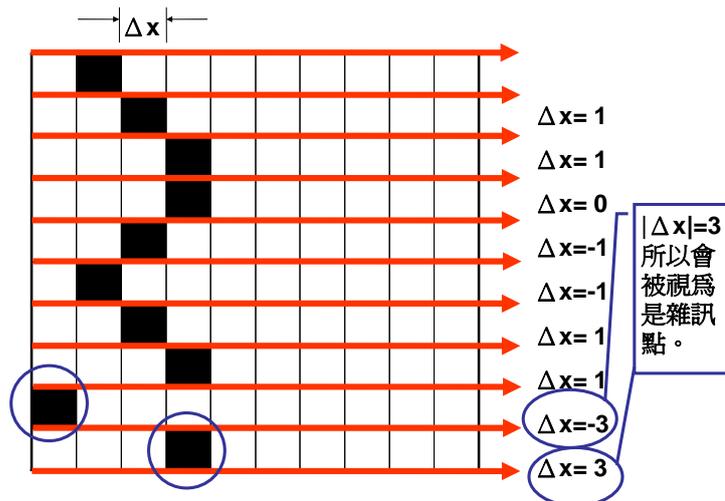


圖 3-7 水平掃瞄的過濾雜點機制之簡單示意圖。

使用上述的過濾雜點的方法刪除一些可能存在的雜點後，我們即可得到較正確的標記邊界。雖然標記本身是矩形的樣式，但由於受到車後攝影機角度和位置的影響，拍攝到的標記在影像中可能會呈現梯形或平行四邊形，所以我們就必須判斷掃瞄到的邊界點是由一個邊組成或是二個邊組成，如圖 3-8 所示。在圖 3-8 中，圖 (a) 表示掃到的邊界點只屬於一條線上的點，圖 (b) 則表示掃到的邊界點分別屬於二條不同的線上的點。如果掃到的邊界點是由二條邊組成，就必須找出二條邊的交界處，將這些掃瞄到的邊界點分群成二條邊上的點，進而找出標記的四個邊。

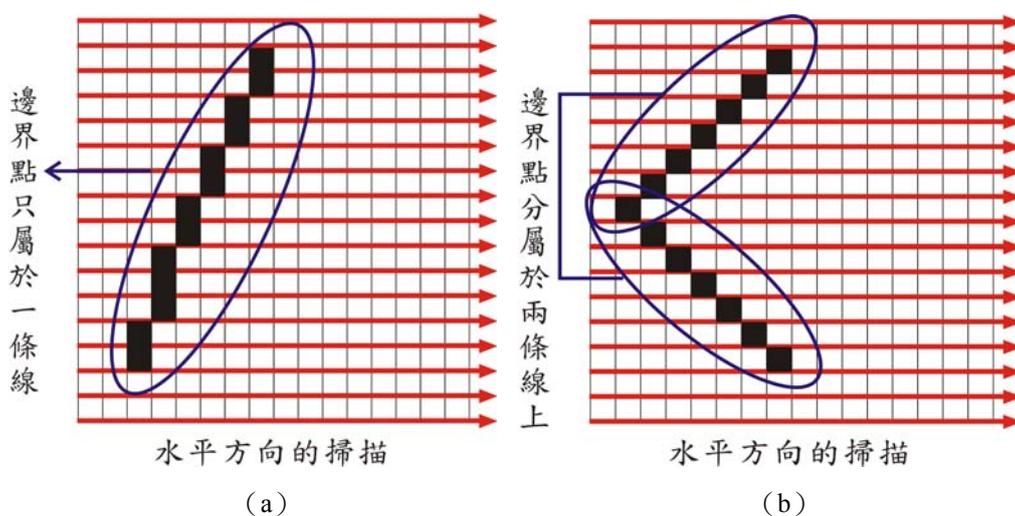


圖 3-8 水平掃瞄到的邊界點之分布情況。

因此我們需要先判斷掃到的點是由一條直線或是二條線組成。所使用的方法

是以斜率來判斷，如果掃瞄到的邊界點只是屬於一條線，則這條線上的點不論從哪個點開始計算斜率，其算出來的結果都是相近的；換句話說，如果掃瞄出的邊界點是分屬二條邊上的點，因為這兩條邊不會是平行線，所以兩條邊上的點其斜率算出來的結果會差距很大。由此方式利用斜率來區分掃瞄到的邊界點是屬於一條線或是屬於兩條線是合理且快速的。我們計算斜率的方式，如果是水平掃瞄的話，會先將掃瞄到的點依 y 座標的值排序，再將 y 座標值最大的幾個點和最小的幾個點分別作斜率的計算，在我們的系統中，是使用 y 座標值最大的十五個點和 y 座標值最小的十五個點，依據 (3.3) 式計算該點的斜率。

$$slope_i = (x_{i+5} - x_i) / (y_{i+5} - y_i) \quad (3.3)$$

在 (3.3) 式中， $slope_i$ 是指掃瞄到的第 i 個點的斜率， x_i 則是指掃瞄到的第 i 個點的 x 座標值， y_i 則是掃瞄到的第 i 個點的 y 座標值。

接著我們就會判斷， y 值最大的十五個點算出來的斜率平均值，與 y 值最小的十五個點算出來的斜率平均值差距是否很小，若是差距很小（在此我們是設定差距在 ± 0.2 之間），就認為掃瞄到的點只屬於一條線。若是差距很大（即差距 > 0.2 ），我們就認定掃瞄到的點分屬於二條邊上。如果是判斷掃瞄到的結果分屬於兩條邊的情況時，則需要更進一步的找出這兩條邊的交界點，以將掃瞄到的點分群到二條線上。

之前我們是利用斜率判斷出掃瞄到的邊界點是分屬於一條線或二條線，在此我們同樣是以斜率來找出二條線的交界點。我們依照邊界點的 y 座標值由小至大的順序，利用 (3.3) 式依序求出每個邊界點的斜率。再將每個邊界點的斜率，和之前算出來的 y 值最小的十五個點的平均斜率比較，如果兩者差距在一定的範圍內（在此，我們設定範圍是 ± 0.2 內），表示這個邊界點還在第一條線上；如果兩者差距超過了我們設定的範圍（即超過 ± 0.2 ），我們會再判斷它的下一個邊界點算出來的斜率是否也跟 y 值最小的前十五個點的平均斜率差距超過設定的範圍，若是差距仍然太大，則可判定二條邊的交界點已出現，即為目前的邊界點的下面第五個點，就是二條線的交界點（假設目前邊界點為第 i 個點 (x_i, y_i) ），則

交界點就是第 $i+5$ 個點 (x_{i+5}, y_{i+5})。找出交界點後，我們就可以把掃瞄到的點分群到二條線上，也就是說，我們已經找出了矩形停車標記外框的二條邊線。

做完以影像左上角為起始點的水平掃瞄後，我們會再以相同起點（左上角）開始做垂直掃瞄，掃瞄最後的結果，同樣的也會得到停車標記的一條邊或二條邊。這麼一來，我們最多可以得到停車標記的四條邊線（其中有二條是重覆）。如圖 3-9 中，因為圖 (a) 和圖 (b) 中的停車標記為不同的角度，所以掃瞄出來的結果也會有所不同。在圖 (a) 中做水平掃瞄及垂直掃瞄的結果都會各得一條邊，但是在圖 (b) 中，垂直掃瞄和水平掃瞄的就會各得到二條邊，但是其中一條是重複的，即圖中虛線框出的邊，就是用水平掃瞄和垂直掃瞄都能掃瞄到的停車標記的框線。因此，影像在經水平掃瞄和垂直掃瞄後最多可以得到三條邊，但是為了之後利用邊的交界找角點特徵點時可以較為精確且快速，我們需要篩選掃瞄到的邊。

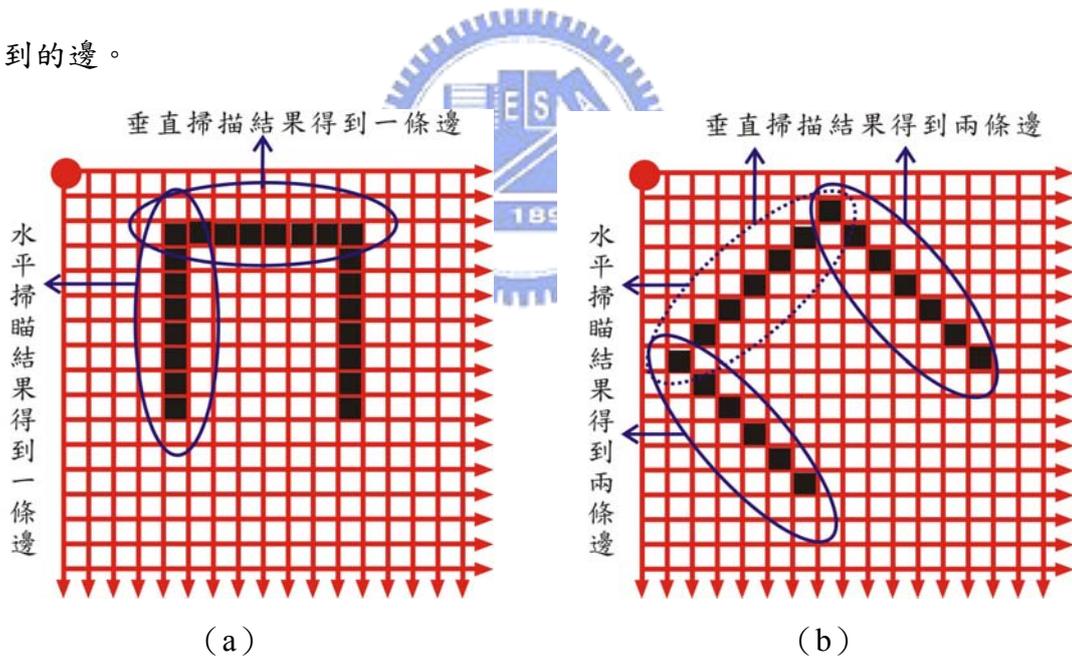


圖 3-9 不同角度停車標記由左上角開始水平掃瞄和垂直掃瞄的結果。(a) 對影像中的停車標記做水平和垂直掃瞄後，最後各得到一條邊的結果，(b) 對影像中的停車標記使用水平和垂直掃瞄後，最後各得到二條邊的結果。

篩選邊的方式很簡單，因為我們希望最後得到的邊界點可以較完整，意即在一條線上要有足夠多的點，而且各點之間，其座標值不要差太多。若是線上的每個點彼此間 y 座標值或 x 座標值都差距到 3~4 個像素，就表示掃到的這些點可能

並不是很穩定，之後當我們要進一步利用這些點找出跟這些點座標值最小的一條直線時，誤差就會相對變大。所以整體而言，我們會將掃描到的每一條邊的點，先做點數的篩選，我們在此是假設每條邊上的點數需大於 30 個點，這是依據停車標記在影像中的大小衡量估計出的最小邊線長度。接著再使用 (3.4) 式來評估線的穩定度 (*stability*)。

$$S_p = (y_n - y_0) / \text{number of the line points} \quad (3.4)$$

在 (3.4) 式中， y_n 表示我們掃描到的這條邊上的點中最大的 y 座標值， y_0 則表示這條邊上的點最小的 y 座標值。利用 (3.4) 式算出來的結果，如果值愈近似 1 的話，就表示這條線上的點應該是平均分布且密集的，不會有零落的分散情況。這裡我們設定的範圍是和 1 差距在 ± 0.3 之內，只要掃描到的邊上的點計算結果在此範圍內，我們就判定這群邊線點是穩定的，圖 3-10 是篩選邊的流程圖。若是最後水平掃描和垂直掃描篩選出來的結果超過二條線的話，我們就利用穩定度篩選出穩定度最好的二群邊線點（即利用 (3.4) 式算出來的結果和 1 最相近的前兩條線），過程如圖 3-11 所示。



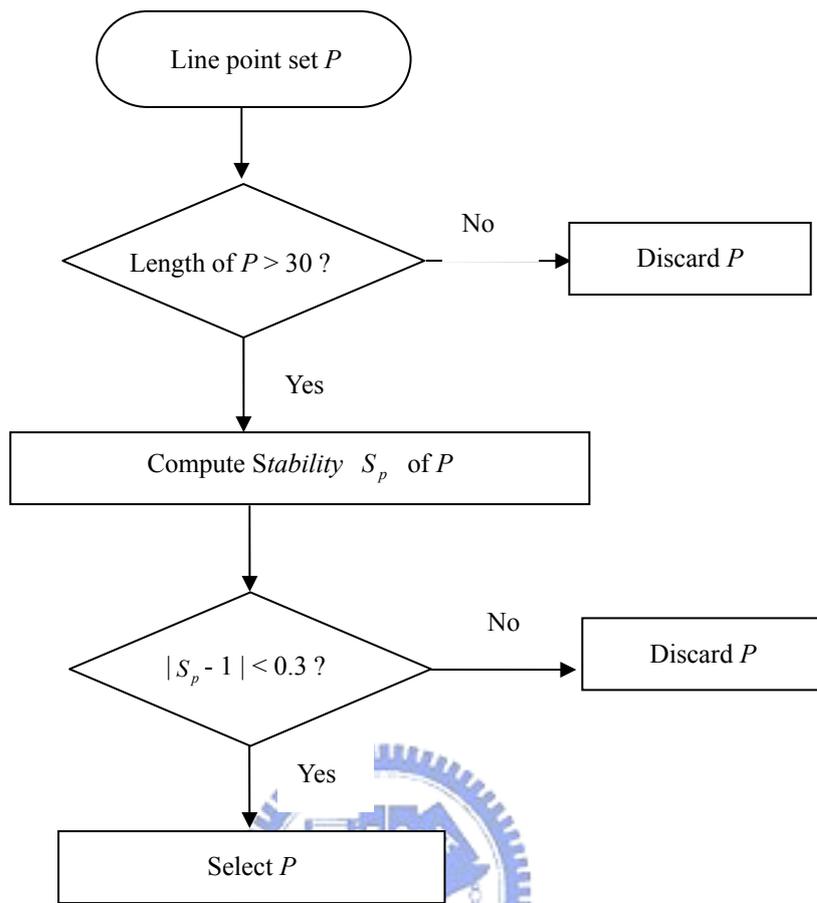


圖 3-10 篩選邊流程圖。

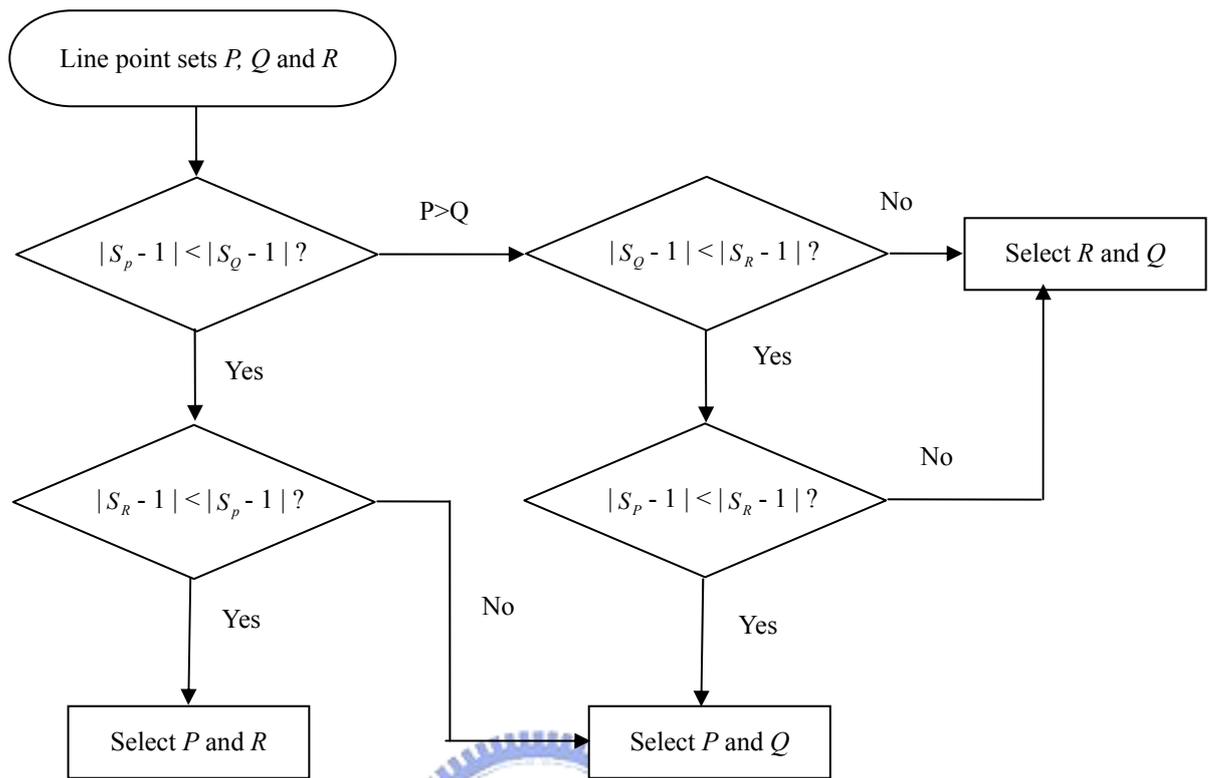


圖 3-11 選出最後二群線上點之流程圖。

在我們分別完成左上角、右上角、左下角及右下角為起點的水平掃瞄和垂直掃瞄後，共可以得到八條邊（其中至少四條以上是重覆的邊）。但一般在追蹤的過程中，只需要追蹤幾個較具有意義的特徵點即可，不需要一次追蹤一大群點，因此我們希望利用掃瞄得到的八條邊找出停車標記的四個角點，並從中選出兩個角點來進行追蹤。但由於掃瞄出來的八條邊是由八群點組成的（其中四群以上是重覆的點），所以要找出停車標記的四個角點，就必須先找出這群點各自對應的直線的交點。因此我們就針對這八群點，以 line fitting 的方法，找出與這些座標點誤差最小的八條直線。這個問題（即 line fitting）有很多數學上的解法，在我們的系統中，是使用主成分分析法（principal components analysis）[11]。

主成分分析法（PCA）是一種將資訊從高維度降低至低維度的統計方法，此技巧利用線性轉換將資料點轉換至一新座標系，在此新座標系內，所有原座標系內的點座標，其最大變異量會沿著第一象限軸方向（稱為第一主成分，first principal component），而次大變異量會沿著第二象限軸方向（稱為第二主成分，

second principal component)，依此類推。在將資訊從高維度降到低維度的過程中，PCA會依變異量保留原座標系資訊的特性，將維度較高的原始資訊特徵，利用維度較低的特徵來表示，如此一來既可以加快計算速度，又可以維持一定的準確度。以下我們將簡單的介紹PCA的計算流程。

已知有 n 個二維空間的點， (x_i, y_i) ， $1 \leq i \leq n$ ，欲使用 PCA 找出最能表示這些點朝向的線。我們先計算出這些點的重心座標，如 (3.5) 式：

$$(\bar{x}, \bar{y}) = \left(\frac{\sum_{i=1}^n x_i}{n}, \frac{\sum_{i=1}^n y_i}{n} \right) \quad (3.5)$$

再將每個資料點的座標減去重心座標，將座標系原點平移，如 (3.6) 式：

$$(x_i', y_i') = (x_i - \bar{x}, y_i - \bar{y}), \quad 1 \leq i \leq n \quad (3.6)$$

得到了這些平移微調過的點座標後，我們再利用這些點座標計算共變異數矩陣

(covariance matrix)，此矩陣的計算方式如下：

$$C = \begin{pmatrix} \text{cov}(X, X) & \text{cov}(X, Y) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) \end{pmatrix} \quad (3.7)$$

(3.7) 式中的 X 及 Y 分別表示 x_i 的座標集合及 y_i 的座標集合，而共變異數 (covariance) 的算法為：

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)} = \frac{\sum_{i=1}^n x_i' y_i'}{(n-1)} \quad (3.8)$$

至此，我們已經由原始資料點算出代表其間相互變異關係的共變異數矩陣，下一步就是對 C 做特徵向量分解 (eigenvector decomposition)，可求得兩個特徵值 (eigenvalue) 及其對應的特徵向量 (eigenvector)。其中，值較大的特徵值對應的特徵向量，稱為是第一主成分向量，而值較小的特徵值所對應的特徵向量稱為是第二主成分向量，則我們選取數值最大的特徵值所對應的特徵向量 (即第一主成分向量) 作為這些資料點的朝向線。圖 3-12 為利用 PCA 得出的主成分 (principal components) 的示意圖。

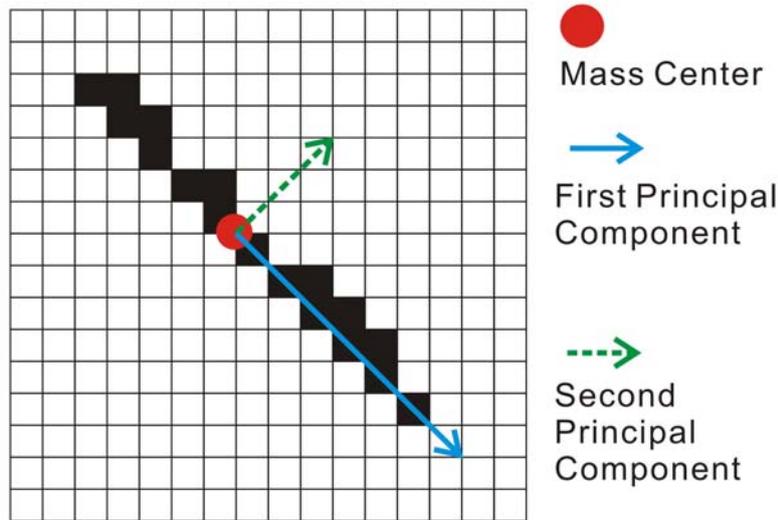


圖 3-12 利用 PCA 得出 Principal Components 之示意圖。

假設我們選取的特徵向量為 $V = [v_1 \ v_2]^T$ ，則此向量即可表示為這些資料點的朝向。若需進一步求出此朝向線的直線方程式 $l: y = ax + b$ ，則將 v_2 、 v_1 相除可得此線的斜率，又知 (3.5) 式中求得的資料點的重心必在此線上，因此有了斜率及直線通過的點，便可以用點斜式表示出此線的方程式，如 (3.9) 式：

$$l: y = \frac{v_2}{v_1} x + \left[\bar{y} - \left(\frac{v_2}{v_1} \right) \bar{x} \right] \quad (3.9)$$

我們可以用此方程式來表示掃描到的八群邊線點。

這一小節所介紹的掃描線配合 PCA 方法，步驟看起來繁複，但每個步驟的計算都很單純，所以運算量很小，因此總共花費的執行時間平均只須 0.3 秒，即為 Hough 轉換的三分之一。

3.2.3 擷取角點特徵點

當我們對由左上角、左下角、右上角與右下角開始的水平與垂直掃描得到的八群點分別做完 PCA 後，可以得到每一群點所表示之直線的直線方程式。根據每二條線的直線方程式，用高斯消去法解聯立方程式，即可求出二條直線的交點。在此需要注意的是，因為八群點中至少有四群以上是重複的邊上點，也就是說，至少會有四條以上相同的直線方程式，而且有些線可能會是平行的狀況，就會沒有交點，因此需要先對這些線做平行的判斷。我們是根據 (3.10) 式，使用角度

方法來判斷：

$$\sin \theta = \frac{\bar{a} \times \bar{b}}{\|\bar{a}\| \cdot \|\bar{b}\|} \quad (3.10)$$

在 (3.10) 式， \bar{a} 和 \bar{b} 表示二條直線的向量（在此即為每群點用PCA算出來的第一主成分向量表示），如果二個向量求出來的 $\sin \theta \cong 0$ ，表示二條直線是互相平行的。

求出八條邊每兩條間各自的交點之後，最多可以得到四個不同的交點（因為停車標示只有四條邊），則我們就擷取出矩形停車標示的角點特徵點了。本節所介紹的擷取角點特徵點的簡單流程圖如圖3-13所示。

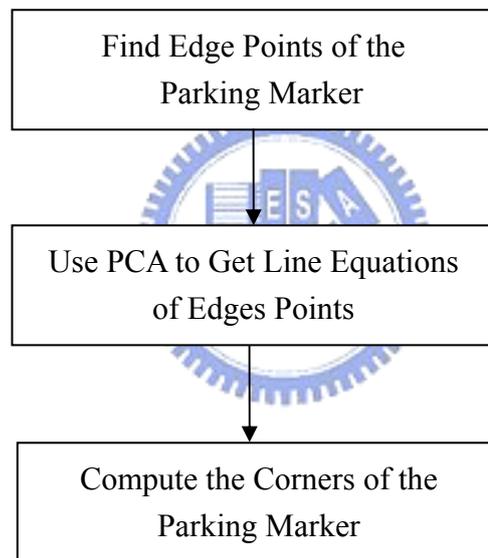


圖 3-13 擷取角點特徵點的簡略流程圖。

3.3 連續影像中的角點特徵點追蹤

一般來說，目前針對連續影像的追蹤皆假設影像是在時間空間變化緩和限制（spatial-temporal smoothness constraint）下實行。運動緩和限制是假設相機或物體在空間中的移動是緩慢且具有慣性的，而不會有突然劇烈的加速或是方向劇變的情形；且連續兩張影像之間的時間間隔非常短，使得兩張影像其差異性不大，

且物體的移動具有連續性。這個限制假設在倒車過程中是合理的，因為駕駛者在倒車時一般都是緩慢移動且不會有突然加速或跳動的情況，所以我們可以在這樣的假設前提下，針對連續影像進行追蹤。

我們在上節中找到了目前影像中停車標示的四個角點後，便將這四個點的座標保存下來，並在下一張影像中所擷取的對應角點集合中，尋找座標值跟這四點座標值，其 x 座標值相差在 10 個像素內， y 座標值也相差在 10 個像素內的點，當作是追蹤到的下一張影像中對應的四個角點。在追蹤的過程中，我們可以根據連續兩張影像對應的特徵角點，找出兩張影像間的旋轉與平移，進一步對連續影像進行合成與接圖。

3.4 影像的合成

車後攝影機在拍攝車後影像時，會受限於目前攝影機的位置與角度，而只能提供部份車後環境影像供駕駛者參考，這樣能提供的環境資訊可能稍嫌不足，因此在本系統中，希望可以將攝影機所拍攝到的過去影像和目前影像中的環境資訊整合，銜接出較完整的車子周圍場景，讓駕駛者可以更清楚與了解車子與周圍環境的關係，簡單的合成預期成果圖如圖 3-14 所示。

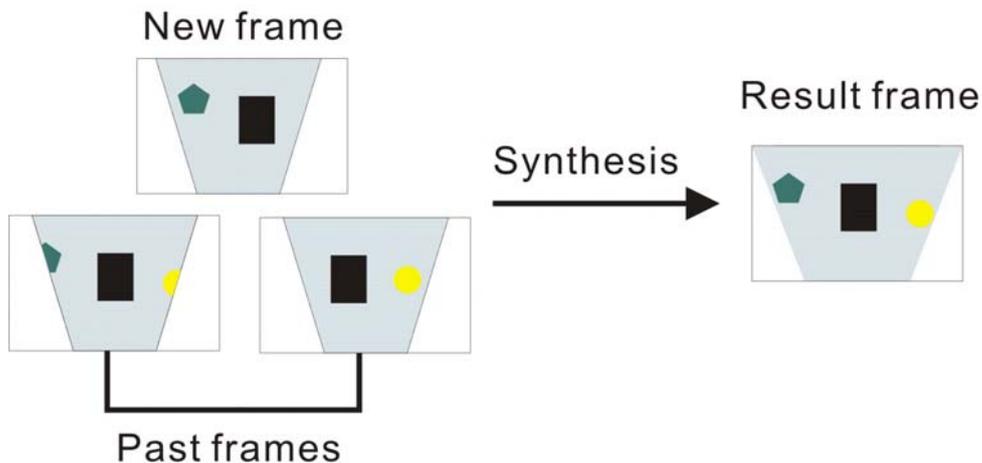


圖 3-14 影像合成預期結果示意圖。

在對過去影像和目前影像進行合成前，我們需要先找出連續兩張影像中旋轉與平移的關係。此時利用之前追蹤時找到的連續二張影像中對應的特徵角點即可找出二張影像間的對應關係。首先先從擷取到的四個角點中，選出二個跟攝影機距離最近的角點（即 y 座標值最大的二個角點），這是考量到距離攝影機愈近較不會有失真情況發生的因素。選出連續兩張影像中對應的二個主要角點後，利用第二章求出的投影轉換矩陣 H ，分別求出二個角點在前後二張俯視圖中對應的座標值，如圖 3-15 所示。在這之後，我們就是利用連續影像轉正後對應的角點特徵點座標，來找出連續影像間的旋轉與平移。

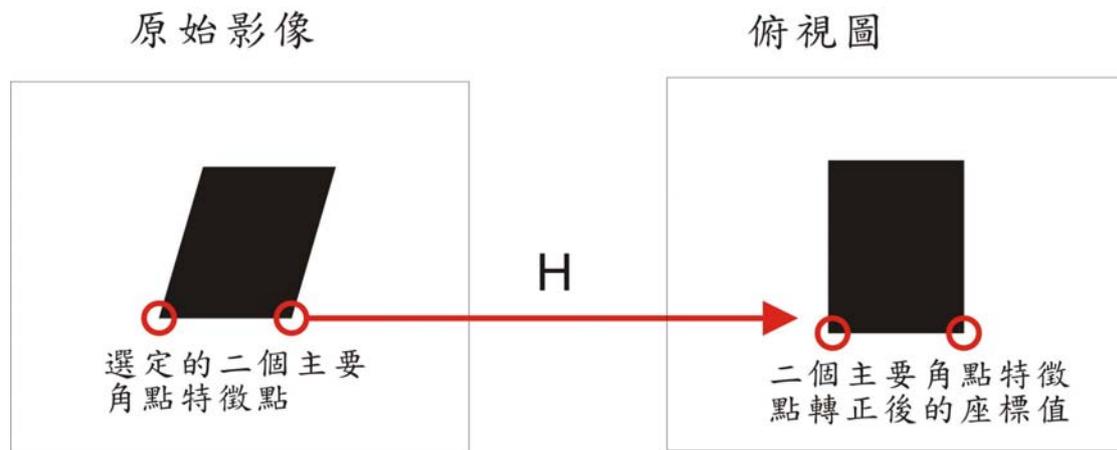


圖 3-15 找出角點校正後的座標值。

因為本系統使用的影像合成方式，主要是以目前校正後的畫面為主，將目前校正後的畫面，根據過去影像中所拍攝到的而目前攝影機沒拍攝到的區域，利用旋轉和平移將這些區域點貼補到目前的畫面中，圖 3-16 為合成的方法示意圖。

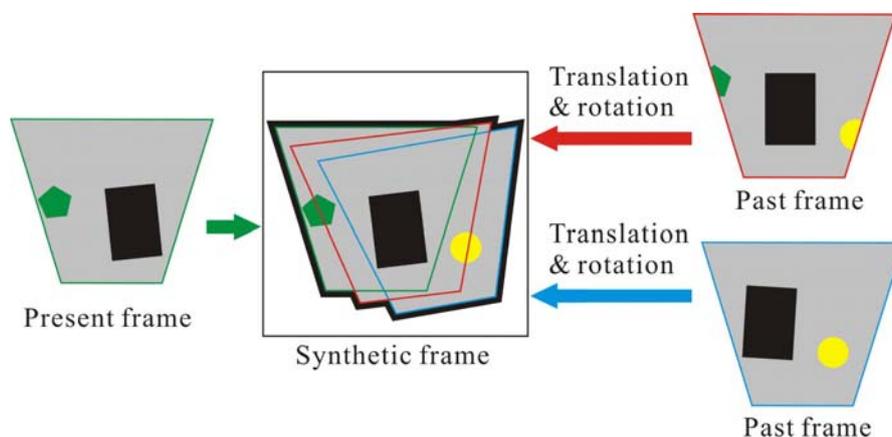


圖 3-16 合成方法簡略示意圖。

找出二張影像間旋轉和平移的計算過程如下：

假設在第 m 張影像轉正的俯視圖中，找到的主要角點為 (p_1, q_1) 和 (p_2, q_2) ，則這兩點的中心座標：

$$(x_c, y_c) = \left(\frac{p_1 + p_2}{2}, \frac{q_1 + q_2}{2} \right) \quad (3.11)$$

再將第 m 張影像轉正的俯視圖中所有的點座標 (x_i, y_i) 值減去中心座標 (x_c, y_c) ，將座標系原點平移，如 (3.12 式)：

$$(x_i', y_i') = (x_i - x_c, y_i - y_c), \quad 1 \leq i \leq n \quad (3.12)$$

若在第 $m+1$ 張影像轉正的俯視圖中， (p_1, q_1) 和 (p_2, q_2) 對應的角點分別是 (p_3, q_3) 和 (p_4, q_4) ，則這兩張影像 (第 m 張和第 $m+1$ 張) 的旋轉矩陣 R 如下式：

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} \frac{\bar{a} \cdot \bar{b}}{\|\bar{a}\| \cdot \|\bar{b}\|} & -\frac{\bar{a} \times \bar{b}}{\|\bar{a}\| \cdot \|\bar{b}\|} \\ \frac{\bar{a} \times \bar{b}}{\|\bar{a}\| \cdot \|\bar{b}\|} & \frac{\bar{a} \cdot \bar{b}}{\|\bar{a}\| \cdot \|\bar{b}\|} \end{bmatrix} \quad (3.13)$$

其中 $\bar{a} = (p_3 - p_1, q_3 - q_1)$ ， $\bar{b} = (p_4 - p_2, q_4 - q_2)$ 。

則利用 (3.12) 式和 (3.13) 式，我們即可將第 m 張俯視圖中的各點，依據 (3.14) 式轉到第 $m+1$ 張影像的座標系，其新座標值為 $(newx_i, newy_i)$ 。

$$\begin{bmatrix} newx_i \\ newy_i \end{bmatrix} = R \cdot \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} + T, \quad 1 \leq i \leq n \quad (3.14)$$

其中

$$T = \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} \frac{(p_3 + p_4)}{2} \\ \frac{(q_3 + q_4)}{2} \end{bmatrix}$$

因此我們可將第 $m+1$ 張影像轉正的俯視圖中空值的點，補上由第 m 張根據 (3.14) 式得到的點的影像值，就完成了影像的合成。這裡值得注意的是，我們會以合成出來的結果圖，當成是新的第 $m+1$ 張影像，而把原始的第 $m+1$ 張影像覆蓋。當有新的影像 (如第 $m+2$ 張影像) 進入我們系統中，作影像合成時，即以新的第 $m+1$ 張影像和第 $m+2$ 張影像作合成，因此就能將所有過去的影像資料都保存起來，將車子周圍的環境做完整的銜接。圖 3-17 為本系統的合成影像方法之簡單示

意圖。

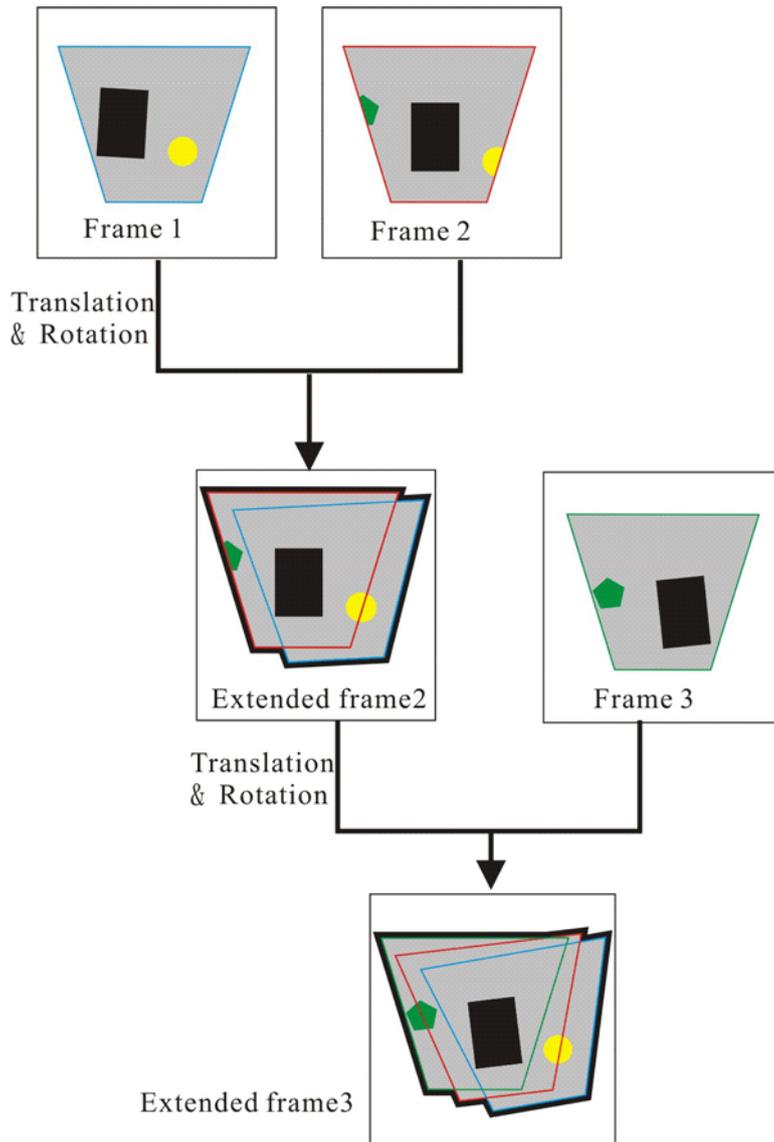
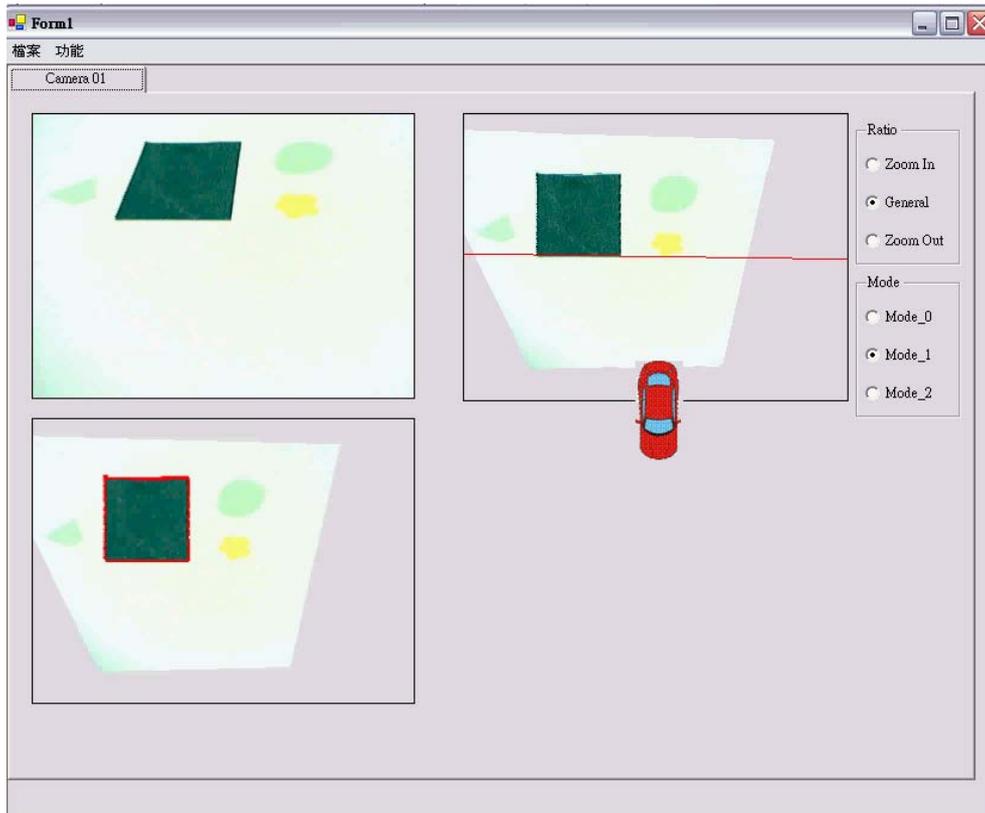


圖 3-17 合成影像過程之示意圖。

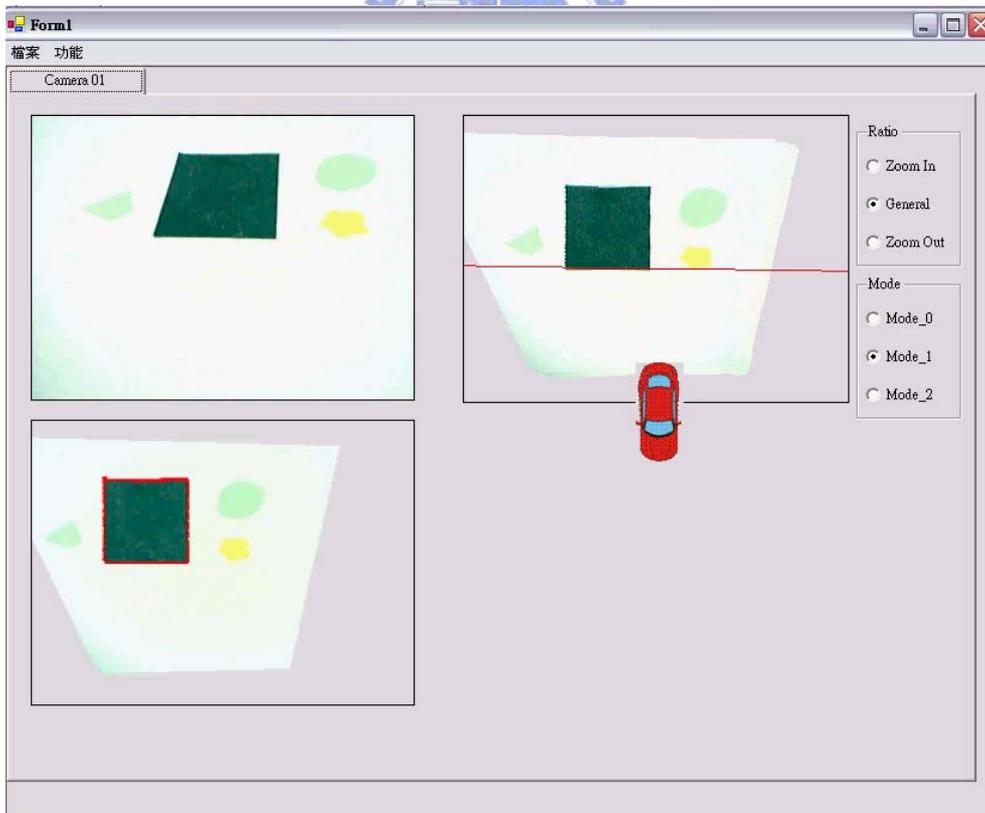
上述方法合成的擴大影像實驗結果如圖 3-18所示，(a) ~ (f) 分別為實驗結果video中第 1、20、40、60、80、100 和 120 張畫面。在圖 3-18每個圖中，左上角的影像是攝影機目前拍攝到的畫面，左下角的影像為第一張畫面轉正後的結果，而右上角的影像則是影像轉正並合成擴大後的結果，其中紅色的線段是我們在每張畫面中追蹤到的二個主要角點的連線。由第 20 張畫面（即圖 3-18 (b)）後，我們可以看到在圖中右上角的影像其周圍環境因合成而逐漸擴大，在第 60 張畫面時（即圖 3-18 (d)），左上角影像中的物體已超出攝影機的範圍，但是因為合成過去影像，所以右上角的影像中物體仍能保持完整的形狀。一直到第 120

張畫面（即圖 3-18 (f)），物體已完全超出攝影機拍攝的範圍，但是因為合成擴大的效果，所以物體仍保持完整形狀，但受累加的誤差影響，物體有些許變形。此一合成擴大影像，大致可以提供駕駛者一個完整且清楚的環境俯視圖，讓駕駛者在倒車停車時可以更為快速且容易。



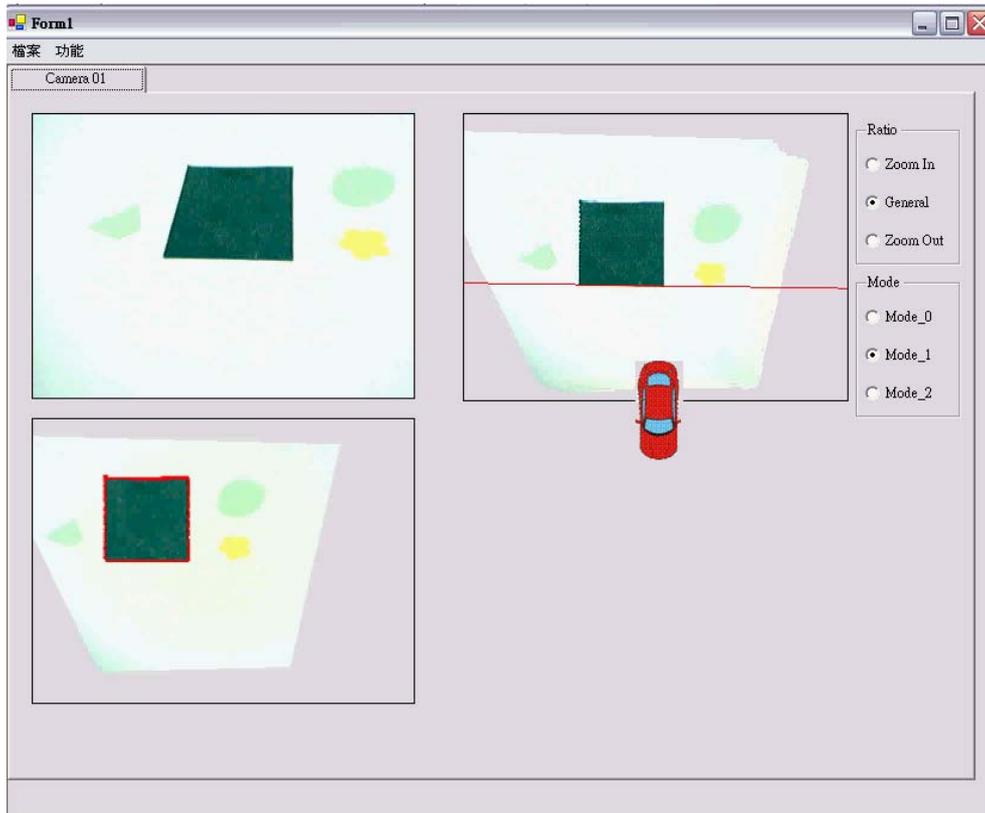


(a) Frame01。

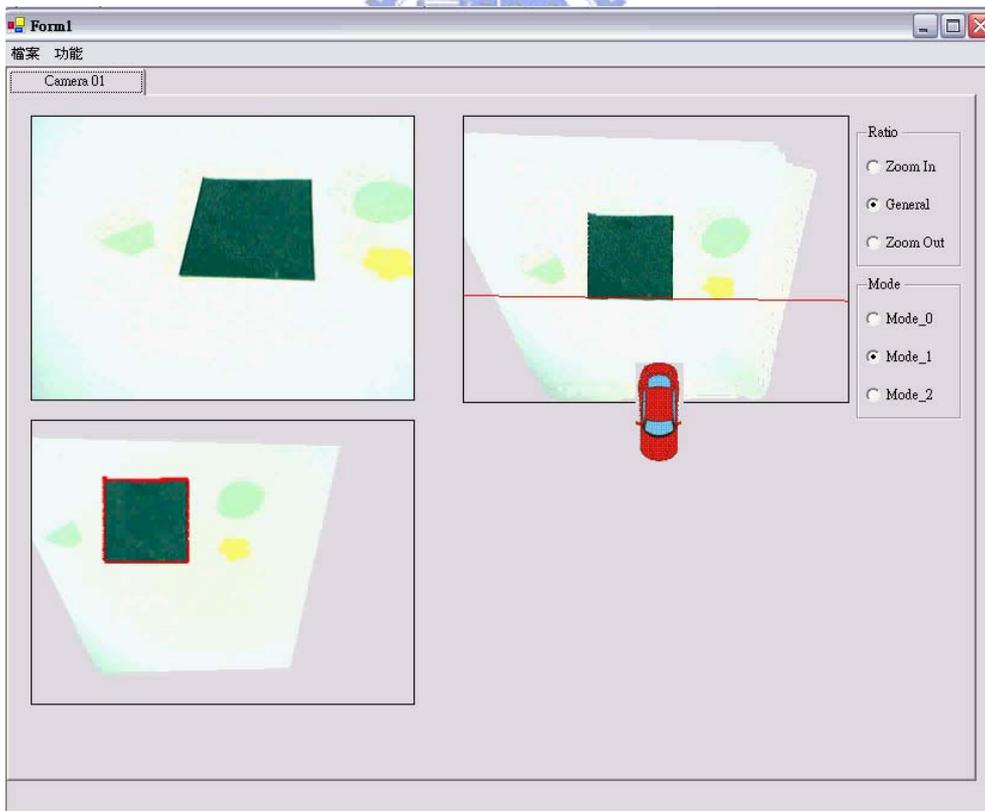


(b) Frame20。

圖 3-18 擴大影像合成之實例。(a) ~ (f) 分別為實驗結果 video 中第 1、20、40、60、80、100 和 120 張 frame。

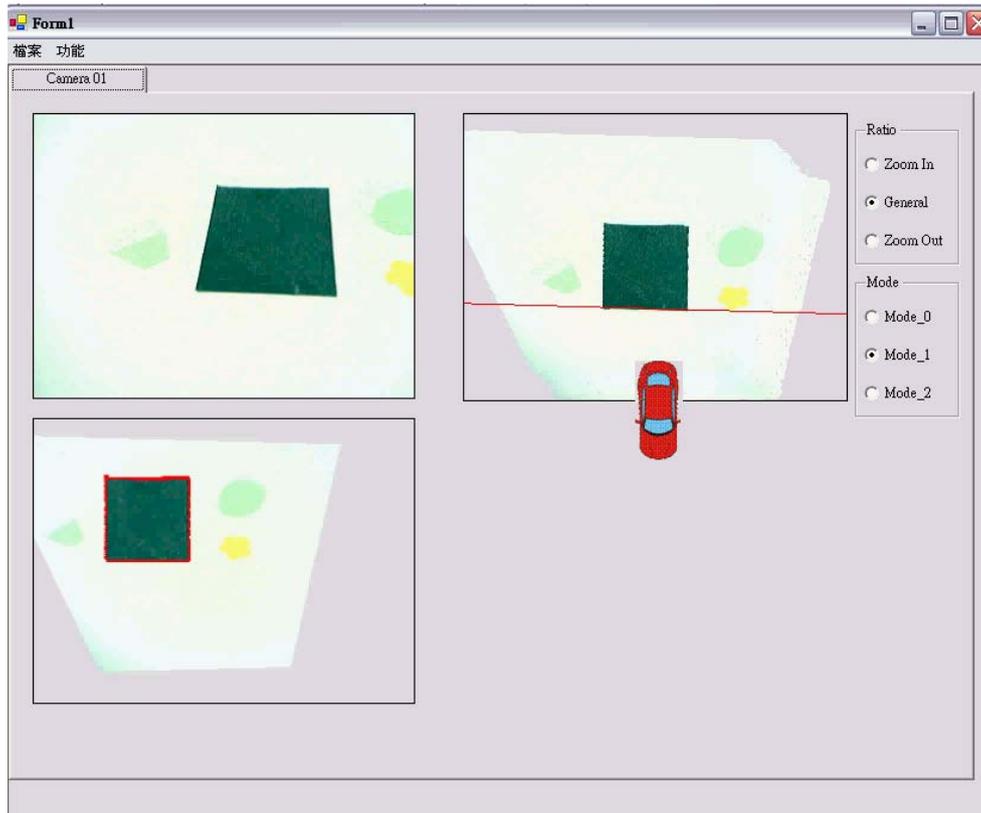


(c) Frame40。

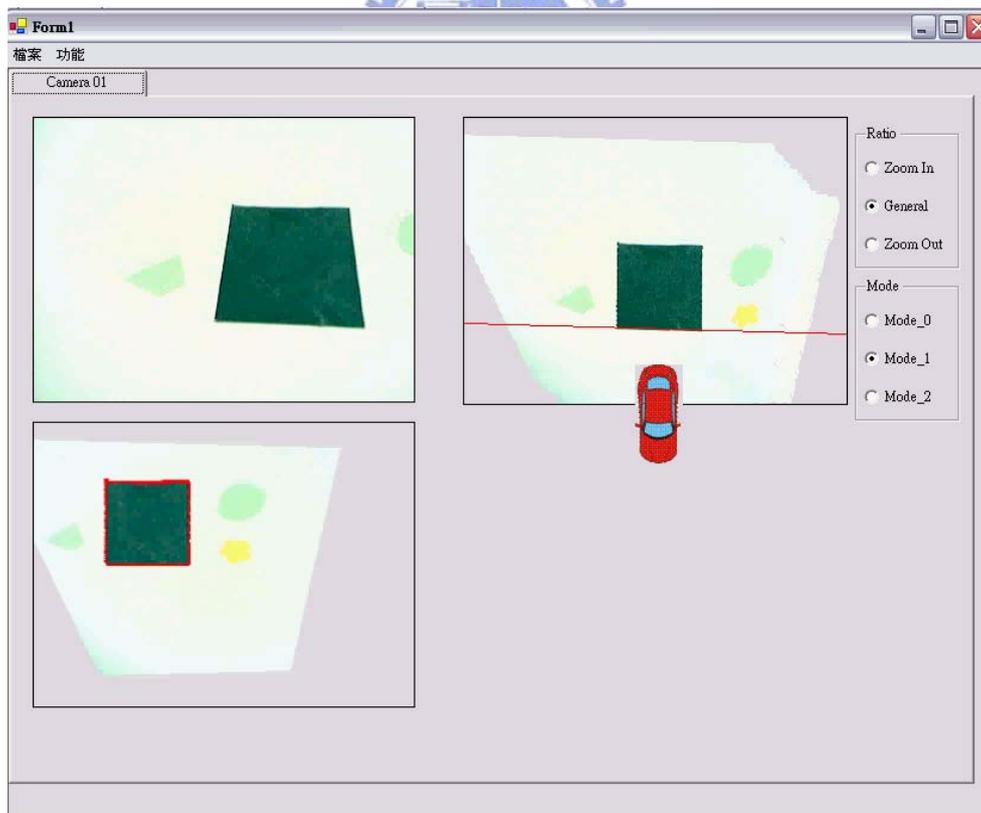


(d) Frame60

圖 3-18 擴大影像合成之實例(續)。(a) ~ (f) 分別為實驗結果 video 中第 1、20、40、60、80、100 和 120 張 frame。

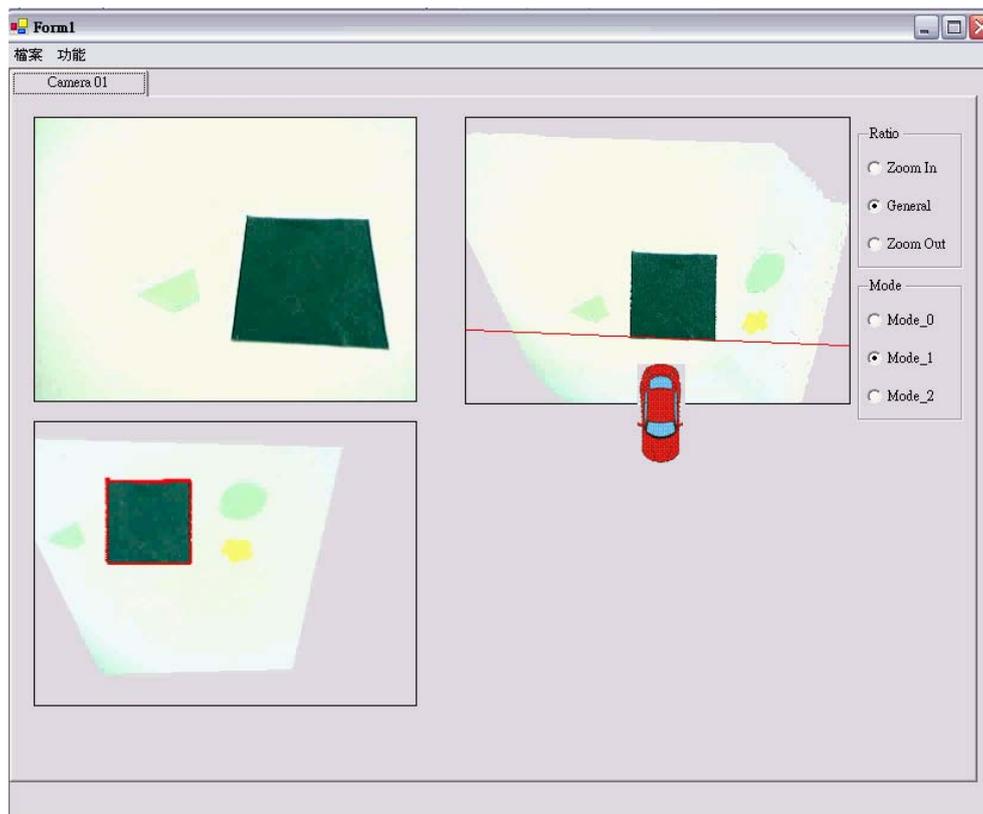


(e) Frame80



(f) Frame100

圖 3-18 擴大影像合成之實例(續)。(a) ~ (f) 分別為實驗結果 video 中第 1、20、40、60、80、100 和 120 張 frame。



(f) Frame120

圖 3-18 擴大影像合成之實例(續)。(a)~(f) 分別為實驗結果 video 中第 1、20、40、60、80、100 和 120 張 frame。

3.5 總結

在本章節中，我們提出了一個較為簡單且快速的方法找出影像中的角點特徵點，即先使用 Sobel 運算子偵測出影像中的邊，再利用 PCA 找出每條邊所對應的直線方程式，以找交點的方式擷取出角點特徵點。之後藉由角點特徵點的追蹤，求出連續影像間的旋轉與平移，進一步完成影像的合成。其實在電腦視覺的領域中，偵測特徵點的方法還有很多，因此我們還做了其它特徵點擷取演算法，以找出連續影像間的對應關係，並進行影像的合成。所以我們將在下一章中介紹另一種我們實作出來擷取特徵點的演算法，就是目前常用的偵測角點特徵點的方法-Harris 特徵點擷取演算法。

第四章 其它擷取特徵點的方法

在上一章中，我們利用影像中邊的資訊，以邊與邊的交點來擷取出角點特徵點，但目前特徵點的擷取還有另一個方向，即是分析影像中的色彩或灰階值來計算梯度 (gradient)、曲率 (curvature) 以擷取特徵點，例如 Moravec[12]、Harris[13]、Susan[14] 所提出的演算法。在此我們決定另外實作目前常用的 Harris 特徵點擷取演算法擷取出角點特徵點，並希望利用在上一章中所介紹的影像合成的方法，將影像環境接圖，以配合在不同環境下可能需要不同特徵點擷取演算法的情況。下面各節中，我們將詳細介紹 Harris 角點特徵點擷取演算法及實作情形。

4.1 Harris 角點特徵點擷取演算法

Harris 角點特徵點擷取演算法是由 Harris 於 1988 年所提出，而在此演算法中，最主要的概念是 1980 年 Moravec 根據觀察局部區域中灰階變化的特性，再加以延伸並改進。因此在本章中我們先介紹 Moravec 的理論，接著再說明 Harris 特徵點擷取演算法。

Moravec 在 [12] 中提出，角點特徵點的擷取可以藉由觀察一個影像中局部的矩形區域，將這個矩形區域向影像中不同的方向做小幅度的移動，來瞭解矩形區域中灰階變化的情形，矩形區域的移動可以如圖 4-1 來表示。

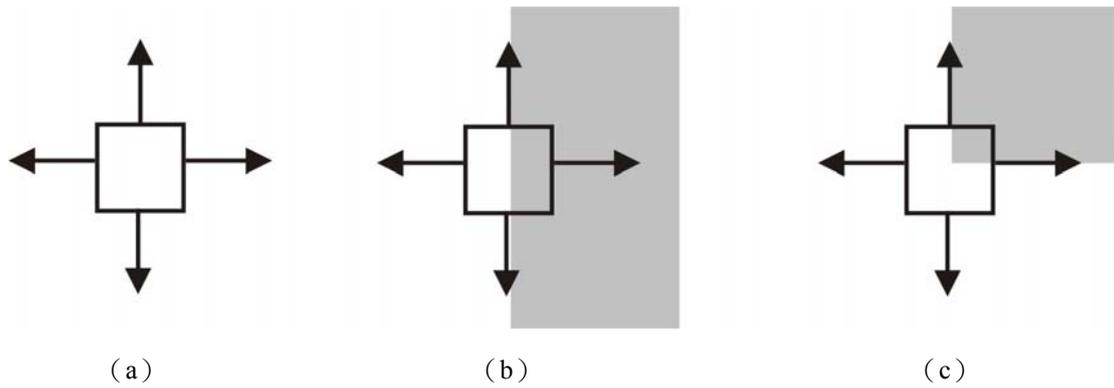


圖 4-1 矩形區域在影像中移動示意圖。(a) 為矩形區域在灰階變化平坦的影像區域中移動，(b) 為矩形區域在邊或線的影像區域中移動，(c) 為矩形區域在具有角點特徵點的影像區域中移動。

觀察矩形區域在影像中不同的區域中移所得到的結果之後，Moravec 提出以下三點結論：

1. 若矩形區域在灰階值變化平坦的影像區域中移動，則不管個矩形區域往哪一個方向移動，在矩形區域中的灰階變化都不會特別明顯，如圖 4-1(a)。
2. 若矩形區域在邊或線的影像區域中移動，則當矩形區域沿著邊或線方向移動時，灰階變化並不明顯；但是若矩形區域移動的向方與邊或線垂直時，則灰階變化就會相當明顯，如圖 4-1(b)。
3. 若矩形區域在具有角點 (corner points) 的影像區域中移動，則不管往哪一個方向的移動，都會造成在矩形區域中顯著的灰階變化，如圖 4-1(c)。

根據以上的觀察，前述矩形區域在各方向移動後其灰階變化的總和可用 (4.1) 式表示之：

$$E_{x,y} = \sum_{u,v} w_{u,v} |I_{x+u,y+v} - I_{u,v}|^2 \quad (4.1)$$

其中 x, y 分別表示在矩形區域在 x 方向以及 y 方向的移動量 (shift)； $w_{u,v}$ 表示所定義的矩形區域 (image window)，若是點的位置在此區域之中，則 $w_{u,v}$ 的值为 1，否則為 0； $I_{u,v}$ 為影像中點在 (u, v) 座標的灰階值。將 (4.1) 式以泰勒展開式展開後可得到 (4.2) 式：

$$E_{x,y} = \sum_{u,v} w_{u,v} [xX + yY + O(x^2, y^2)]^2 \quad (4.2)$$

其中 X 和 Y 可用下式來估算：

$$X = I \otimes (-1,0,1) = \frac{\partial I}{\partial x} \quad (4.3)$$

$$Y = I \otimes (-1,0,1)^T = \frac{\partial I}{\partial y} \quad (4.4)$$

(4.3) 式和 (4.4) 式分別是將影像 I 與 $(-1,0,1)$ 和 $(-1,0,1)^T$ 旋積 (convolution) 所得到的結果，也就是影像 I 在 X 方向和 Y 方向的梯度。得到了 $\frac{\partial I}{\partial x}$ 和 $\frac{\partial I}{\partial y}$ 的結果之後，假設 x 和 y 方向的移動量很小，忽略 (4.2) 式中的高次項之後，可改寫 (4.2) 式為：

$$E_{x,y} = Ax^2 + 2Cxy + By^2 \quad (4.5)$$

其中 A 、 B 、 C 定義如下：

$$A = \left(\frac{\partial I}{\partial x}\right)^2 w_{u,v} \quad B = \left(\frac{\partial I}{\partial y}\right)^2 w_{u,v} \quad C = \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) w_{u,v}$$

在 Moravec 的演算法中，若點的位置落在矩形區域中，則 $w_{u,v}$ 的值為 1，若點落在矩形區域外，則值為 0。但在 Harris 的觀點中，為了減少影像中雜訊的影響，便將 $w_{u,v}$ 以 Gaussian 函數來取代，即如 (4.6) 式：

$$w_{u,v} = e^{-(u^2+v^2)/2\sigma^2} \quad (4.6)$$

則將 (4.5) 式以矩陣的形式表示則為：

$$E(x,y) = (x,y)M(x,y)^T \quad (4.7)$$

其中 M 為一個 2×2 的對稱 (symmetric) 矩陣：

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

其中 A 、 B 、 C 如 (4.5) 式所定義。

由 (4.7) 式來看， E 可能得到的結果會與矩陣 M 有一定程度的關聯。在此令 α 、 β 為矩陣 M 的兩個特徵值 (eigenvalue)，在 Moravec 的想法中，所得到的 $E_{x,y}$ 越大表示不管矩形區域往哪一個方向移動，灰階變化皆非常明顯；而經由對 α 、 β 與灰階變化關係的分析，Moravec 認為有以下幾種情形：

- A. 若 α 、 β 值皆小，則表示此影像區域的灰階變化不明顯。
- B. 若 α 、 β 值為一大一小，則表示矩形區域沿著某一個方向其灰階變化較大，也就是在此影像區域中有邊或線的存在。
- C. 若 α 、 β 皆大，則表示矩形區域不管往哪一個方向移動其灰階變化皆很明顯，也就是在此影像區域中有角點的存在。

α 、 β 的關係如圖 4-2 所示。

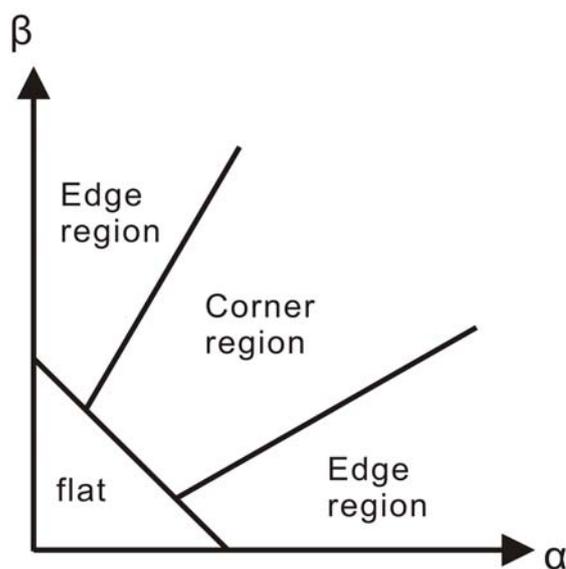


圖 4-2 特徵值 α 、 β 與影像灰階變化關係圖。

因此我們可以根據矩陣 M 的兩個特徵值 α 、 β 的大小，衡量出此影像區域為平坦區域、邊或角點特徵點的可能性。

4.2 利用 Harris 演算法擷取角點的實驗結果

根據Harris的角點特徵點擷取演算法，我們令矩形區域大小為 5×5 ，Gaussian 函數的變異數 σ 大小為 1.5， α 、 β 的臨界值為 2000，則所擷取出的角點特徵點如圖 4-3 所示。

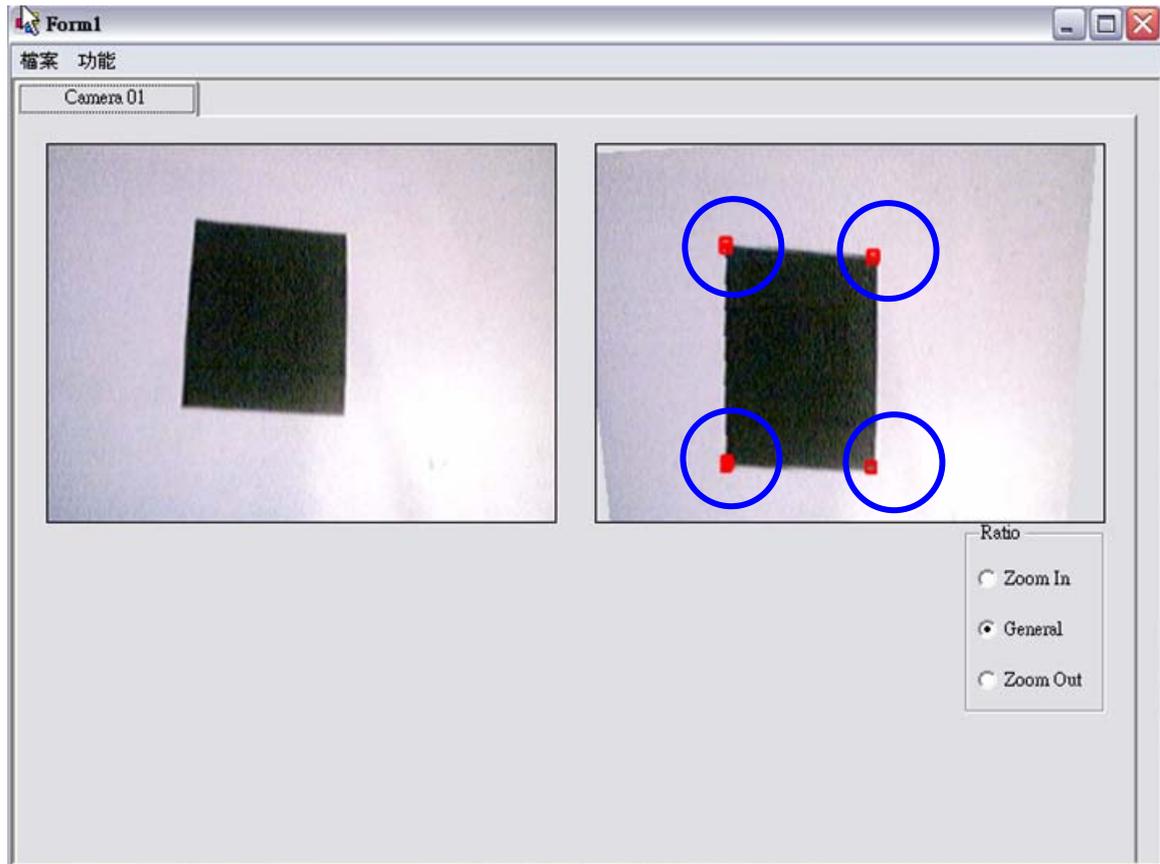


圖 4-3 使用 Harris 特徵點擷取演算法，特徵值超過臨界值 2000 的影像點。

在圖 4-3 中，我們可以明顯觀察到只要是角點或是在角點周圍的點（以紅色框格表示），因為其計算出來的 α 、 β 值都很大，所以都會被擷取出來。但是其實我們只希望擷取到剛好是角點的點（也就是停車標示的四個頂點），以利之後做影像追蹤與合成時的穩定性與正確性，所以在找到使用 Harris 的角點特徵點擷取演算法中特徵值 α 、 β 大於臨界值的點後，則會再針對該點其 5×5 範圍內的點，做特徵值最大值的篩選。篩選方法的示意圖如圖 4-4 所示。最後經過最大值篩選的結果圖如圖 4-5，我們可以發現已經準確的擷取到影像中停車標記的四個頂點了。

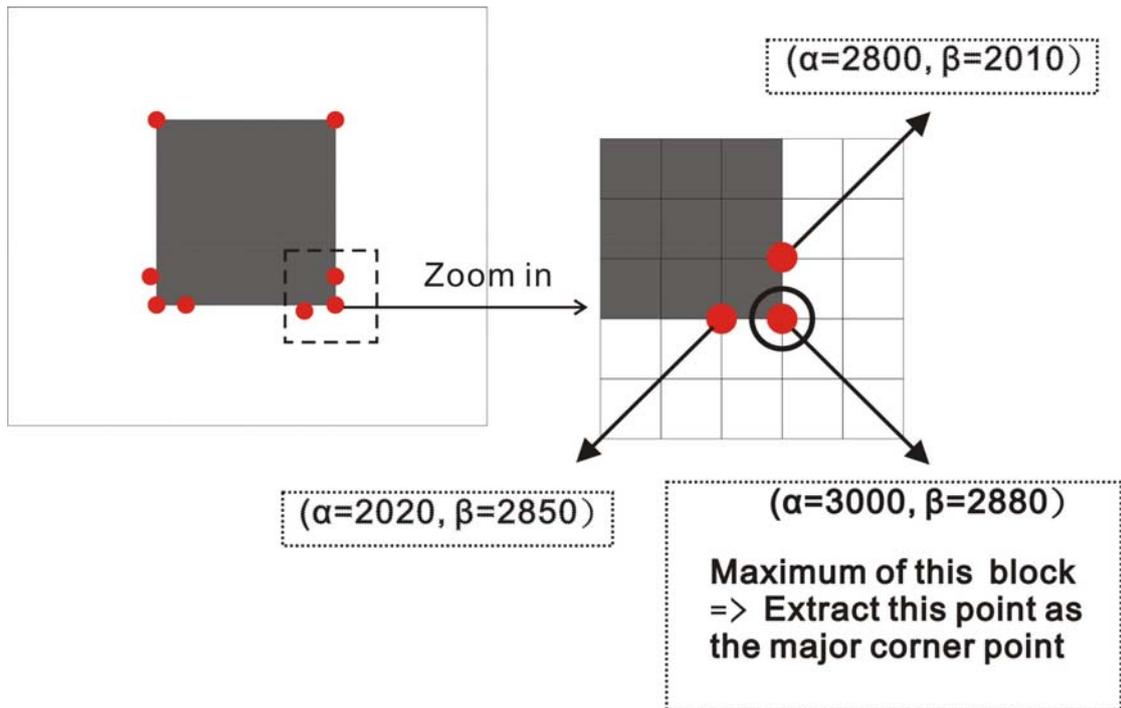


圖 4-4 主要角點篩選方法示意圖。

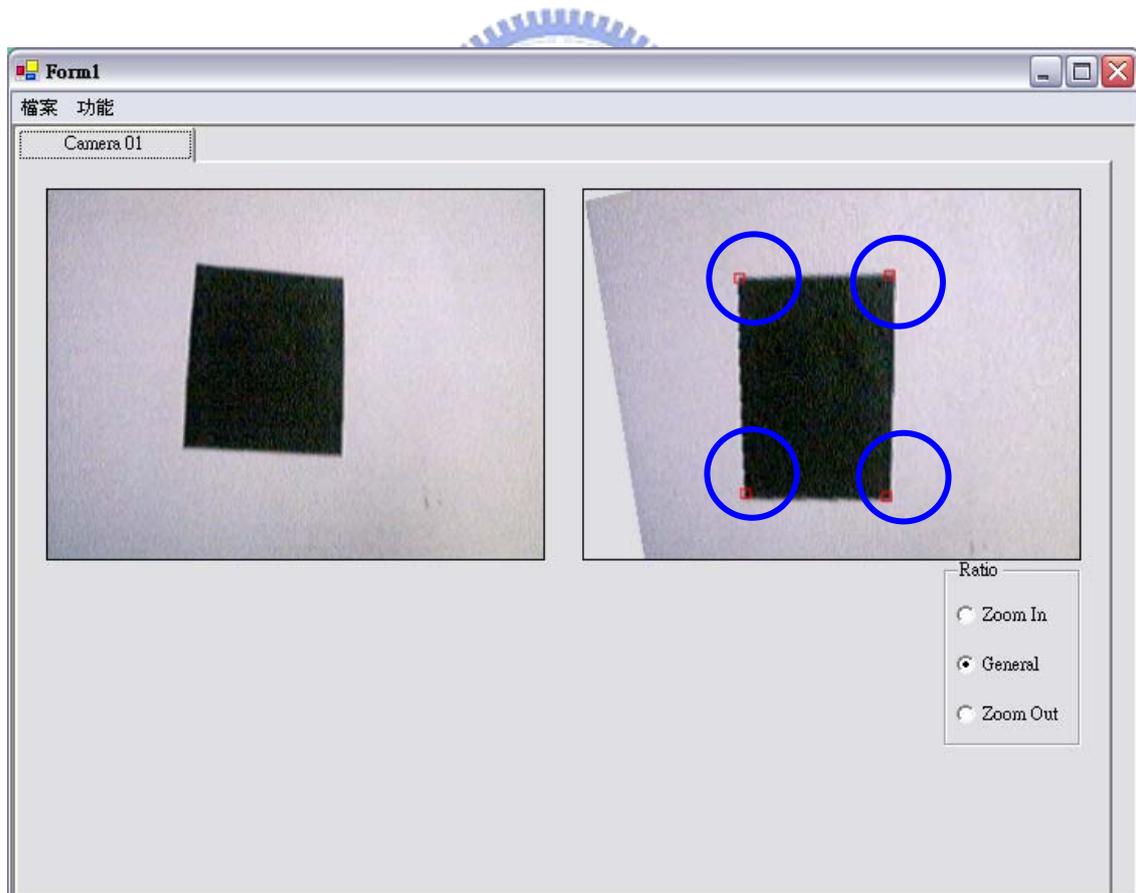


圖 4-5 使用 5x5 block 做區域最大值篩選。

4.3 總結

在本章中，我們使用 Harris 角點特徵點擷取演算法擷取出影像中停車標記的四個角點，之後我們可以再依據第 3.3 節和第 3.4 節的方法對影像進行追蹤與合成。在第三章中，我們擷取特徵角點的方法是先偵測影像中的邊，再利用邊的資訊得到影像的角點，所以當影像環境地面上出現線條花樣時，我們所擷取到的角點會受環境影響而有誤判的可能，但若是採用 Harris 演算法擷取角點特徵點的話，就較不易被線條所干擾。不過使用 Harris 演算法時，在擷取連續影像中對應的角點特徵點時可能會有些微誤差，導致追蹤點的準確度較低，合成時就會出現問題，所以我們可視不同的情況，使用不同的特徵點偵測演算法，以達到最好的影像合成結果。



第五章 結論與未來工作

在本篇論文中，我們提出了一個具有視訊畫面的倒車系統，我們先利用了影像平面與三維參考平面間的平面投影轉換關係，將車後攝影機拍攝到的連續影像轉成擬真的鉛直俯視角度的畫面，讓駕駛者可以根據俯視圖更清楚場景和車子的相對關係。之後再利用特徵點的偵測與擷取，找出連續影像間的對應關係，將過去影像和目前影像合成，銜接出一個完整的環境圖讓駕駛者參考。而且在我們的系統中，並不需要使用到其它特殊的裝置設備，如 Gyro 或其它特殊感應器，所以不會導致駕駛者的負擔。值得注意的是，本實驗假設的環境地面，是單純沒有任何圖樣存在，只有一塊停車標記在上面，這是因為若是環境中存在線條或過多雜點時，會導致系統擷取角點特徵點時出現問題。雖然使用 Harris 演算法可以應用在環境地面上有線條的場景，但若是環境地面上有許多構成格紋或角的圖樣時，利用 Harris 演算法擷取連續影像中對應的特徵點，可能穩定性和準確性都會有問題。所以目前我們仍繼續針對具有雜點跟格紋圖樣的複雜地面進行研究，希望可以準確擷取出格紋上的線，再利用線的交點當作特徵點，找出影像對應關係，以完成影像的合成與接圖。此外，之後若能將各種擷取特徵的方法整合，使系統可以根據不同的停車格環境使用不同特徵點擷取來進行倒車輔助，則將使系統可適用於更多不同的路面狀況。

參考文獻

- [1] K. Fintzel, R. Bendahan, C. Vestri, S. Bougnoux, S. Yamamoto and T. Kakinmi, “3D vision system for vehicles,” in *Proc. 10th IEEE Intelligent Vehicle Symposium*, Columbus Ohio, pp. 174-179, 2003.
- [2] K. Fintzel, R. Bendahan, C. Vestri, S. Bougnoux, and T. Kakinami, “3D parking assistant system,” in *Proc. Intelligent Vehicles Symposium*, pp. 881-886, 2004.
- [3] J. Shi and C. Tomasi., “Good feature to track,” in *Proc. Conference on Computer Vision and Pattern Recognition*, USA, pp. 593-600, 1994.
- [4] <http://www.aisin.com/index.html>
- [5] 吳瑞鴻, 劉景富, “行徑軌跡指示之倒車輔助系統開發,” *車輛研測資訊* pp. 2-7, 2006.
- [6] S.J. Chang, “Design and implementation of fuzzy parking control methodology of car-like mobile robots,” *Ph.D Dissertation*, Department of Electrical Engineering, National Cheng Kung University, Taiwan, R.O.C, 2003.
- [7] DENSO The 38th Tokyo Motor Show 2004, <http://www.denso.co.jp/motoshow/2004>
- [8] Y. Suzuki, M. Koyamaishi, T. Yendo, T. Fujii, and M. Tanimoto, “Parking assistance using multi-camera infrastructure,” *Intelligent Vehicles Symposium*, pp. 106-111, 2005.
- [9] D. Forsyth and J. Ponce, *Computer Vision A Modern Approach*, Prentice Hall, 2002.
- [10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- [11] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Elsevier Academic Press, 2003.
- [12] H. P. Moravec, “Toward automatic visual obstacle avoidance,” in *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pp. 584-587, 1977.

- [13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvery Vision Conf. Manchester*, pp. 189-192, August 1988.
- [14] S.M Smith and J. M. Brady, "SUSAN: A new approach to low level image processing," *International Journal of Computer Vision*, pp. 45-78, 1997.

