

一、緒論

1.1 研究動機與目標

近幾年來在台灣及國外，智慧型運輸系統 (Intelligent Transportation Systems, ITS) 不論在研究發展或運用推廣等各方面，均有長足的進步。尤其目前資訊與通訊技術的不斷進步，使得 ITS 相關系統更加實用並且符合成本效益。ITS 之所以廣受重視，可以從 ITS 將對社會及個人產生的巨大影響，來了解其重要性：包括節省運輸所需的時間及燃料、提高交通運輸安全，以及更加人性化的管理等。

何謂 ITS？ITS (Intelligent Transportation System) 乃是應用先進的電子、通信、資訊與感測等技術，以整合人、路、車的管理策略，提供即時(real-time)的資訊而增進運輸系統的安全，效率及舒適性，同時也減少交通對環境的衝擊。本篇論文即希望利用現有視覺監控系統技術 (Visual-based surveillance system techniques)，來發展 ITS 中一個相當重要的應用：自動化停車場管理系統。

現今的社會，由於車輛數目遽增，使得停車的問題日益嚴重，停車場扮演了一個相當重要的角色，但現今停車場多需依賴管理員或機械式柵欄在出入口取票、收費，往往造成許多車輛大排長龍等待進入或等待繳費出去，非常的沒有效率，且安全上也沒有獲得良好的保障，不時有偷車或是車輛被破壞的情形發生，本篇論文即在解決以上問題，提供一個無人管理，卻有效率，迅速，而安全的自動化停車場管理系統。

而現今視覺監控系統(Visual-based surveillance technique)，則大致可分為三個部份：資訊收集、資訊傳輸及資訊處理。資訊收集：感測技術，主要是利用攝影機抓取欲監控場景的影像。資訊傳輸：通信技術，利用無線及有線的通信技術可迅速地將大量影像資訊傳送到電腦端做處理。資訊處理：電腦技術，將接收到的影像資訊透過電腦視覺及影像處理技術的協助，擷取出有用的資訊。本篇論文所著重的，在於資訊處理部份，即將接收到的影像資訊，透過本篇論文發展出的技術，提供停車場管理者與使用者最佳的協助。

為達成這個目標，首先必需能找出場景中待追蹤的目標物。也就是必須擁有前景/背景分離的能力達到偵測場景變化的能力。在進行前景/背景分離的同時，為为了提高監控的準確度我們必須同時考慮

雜訊的干擾、光照變化、陰影等所造成的影響，並盡可能的排除場景中無意義的變化及適度的更新我們的背景。在能偵測場景的變化後，我們開始持續追蹤場景中變化的物體，以分析其行為模式，達到監控的目的。

1.2 相關研究

而在進行自動化監控系統時，首要的條件就是能將前景(目標物)由目前影像中分離出來。現有的前景與背景分離的方法，大致上可以分為下列三種:

(1) Temporal Differencing :

或又稱做 **Frame Differencing**。本方法與 **Fix background subtraction** 類似，但不使用事前拍攝的場景作為參考背景。將時間軸上連續或相隔一定距離的兩張影像作相減的動作。

本方法的優點：不易受場景光照變化的影響。缺點：當前景的人物停止不動時，用前後時間攝得的影像相減，可能無法找出變化，必須改採靜態背景相減的方式偵測之。另一方面，若攝影機不是固定的，則當攝影機與場景中的目標物同時都在移動，由時間軸上相鄰的影像相減，會使前景與背景皆有差異而難以分離。此時則必須做共同靜態背景的疊圖定位(**Background Registration**)。在扣除背景後，才能找出前景物體。

(2) Fix Background Subtraction :

這個方法是使用事前拍攝完成的場景作為參考的 **Background Image** 和目前利用攝影機拍得到影像 **Current Image** 作相減的動作，所得到的差異部分便是可能需要監控的目標物。此法的優點：設計簡單且運算量低。缺點：背景內容一旦改變，就必須重新拍攝。

我們系統中為減低計算量、儘可能的降低運算所需時間。採用的方法為靜態影像相減的方式，所以必需考慮亮度不同時所造成的影響。

接著，我們希望對所偵測出的前景開始進行追蹤，現有追蹤方法的類型，大致上可以分為下列二種:

(1) Matching Tracking : [11]

比對追蹤是對在偵測步驟中不同時間點所得出來的前景，計算兩個個體間彼此的關聯程度，如果關聯程度越高，亦即表示兩者越有可能是相同物體，如此比對在不同時間點的前景，希望接藉此找出物體移動的軌跡，以達到追蹤的目的。如我們可將關連程度定義為：任兩個不同時間點所得到前景之距離，假如某兩個不同時間點所得到的前景其距離最短且在合理範圍內，我們就說這兩個不同時間點所得到的前景為同一個物體。

(2) Prediction Tracking :

預測在追蹤系統中是一個相當有用的方法，其做法是找出物體的移動習慣，譬如球在被拋出所做出的拋物線運動，火箭或飛機在天空飛行的移動方式或是車輛在車道上所做的運行，都存在一個規則而預測變是希望找出物體移動的規則，利用方程式表現出來，進而去預測出物體下一個時間點應該會出現的位置。Kalman Filter 即為其中一種 Prediction-based 為主的方法。

我們的系統將此兩種方法一起使用，對於每一個前景物體，我們先用 Prediction Tracking 法去預測下一個時間點可能的位置，再根據我們實際上所偵測到的前景，套用 Matching Tracking 法，找出最有可能的物體，同時修正我們的預測，以達到最正確的追蹤。

國內外相關研究如下：

Collins 在卡內基美濃大學的研究團隊，提出一套完整的監視系統 (VSAM, Video Surveillance and Monitor)，利用各種不同的攝影機，分別架在不同的位置，透過網路及分散式系統連接，並能自動偵測、追蹤、辨識大範圍的人與車輛，而攝影機之間也能夠相互溝通，搭配資料庫系統作紀錄

Stenfan Huwer[4] 在即時監視系統中，提出 ADC (Adaptive Detection Change) 結合背景相減法能夠找到完整影像及時間差異法的高光源變化容忍(及背景不斷適時更新)的優點，做法為分別對輸入影像作背景相減法及時間差異法，找出各自目標物影像後再取交集影像。

1.3 研究流程概述

依停車場管理的流程可以分為車輛辨識，車輛偵測與追蹤及車輛監控三個階段，每一階段皆有各自的執行重點並分述如下：

(1) 車輛辨識階段

- (a) 偵測車輛之進入：預先取出一張無車輛進入的圖像，也就是純背景圖，接著對於時間點位於背景圖之後的圖像，不斷的與純背景圖相減以偵測是否有車輛即將要進入停車場，當偵測到的變化度大於所設定的臨界值時，也就是車子完全進入攝影機(停車場)後，開始進行車牌辨識。
- (b) 車牌辨識：對於進入停車場的車輛進行車牌辨識，包括以下步驟：車牌定位、字元切割與車牌 OCR (Optical Character Recognition) 辨識。

(2) 車輛偵測與追蹤階段

- (a) 車輛偵測與追蹤：偵測進入停車場內的車輛，並持續追蹤該車輛，以便隨時掌握該車輛在停車場內的狀態。

(3) 車輛監控階段

- (a) 監控車輛有無正確停放在停車格內：偵測該車輛是否已正確停入停車格內、有無超過停車格及是否沒停在停車格內而直接停在車道上等。
- (b) 監控車輛是否發生碰撞：監控在停車場內移動的車輛，是否有碰撞到其他早已停放在停車場內的車輛。
- (c) 監控異常情況：車主將車輛停入停車場格後，是否就待在車上而沒有下車；車主前來取車時，是否進入車內後，卻遲遲未將車開離。
- (d) 監控車輛是否要離開：高空攝影機偵測到有車輛離開停車格後，便通知資料庫管理系統作處理，資料庫管理系統會做適當的處理，並釋放該停車格，而免除了出口再作一次車輛辨

識的麻煩。

1.4 論文組織

本論文除本章為“緒論”外，在第二章中，本文將對停車場管理流程作一詳細描述。此外，也將討論如何建置停車場資料庫管理系統。

在第三章中，介紹如何利用 VECTOR MODEL 觀念來進行場景變化偵測。同時提出如何隨時間更新背景影像，以及當無法取得純淨的背景圖，也就是背景影像中含有所謂假背景時，如何利用目前的影像來還原真正的背景。最後，是本篇文章的核心，前景物體的追蹤，解決了在追蹤常遇到問題，包括基本前景物體追蹤、前景發生交錯時的追蹤及在交錯的過程中如何判定誰遮蔽了誰。

在第四章中，介紹停車場事件偵測與處理。包括如何偵測各種停車場事件，及相對應的處理方式。

第五章中，則為整合所有研究程序所進行的各項實驗所採用影像及數據，及最後成果的展示。

第六章為此次研究的結論與未來可以改進的空間的討論。

二、停車場自動監控系統

2.1 系統架構

此系統提出一個從連續影像序列中自動監控與管理停車場的技術。此系統由四個主要部分組成：影像輸入、物件偵測、物件追蹤、車輛監控，每一部份概述如下：

影像輸入：此系統的輸入是在一般的停車場下，用一般數位攝影機所拍攝的連續影像序列，儲存成 24bit 的 RGB 連續影像，之後並轉成灰階圖片作處理。

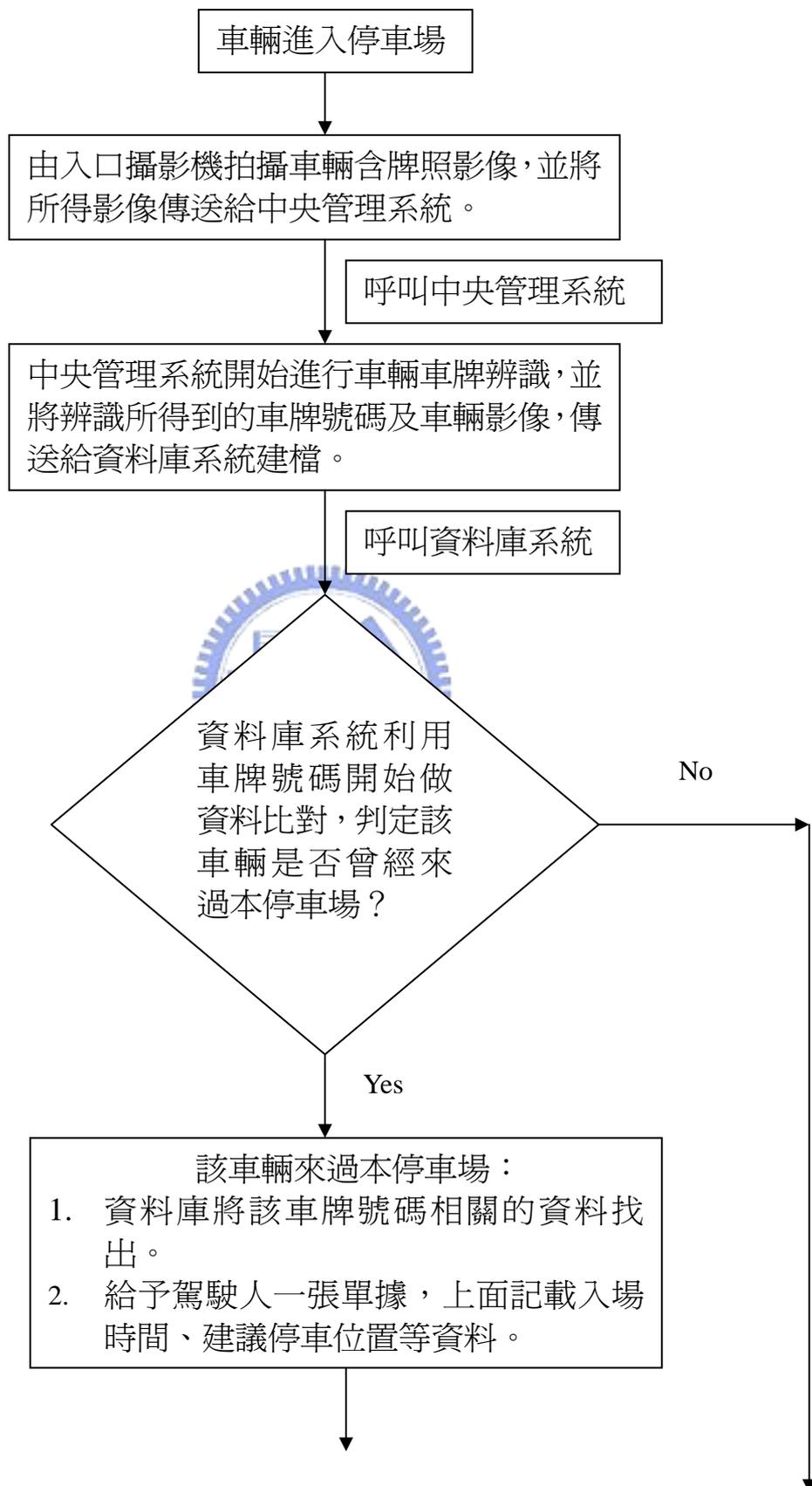
物件偵測：首先利用 VECTOR MODEL 法將與背景不同的點切割出來，再利用編號演算法將這些點所圍成的許多不同區塊編成不同的編號，並將大部分的小雜訊濾除，同時將剩下較大的區塊設為車輛，較小的區塊設為人，最後處理因為光線所造成的影響，並不斷的去更新背景。

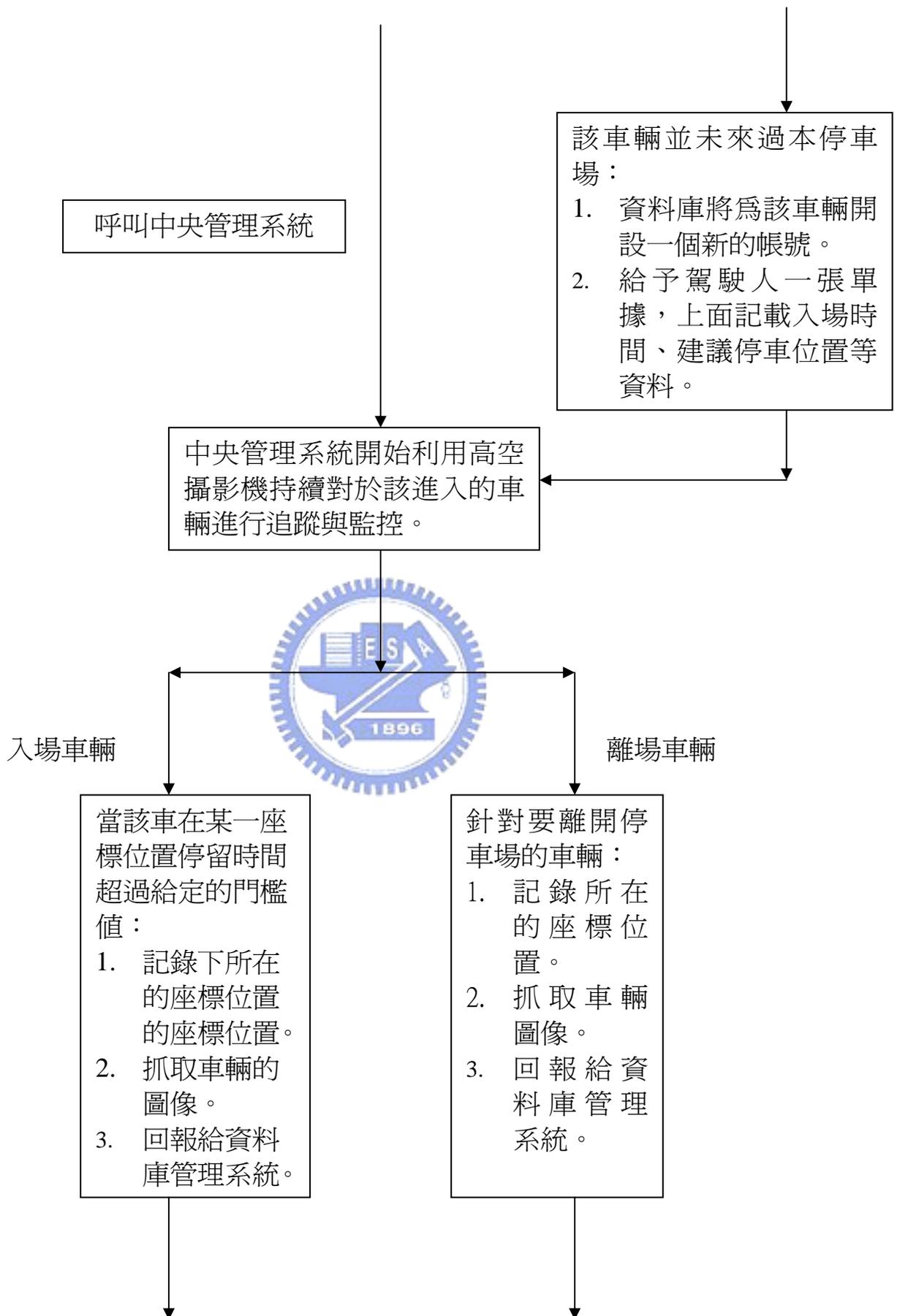
物件追蹤：將我們所偵測出的前景(車輛)，利用比對追蹤法及預測追蹤法，正確的追蹤在不同的時間點的相同物件。

車輛監控：持續追蹤偵測出來的車輛，以了解其行為模式，包括是否正確停放在停車格內或是違規停放。

資料庫管理系統：記錄了停車場資訊、使用者資訊及停車場事件紀錄。停車場資訊包括：停車場停車平面圖、停車格使用狀況；使用者資訊則包括：車牌號碼資訊、進場與離場時間、與尚餘金額等等；停車場事件紀錄則包括：發生違規事件後資料庫會出現警示狀況與處理情形等。

停車場管理流程架構圖





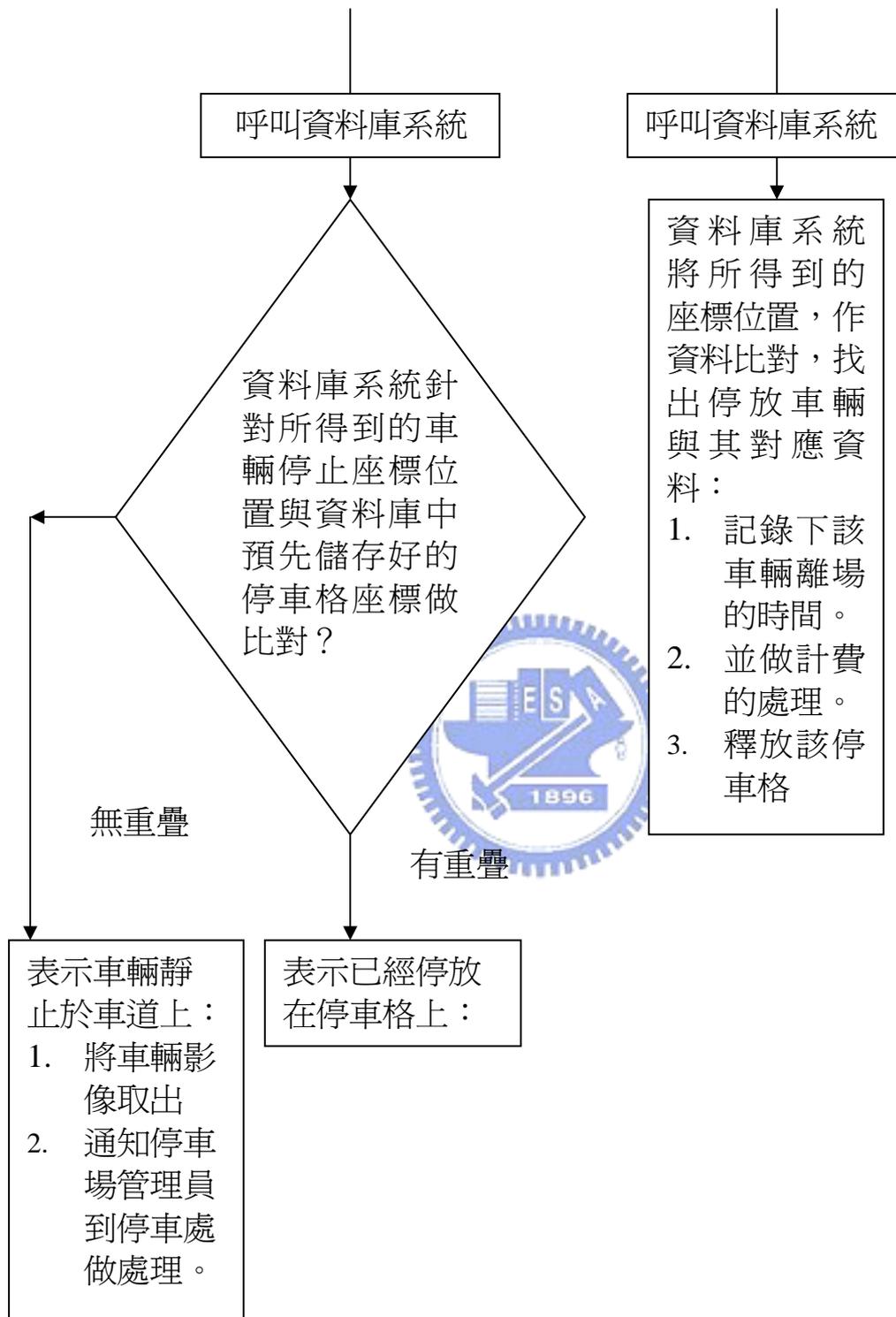


圖 2.1 停車場管理流程圖

2.2 資料庫系統簡介

本部份的關連性資料庫利用MySQL (<http://www.mysql.com/>) 建置，並以PHP (<http://www.php.net/>) 來撰寫管理介面以及與處理程式的溝通介面。

2.2.1 資料庫部份

資料庫的主要資料表及其關係如下圖：

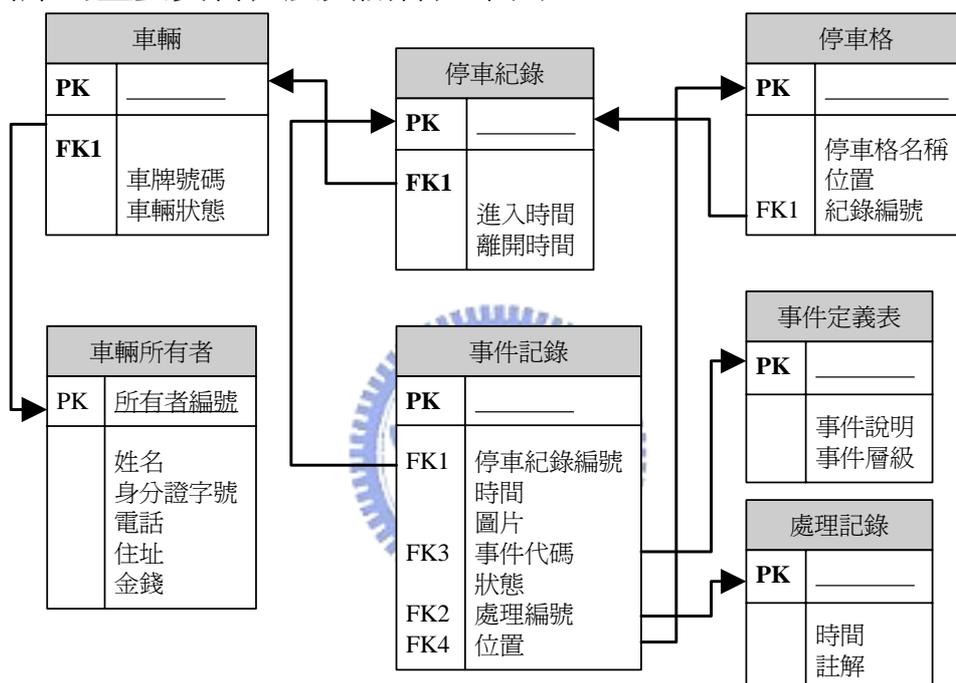


圖 2.2 資料庫的主要資料表及其關係

各資料表的功能如下：

▲ 車輛所有者

紀錄車輛所有者的資料，並將其停車預放金額儲存與此，已達到自動扣款的功能。

▲ 車輛

儲存車輛的車牌、狀態。如有需要，也可進一步儲存車輛廠牌、顏色、車型等資料。

▲ 停車紀錄

用以儲存停車的紀錄。從車輛進入到車輛離開，計做一次完整的「停車紀錄」；期間可含有數個「事件記錄」。

▲ 事件記錄

所有接收到的事件，主要都將儲存與此；管理介面也會根據

車輛編號

所有者編號

此表格來通知管理員。

- ▲ 處理記錄
管理員可將事件處理做紀錄，其內容儲存於此資料表中。
- ▲ 事件定義表
定義事件編號與事件內容、層級。可根據各系統需求做修改。
- ▲ 停車格
儲存停車格的相關資訊與目前狀態。

2.2.2 事件觸發介面

而事件觸發的介面，也是網頁的形式。程式只需在偵測到事件時，呼叫此介面即可。其參數傳遞的格式如下：
`enent.php?NUM=車牌號碼&PIC=照片&EVENT=事件代碼&LOC=位置`
各事件的詳細說明請參考下節。

2.2.3 管理介面

管理介面如下圖：



圖 2.3 資料庫管理介面

分為三個區域：

甲、選單區

位於左上角的框架頁，有「停車場資料」、「用戶資料」、「車輛資料」、「停車紀錄」、「事件記錄」等選項；點選後，主畫面將會進入所點選的功能頁。



圖 2.4 資料庫選單區

乙、 事件區

位於左下角的框架頁；每秒鐘會自動更新，以檢查停車場狀態（車輛數、車位數量），如發現不正常狀態，會顯示意外事件，並發出聲音警示。點選事件後，主畫面將進入事件處理的頁面。



圖 2.5 資料庫事件區

丙、 主畫面

位於右方的框架頁中，為管理介面主要呈現資訊、修改資料的畫面。

管理介面的各項功能：

甲、 停車場資料

主畫面的預設頁面，每 3 秒會自動更新，以確保停車格資訊正確。以圖形化的方式，顯示目前停車場內的停車格狀態。（如下圖）



圖 2.6 資料庫停車格平面圖

乙、 用戶資料

顯示、編輯用戶的相關資料。

丙、 車輛資料

顯示、編輯車輛資料。

丁、 停車紀錄

檢閱停車紀錄。



戊、 事件記錄

檢閱事件記錄、提供事件照片讓管理員處理事件。

此外，也提供車主利用車牌或停車編號找車的功能，並以圖形的方式呈現給車主(如下圖)。

停車場資料

停車編號： 1

車牌號碼： 1

進入時間： 2004-07-09 00:08:35

您的停車格為下圖有顏色的部份

01	02	03	04	05	06	07	08	09	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27						
28	29	30	31	32						

圖 2.7 車輛進入停車場時，分發給使用者的停車資訊單據。



三、車輛偵測與追蹤演算法

3.1 車輛偵測[6]

在傳統的停車場監控系統中，機器只負責拍攝影像，而檢查是否有車輛進入場景與判斷其行為的工作則由人眼執行，如此意謂著必需付出額外的人力來負責處理這項工作。因此本文希望能讓停車場監控系統具有偵測車輛與處理事件的能力，達到提高效率、自動化與智慧化的目的。

車輛偵測主要的目的是偵測新進入停車場的車輛，亦即將攝影機所拍攝到影像中的前景部份(車輛)與背景部份(固定不變的場景)分離開來。傳統的作法是採用一張事先拍攝、乾淨的背景圖與事後所拍得影像的每個像素進行比對[7,8]。所得到差異的部分便是可能需要監控的前景物體。此法的優點在於簡單且運算量低，缺點則是易受雜訊的影響，而使得相減所得的差異雜亂無章，且當場景的光照不同時，直接相減會造成整張影像都有差異，無法判斷何者是前景何者是背景。這對將來停車場監控系統在追蹤車輛時會發生極大的困擾。以下即為再兩個不同時間所拍攝的影像：



圖 3.1 原始影像與目前變亮影像

利用傳統固定背景相減的結果



圖 3.2 目前影像與背景影像每一像點相減差值 > 50 設為前景

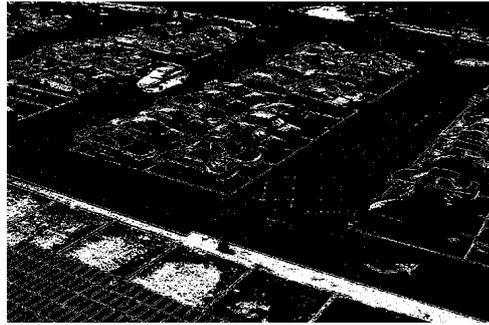


圖 3.3 目前影像與背景影像每一像點相減差值 > 70 設為前景

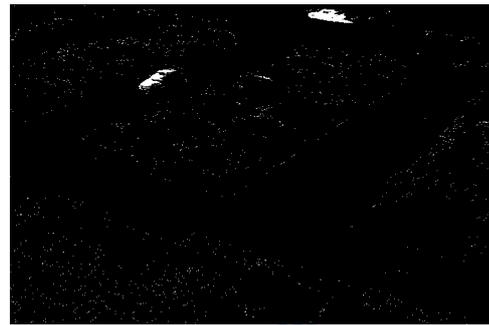


圖 3.4 目前影像與背景影像每一像點相減差值 > 90 設為前景

由以上結果可以得知，傳統的固定背景相減在遇到光線明顯變化的時候，不論怎麼調整臨界值都無法完整切出前景物體，因此傳統的固定背景相減並不適用在全天候監控的停車場內，因此我們採用以下方法。

3.1.1 以 VECTOR MODEL 判斷兩張影像的變化

為了解決上述方法所產生的問題，本文採用的方法是以 VECTOR MODEL 為基礎的偵測方式，該方法在[1,13]有詳細的介紹，在此僅將該方法簡單的介紹如下：

首先，將影像上的每一點鄰近 $n \times n$ 大小範圍內的像素之亮度值排列成一個 vector，以一個 3×3 的例子如下圖 3-1 所示：

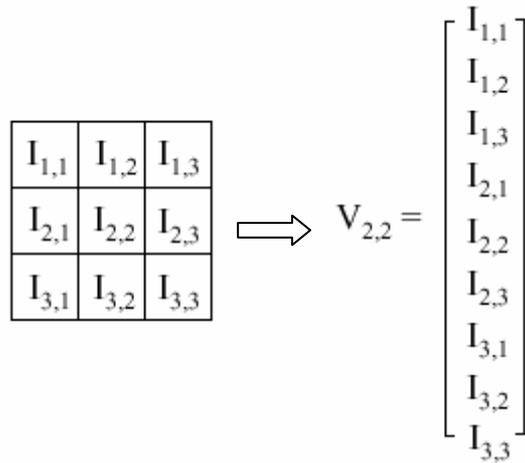
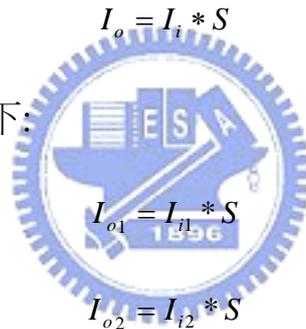


圖 3.5 VECTOR MODEL 示意圖

根據光照的理論，一物體在某一點的亮度值(intensity) I_o ，大致上可以視為光照量 I_i 和物體表面反射係數 S 的乘積[2]，即：

$$I_o = I_i * S$$

今在兩不同的光照條件下：



$$I_{o1} = I_{i1} * S$$

及

$$I_{o2} = I_{i2} * S$$

對同一物體的同一點而言，反射係數 S 視為相同，因此可以導出下面的關係：

$$\frac{I_{o1}}{I_{o2}} = \frac{I_{i1}}{I_{i2}} = \alpha$$

所以該點的亮度值與光照量成等比例。若同時考慮兩張同一場景之相對應window內所有像素的亮度值的話， V_1 為代表第一個window內所有像素的亮度值所排成的向量， V_2 為代表第二個window內所有像素的亮度值所排成的向量，若場景僅僅是光照量發生變化，根據上面的關係可以寫出下列式子：

$$V_1 = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_8 \\ I_9 \end{bmatrix}, \quad V_2 = \begin{bmatrix} I_1' \\ I_2' \\ \vdots \\ I_8' \\ I_9' \end{bmatrix} = \begin{bmatrix} \alpha \cdot I_1 \\ \alpha \cdot I_2 \\ \vdots \\ \alpha \cdot I_8 \\ \alpha \cdot I_9 \end{bmatrix} = \alpha \cdot \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_8 \\ I_9 \end{bmatrix}$$

因此若兩張同一場景之相對應視窗內所有像點，僅有光照度發生改變的話，則所代表的兩個 vector 以向量的角度來看，將呈現一平行的關係。

所以，要決定兩張影像中的同一點是否有變化，可藉由檢查對應的兩個向量是否為平行。若兩點的值並不相等但分別對應的向量卻具有平行關係，則表示該點週圍其他點的值也呈等倍數的增減，即意謂著可能是整個環境的光照度發生變化，該點可以視為沒有變化而歸類為背景。相反的，若兩向量非平行，即代表該 $n \times n$ 範圍內的值的變化並不一致，也就是說，有可能是前景所在的位置。

檢查平行的方法是利用向量內積的性質。先將兩向量各別取其同方向的單位向量，之後取其內積，內積所得值再利用 \arccos 函數即可求得兩向量間的夾角。其圖示如下。

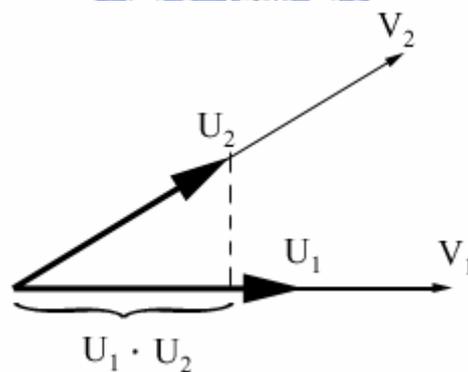


圖 3.6 向量夾角。

除此之外，若僅僅以影像上像素的亮度值的差異來計算，因

$$Intensity = \frac{(R + G + B)}{3}$$

有可能發生顏色與背景不同，但亮度(灰階)值相同的外物進入場景中，而無法辨識的狀況。為了避免這樣的情況，處理時是對 R、G、B 三個不同的色頻同時檢查，任一色頻發生了變化即認為該點發生了

變化。

而在實際應用上，VECTOR MODEL 的前景偵測結果會取決於所開啓視窗大小，也就是 n 的大小，當視窗開的較大時，優點是前景的偵測會較完整，不易有破碎情形發生，缺點則是計算時間較久，且物體輪廓會明顯變胖；反之，當視窗開的較小時易受雜訊的影響，偵測到的物體容易破碎而不完整，優點是計算時間短且輪廓型形狀與原始物體較為接近。由於在此希望前景物體偵測的較完整，所以將視窗大小 n 值設的比較大，在第五章實驗結果有列出不同的視窗大小與臨界值的結果。



圖 3.7 利用 VECTOR MODEL
在影像變亮後的偵測結果

詳細的傳統背景相減與 VECTOR MODEL 方法之比較實驗結果見第五章。

3.1.2 去除雜訊造成的零碎區塊並切割出前景物體

在此採用的方法是將由 3.1.1 所得到結果影像上相鄰的像素組合成的區塊找出，在相鄰的判斷上採用的是 8-connected 的判斷法，也就是說，只要是在該像素上緊臨的八個位置上即視為相鄰。不同的區塊配以不同的標記 (Labeling)，進行區塊標記 (Component Labeling)，如下圖所示：

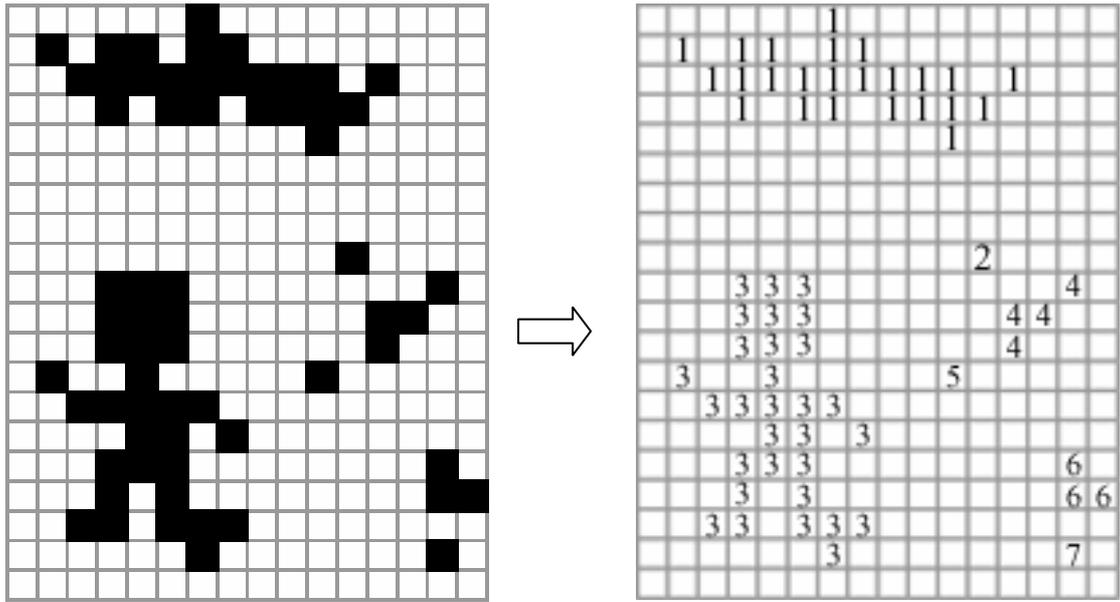


圖 3.8 Component Labeling 示意圖

再統計每個區塊中像素的個數，定立一個門檻值，將像素個數太少的區塊加以移除，而剩下的區塊就是想要偵測的前景物體。接著切割出偵測到的前景物體：



圖 3.9 利用 Component Labeling 去除雜訊並將內部破洞作修補後之影像



圖 3.10 切割出的前景物體

由於爲了希望輪廓的完整，將 VECTOR MODEL 的視窗加大，造成了輪廓明顯變胖，因此必須藉由形態學運算 Erosion 加以修正，而由於物體向外擴張的輪廓大約 4 個像素，所以選擇使用 9X9 mask 進

行一次的 Erosion，即可得到欲偵測的前景物體。修正之後的結果如下：



圖 3.11 經過 Erosion 運算後切割出的前景物體

3.1.3 車輛偵測演算法

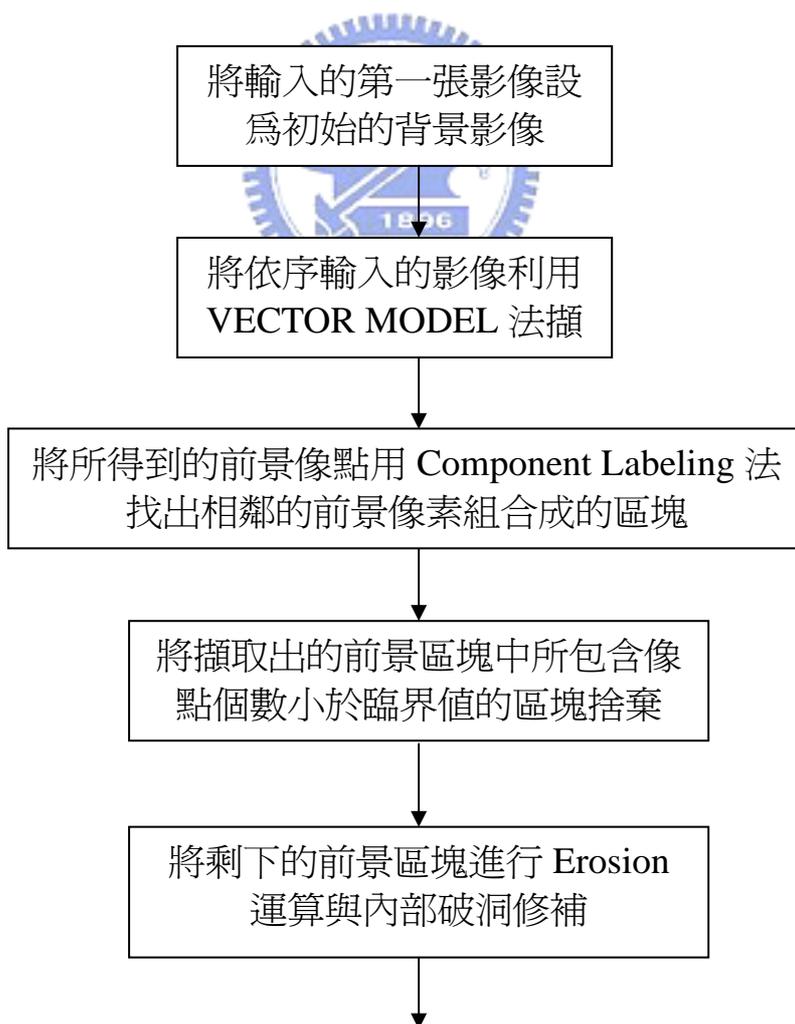
車輛偵測演算法詳細過程如下：

演算範疇：

輸入：不同時間點所拍攝的停車場之連續影像序列。

輸出：偵測並切割出停車場中移動的物體，包括車輛與人。

流程圖：



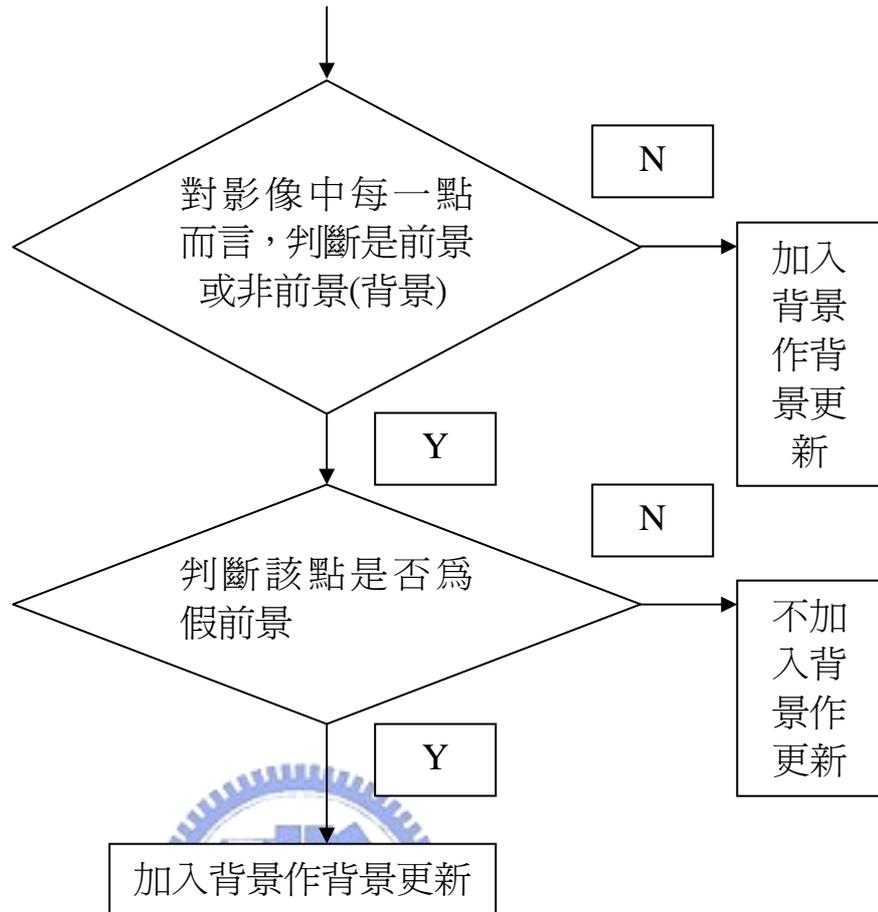


圖 3.12 偵測演算法流程圖

詳細演算法：

- (1) 將輸入的第一張影像設為初始的背景影像。
- (2) 計算影像序列中各影像像點周遭 $n \times n$ 大小範圍內的像素的亮度值所排列而成的 vector 與背景影像中相同位置像點周遭 $n \times n$ 大小範圍內的像素的亮度值所排列而成的向量間之夾角。
- (3) 當兩個向量之間的夾角大於事前所給定的門檻值，就說該像點發生了變化，即將該點視為前景。
- (4) 接著利用區塊標記(Connected Component Labeling)法，將影像上相鄰的前景像素組合成的區塊找出。
- (5) 統計每個區塊中像素的個數，定立一個門檻值，將像素個數太少的區塊加以移除。

- (6) 將剩下的前景區塊利用一次 Erosion 運算，讓前景物體的輪廓能夠更精確，修正因為 VECTOR MODEL 開啓較大視窗造成輪廓不精確的情形。
- (7) 最後，根據所得到的前景區塊，在目前的影像中切割出前景物體，並將非前景像素點加入背景做更新。

3.1.4 背景更新[4]

到目前為止所考慮的均為在固定背景的情況，但實際上背景的光源會隨著時間的改變而變動，或受到局部雜訊的影響而改變，因此必須對背景作適當的調適與補償。

除了開始的初始背景，其餘的每一個背景皆可能會因為光源改變，必須隨著時間的改變而做適當的更新，對於連續影像中的目前影像 I_t ，目前背景影像 B_{t-1} ，也就是經由時間 $t-1$ 更新過的背景來當時間 t 的背景影像採用以下的更新方式：

$$B = W_1 * I_t + W_2 * B$$

其中 w_1 與 w_2 表示目前影像與目前背景影像所佔的權值

如 $w_1 = 1/2$ ， $w_2 = 1/2$ 表示的 I_t 、 B_{t-1} 影響力均等

$w_1 = 1/3$ ， $w_2 = 2/3$ 表示 I_t 影響力較 B_{t-1} 小

$w_1 = 2/3$ ， $w_2 = 1/3$ 表示 I_t 影響力較 B_{t-1} 大

但是，由以上的式子會發現一個問題，將 I_t 內所有的點不論為前景或背景，都加入更新背景影像，這樣會造成背景不正確。事實上應該避免移動物體(前景)也被加入更新，而僅加入真正的背景影像去更新背景，因此將上式修正如下：

對於 I_t 中的每一點 (i, j) 而言，採用以下方式更新背景：

$$\begin{cases} B(i, j) = w_1 * I_t(i, j) + w_2 * B(i, j) & , \text{假如}(i, j) \text{在 } I_t \text{中爲背景} \\ B(i, j) = I_t(i, j) & , \text{假如}(i, j) \text{在 } I_t \text{中爲假前景} \end{cases}$$

3.1.5 原始背景被遮蔽之更新

事實上，由於停車場車輛來來往往所以很難達到完全淨空的狀態，因此無法取得停車場完全淨空的純背景圖，只能以第一張獲得的影像當成純背景圖，這樣做會造成假背景的出現，舉例來說，原先停車場部分停車格就停有車輛，卻以此圖當成純背景圖，之後的每張影像都與此張影像相減，一旦這些一開始就停在停車格的車輛開始移動後，與純背景圖相減的結果將造成假前景的出現，誤以為真的有新前景物體出現，因此必須要能判斷出此前景物體並非真正的前景物體時是屬於背景像的一部分，判斷出來之後還必須將此實為背景區域加入背景影像中更新。判斷原則為：對於第一次出現在場景中的前景物體開始進行數張影像的監控，當發現該前景物體在出現後就不再有任何移動，就說該前景物體是假的前景物體，實際上是背景區域。



圖 3.13 初始的背景影像



圖 3.14 $t = 5$ 原本屬於背景的車輛開始移動，使得真正的背景漸漸顯露出來



圖 3.15 經由 VECTOR MODEL 偵測的結果



圖 3.16 經由前景偵測演算法偵測結果



圖 3.17 $t = 10$ 真正的背景全部顯露出來



圖 3.18 經由 VECTOR MODEL 偵測的結果



圖 3.19 經由前景偵測演算法在原先車輛停放的位置偵測出一個新的前景物體，開始監控此前景物體狀態變化。



圖 3.20 經過一段時間該前景物體仍然沒有改變狀態，正式將我們切割出來的假前景加入背景做更新，此為更新後的背景影像，原先被遮蔽的背景部份以完全加入背景做更新，我們獲得更正確的背景影像資訊。

以上詳細的過程列出在第五章實驗結果。

3.2 車輛追蹤[2,3]

當正確偵測出影像序列中之每一個前景物體後，仍然無法得知在不同影像序列間前景物體之相互關係，此時就必須利用追蹤技術，比對不同影像序列間前景物體的特徵來正確追蹤不同影像序列中之相同的前景物體，以達到車輛監控與停車場事件處理的目的。

然而，在車輛追蹤過程中常常會遇到許多問題，包含車輛交錯與

車輛遮蔽問題，車輛交錯主要是由於車輛在行進的過程中發生了錯車(或會車)，造成在前景偵測時雖然理論上有兩輛車存在，但實際上卻僅僅偵測到單一車輛，如果不加以處理，發生一輛車憑空消失，而另一輛車卻又追蹤不到正確位置的情況。另一個則是由車輛交錯衍生出的車輛遮蔽問題，車輛在交錯的過程中，往往就會發生車輛遮蔽，此時我們必須判定出哪一輛車遮蔽哪一輛車或哪一輛車被遮蔽，才能夠正確的掌握住車輛在發生交錯時的輪廓外型變化。

3.2.1 基本車輛追蹤[12]

問題描述：

假設經由車輛偵測演算法在時間 t 時偵測到 $OB1$ 與 $OB2$ 兩個獨立的物體，而在時間 $t+1$ 時偵測到 $OB3$ 與 $OB4$ 兩個獨立的物體，希望能夠得知在時間 t 的 $OB1$ 與 $OB2$ 在時間 $t+1$ 時變成 $OB3$ 或是 $OB4$ 。

演算法則：

1. 由於前後影像序列間隔時間很短，所以可以根據 $OB1$ 由時間 $t-1$ 到時間 t 所前進的距離與方向，來預測 $OB1$ 在時間 $t+1$ 時所應在的位置。



圖 3.21 追蹤預測圖

2. 計算時間 $t-1$ 與 t 車輛輪廓形狀的特徵值與特徵向量，長軸即為較大的特徵值所對應的特徵向量，我們以長軸當成車輛前進的方向，接著計算時間 $t-1$ 與 t 兩根長軸之夾角，以此夾角當成預測在時間 $t+1$ 時車輛轉彎的角度。



圖 3.22 車輛輪廓之長軸與短軸

3. 以 1,2 計算的結果，預測在時間 $t+1$ 重心所應該在的位置，及車輛輪廓所應該轉動的角度。
4. 根據所預測的重心位置，開啓一個適當大小的追蹤視窗

(假如在所開啓的追蹤視窗內沒有任何前景，表示必須加大我們的追蹤視窗)，紀錄所有落在此追蹤視窗內的前景物體，這些前景物體即有可能為在時間 $t+1$ 時的 $OB1$ 。

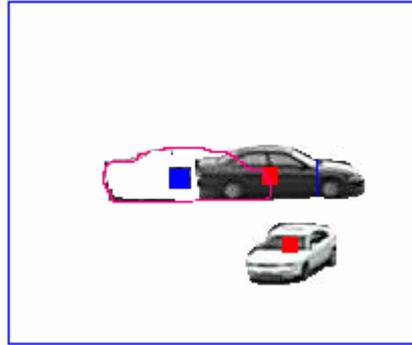


圖 3.23 藍色即為追蹤視窗，發現在時間 $t+1$ 時追蹤視窗內有兩個前景物體

5. 將這些落在 $OB1$ 之追蹤視窗內的前景物體分別與 $OB1$ 比對顏色分佈資訊(RGB Histogram)，找出與 $OB1$ 顏色分佈資訊(RGB Histogram)最為接近的前景物體，此物體即為時間 $t+1$ 的 $OB1$ 。
6. 而 $OB2$ 的作法亦同。

但是，有可能會有車輛交錯與車輛遮蔽的情況發生，必須靠以下的演算法加以解決。

3.2.2 車輛遮蔽的判定

問題描述：

假設在時間 t 有兩個獨立物體 $OB1$ 與 $OB2$ ，在時間 $t+1$ 到 $t+n$ 期間發生了交錯，造成 $OB1$ 與 $OB2$ 在時間 $t+1$ 到 $t+n$ 期間合成單一物體 $OB3$ ，我們希望能夠得知在此期間是 $OB1$ 遮蔽了 $OB2$ 、 $OB2$ 遮蔽了 $OB1$ 或兩者並沒有相互遮蔽的現象發生。

演算法則：

1. 記錄 $OB1$ 與 $OB2$ 在時間 t 尚未發生交錯時的輪廓形狀。
2. 當車輛開始在時間 $t+1$ 發生交錯時，利用追蹤的預測機制及車輛尚未發生交錯時的輪廓形狀，預測在時間 $t+1$

時 OB1 與 OB2 車輛輪廓形狀所應該在的位置。



圖 3.24 藍色與綠色的輪廓分別代表在時間 $t+1$ 綠車與白車車輛輪廓形狀及所應該在的位置，可以觀察到交集區域面積不足，尚無法判斷。

3. 計算 OB1 與 OB2 之預測輪廓交集區域的面積，假如交集區域的面積大於假設的門檻值，就開始比對該交集區域中的每一像點與尚未發生交錯之 OB1 與 OB2 輪廓中相對應區域之對應像點顏色資訊(RGB 差值)何者較為接近，並做累計，最後會得到該交集區域與 OB1 及 OB2 相對應區域中相似像點的個數，如下圖所示；假如交集區域的面積小於我們假設的門檻值，則不予比對，而在下一張影像中繼續觀察交集區域的面積。]

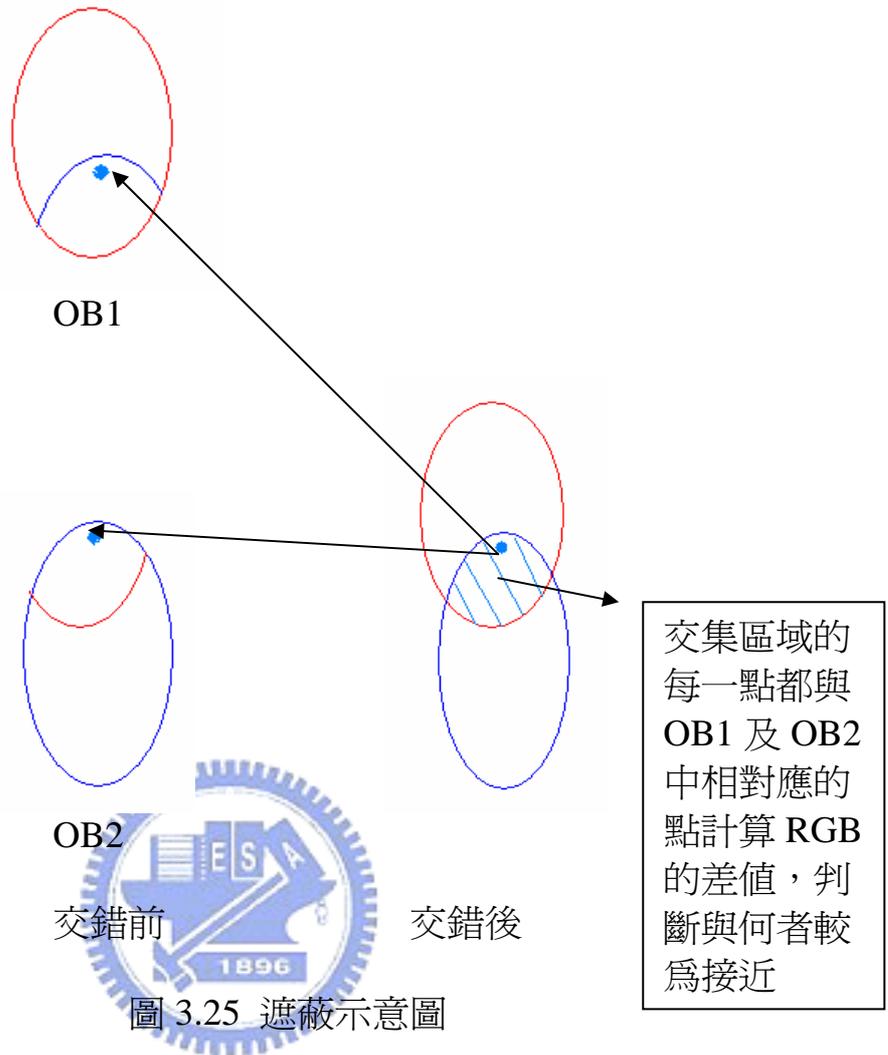


圖 3.26 藍色與綠色的輪廓分別代表預測在下一個時間綠車與白車車輛輪廓形狀及所應該在的位置，交集區域的面積已經大於我們所假設的臨界值，開始進行點對點顏色資訊分佈的比對，經由比對之後我們發現交疊區域與綠色車輛相似的像點個數為 213 點而與綠色車輛相似的像點個數為 13 點。

4. 經過比對之後，假如該交集區域之顏色分佈資訊(RGB Histogram)與 OB1 較為相近，我們就認定在此交錯期間 OB1 遮蔽了 OB2，否則就說 OB2 遮蔽了 OB1，而在接下來的交錯期間不再做判斷。

3.2.3 車輛交錯的追蹤[5]

問題描述：

假設在時間 t 有兩個獨立物體 OB1 與 OB2，在時間 $t+1$ 時發生了交錯，造成 OB1 與 OB2 在時間 $t+1$ 時合成單一物體 OB3，我們希望能夠把 OB3 中所包含的 OB1 與 OB2 給分離開來，使得我們能夠正確得知 OB1 與 OB2 在時間 $t+1$ 時的位置及輪廓形狀。

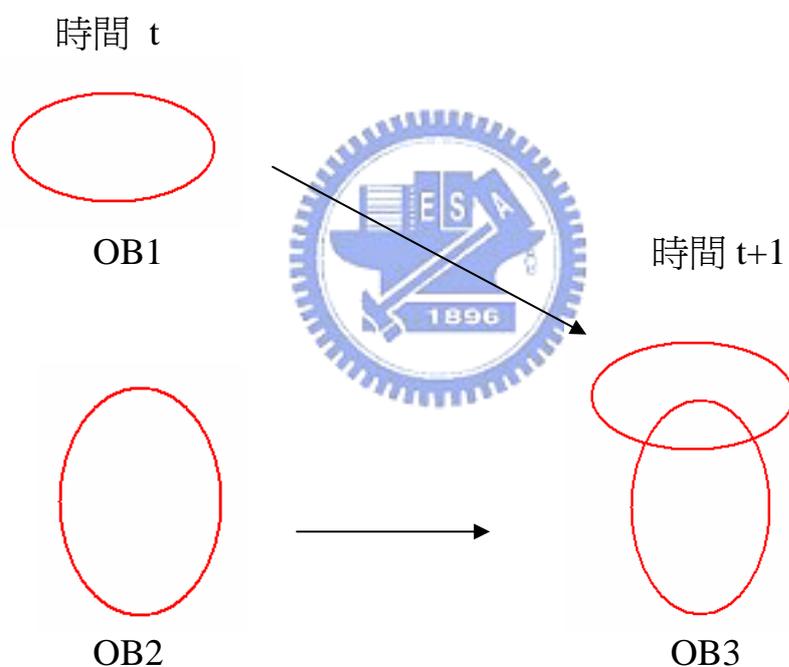


圖 3.27 前景物體發生交錯

演算法則：

1. 首先，根據 OB1 由時間 $t-1$ 到 t 所前進的距離、方向，來預測 OB1 在時間 $t+1$ 時重心所應該在的座標位置。
2. 同樣，根據 OB2 由時間 $t-1$ 到 t 所前進的距離、方向，來預測 OB2 在時間 $t+1$ 時重心所應該在的座標位置。

3. 接著，觀察 OB1 與 OB2 在時間 $t+1$ 的預測重心位置，如果這兩個預測重心位置都落在時間 $t+1$ 的某一個前景 (OB3) 物體輪廓形狀內，我們就說 OB1 與 OB2 在時間 $t+1$ 時發生了交錯，合成了單一物體 OB3。
4. 呼叫車輛遮蔽演算法，決定是 OB1 遮蔽 OB2、OB2 遮蔽 OB1 或兩者並沒有遮蔽的發生。
5. 在此分為兩種情況做討論：
 - 假設 OB1 遮蔽 OB2：
 1. 以 OB2 在時間 t 的輪廓形狀為一個 mask，並根據追蹤預測機制，得到此 mask 在時間 $t+1$ 所應該在的位置。
 2. 將此 mask 在預測位置附近不斷移動，找出與 OB2 顏色分佈資訊 (RGB Histogram) 較為接近的數個區塊。



圖 3.28 比對圖

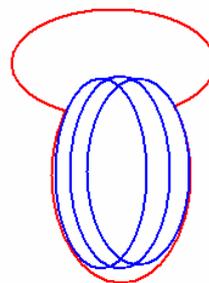


圖 3.29 藍色區塊即顏色分佈資訊與 OB2 較為接近的的區塊

3. 從 OB3 中去除掉與 OB2 顏色資訊分佈較為接近的區塊(也就是希望將 OB3 中可能是 OB2 的點都與以去除，僅留下 OB1 的部分)，而將剩餘的點利用 Component Labeling 法找出最大的區塊，定義為 OB1'。

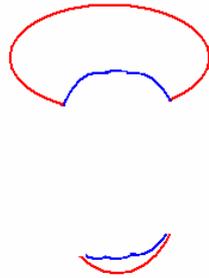


圖 3.30 上方的大區塊即為 OB1'

4. 再用 OB3 減去 OB1'得到一個新的區塊，定義為 OB2'。

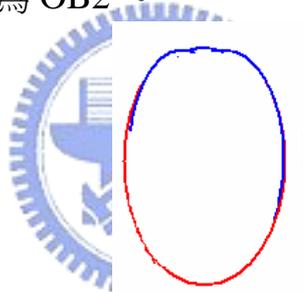


圖 3.31 OB2'

5. 再以 OB1 的輪廓視為為一個 mask，套用至 OB1'中，找出與 OB1 顏色分佈資訊(RGB Histogram)最為相近的區塊，定義為 OB1''。

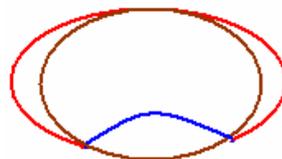


圖 3.32 咖啡色部分為 OB1''

- 再以 OB2 的輪廓視為為一個 mask，套用至 OB2' 中，找出與 OB2 顏色分佈資訊(RGB Histogram)最為相近的區塊，定義為 OB2''。(5,6 目的在於：在把不同物體的分開之後，重新套一次，更能找出最佳配對，而不受另一個物體影響)

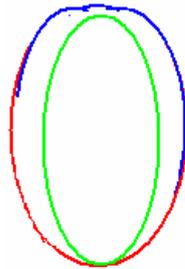


圖 3.33 綠色部分為 OB2''

- 找出 OB1'' 與 OB2'' 交集的區域，定義為 OB4，比對 OB4 之顏色分佈(Color Histogram)資訊與 OB1'' 及 OB2'' 何者較為接近，假設 OB4 與 OB2'' 較為接近，因此就把 OB1'' 減去 OB4，得到一個新的區塊 OB1'''(反之，假如 OB4 與 OB1'' 較為接近，就把 OB2'' 減去 OB4，得到一個新的區塊 OB2''')。

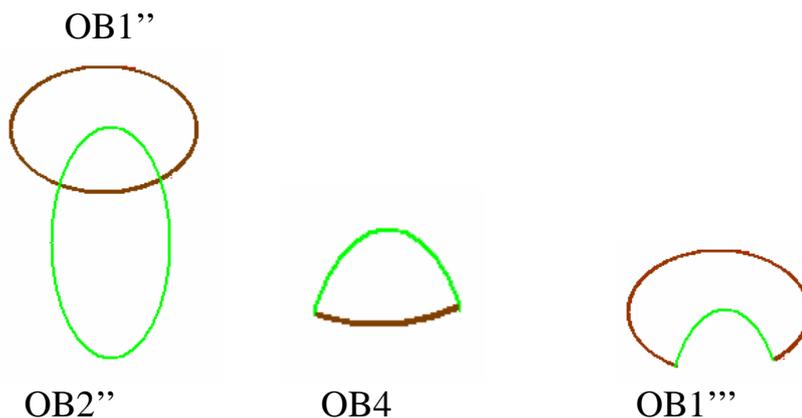


圖 3.34 切割後的結果

8. 再把 OB3 減去 OB1''' 與 OB2''，得到一些零碎的區塊，對於這些零碎區域的每一個像點，都去計算與 OB1''' 及 OB2'' 輪廓形狀的最短距離，假如該像點與 OB1''' 輪廓的最短距離小於與 OB2'' 輪廓的最短距離，就把該像點加入 OB1''' 中，否則就加入 OB2'' 中。



圖 3.35 剩餘的零碎區塊

9. 最後，OB1 在時間 $t+1$ 的位置及輪廓形狀就是 OB1'''，而 OB2 在時間 $t+1$ 的位置及輪廓形狀即為 OB2''。
- 無法判定誰遮蔽誰：
表示交集區域的面積不大，並沒有明顯的遮蔽效應出現，因此可以假設 OB1 遮蔽 OB2 或 OB2 遮蔽 OB1 並重複上述步驟即可。

四、停車場車輛事件偵測與處理

4.1 停車格簡介

對於大多數停車場而言，監視攝影機往往固定後即不再有所變動，因此可以事先以手動的方式告知攝影機停車場之停車格座標位置，並以此停車格座標位置，來當作之後事件判斷的基礎。人工定義停車格平面圖如下所示：



圖 4.1 停車格平面圖

手動標示的停車格如下所示：

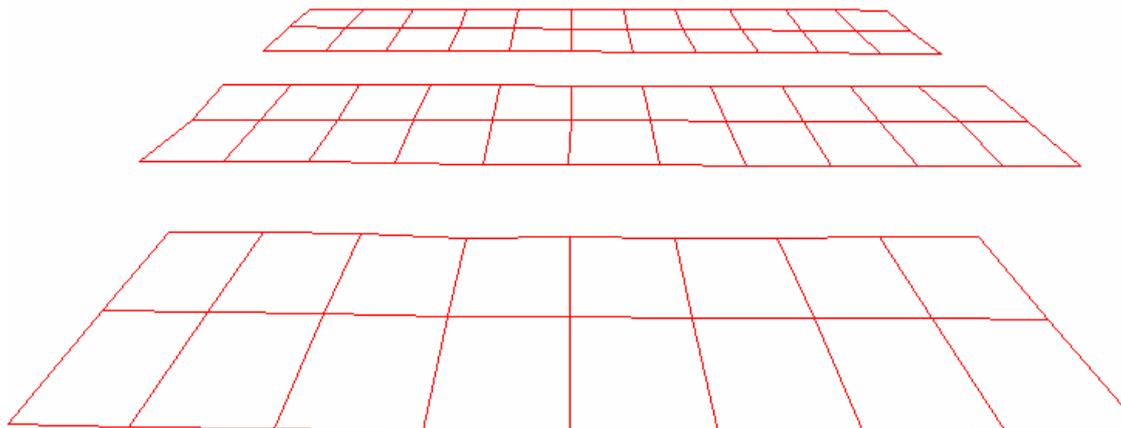


圖 4.2 手動標示停車格平面圖

將停車格套用到停車場平面圖的結果如下所示：



圖 4.3 將手動標示停車格套用至實際停車場

4.2 停車場車輛事件偵測與處理

當車輛進入停車場後，往往會有許多事件發生，包括車輛違規停放、沒有正確停入停車格或是其他一些可疑的情況等等，對於這些事件都要能夠偵測，並做出適當的處理，以確保停車場的正常運作。

4.2.1 違規停放車道上

對於進入停車場的車輛，用以下的步驟判定是否違規停放在車道上：

1. 當車輛進入停車場後，開始啓用的追蹤機制，監控進入停車場的車輛。
2. 當發現車輛停止不動後，開始計算停止的時間。
3. 當停止的時間超過我們預定的門檻值時，就假設車輛目前的狀態是靜止的。
4. 立刻將車輛所在的座標位置，與所有事先定義好的停車格座標位置做比對，判斷是否與停車場內某一個停車格座標位置有所交疊。
5. 若該車輛靜止的座標位置，並沒有與任何停車格座標位置有所交疊，就表示該車輛違規停放在車道上，立即將該違規影

像傳給資料庫管理系統，並通知停車場管理員做處理。

4.2.2 是否正確停入停車格內

對於進入停車場的車輛，用以下的步驟判定是否正確停入停車格內：

1. 當車輛進入停車場後，開始啓用追蹤機制，監控進入停車場的車輛。
2. 當發現車輛停止不動後，開始計算停止的時間。
3. 當停止的時間超過預定的門檻值時，就表示車輛目前的狀態是靜止的。
4. 立刻將車輛所在的座標位置，與所有事先定義好的停車格座標位置做比對，判斷是否與停車場內某一個停車格座標位置有所交疊。
5. 若該車輛靜止的座標位置，確實與事先定義好的停車格座標位置有所交疊，則找出交疊面積最大的停車格。
6. 利用以下兩項條件，判定該車輛是否正確停入該停車格內：
 1. 計算該停車格被該車輛遮蔽的面積佔全部停車格的面積的百分比，必須超過所給定的臨界值。
 2. 利用特徵向量計算車輛輪廓之長軸與短軸，長軸與短軸代表車輛停放的方向，停放的方向必須與停車格的方向相吻合，也就是長軸與停車格的較長一邊之夾角必須小於我們事先所給定的臨界值。

參數：車輛覆蓋停車格面積之比例臨界值為 75%

車輛長軸與停車格較長邊之夾角臨界值為 15 度

如圖例：



圖 4.4 車輛違規停放一

車輛覆蓋停車格面積之比例：43.5%

車輛長軸與停車夾角：30 度

首先，停車方向與停車格方向夾角車超過所定的臨界值，表示車輛停放位置有所偏移與停車格方向不一致，且車輛覆蓋停車格面積之比例不到 75%，因此判定該車輛並沒有正確停放停車格內。



圖 4.5 車輛違規停放二

車輛覆蓋停車格面積之比例：80.3%

車輛長軸與停車夾角：45 度

首先，雖然車輛覆蓋停車格面積之比例高達 80%，超過了臨界值，但停車方向與停車格方向夾角卻車過所定的臨界值，表示車輛停放位置有所偏移與停車格方向不一致，因此判定該車輛並沒有正確停放停車格內。

7. 假如以上兩項條件都滿足，我們就說該車輛的確正確停入停車格內，並將該張影像回報給資料庫管理系統，以便之後提醒車主車輛停放位置；否則就說該車輛未正確停入停車格內，同樣將該張影像回報給資料庫管理系統，並通知管理員做處理。

五、實驗數據與結果

5.1 系統執行環境與影像之取得

本系統所擷取的影像序列，是在停車場旁高樓架設數位攝影機，以便取得停車場的高空俯視影像。數位攝影機為 SONY DCR-PC10，之後所得影像序列透過網路(web browser)將之傳送至執行實驗的機器做處理，影像序列資料格式為 24bit 的 RGB 影像，480x720 像素的大小的畫面解析度，每秒 29.976 畫面的 MPEG2 視訊檔，用來模擬的硬體設備為 CPU Pentium IV 2.4G、RAM 1.0G 的電腦，以 Mathworks 公司出的 MatLab 6.5 為模擬的開發整合環境。實驗影像是採用每秒 15 張的 24bit 的彩色連續影像序列(JPEG 圖像檔)。



5.2 車輛偵測結果

5.2.1 利用 VECTOR MODEL 在不同參數下場景變化偵測結果

實驗目的：得知 VECTOR MODEL 在不同參數下場景變化偵測結果



圖 5.1 初始背景影像



圖 5.2 目前輸入影像

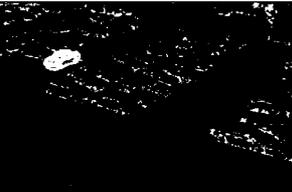
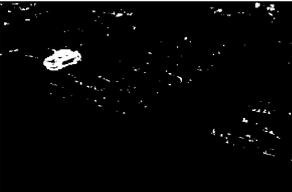
	Angle = 6	Angle = 9	Angle = 12
3×3			
5×5			
7×7			
9×9			

圖 5.3 VECTOR MODEL 在不同參數下的前景偵測結果

結論：經由以上的實驗數據可以得知，當視窗大小 7×7、角度臨界值 = 9 度時，最能保持前景物體的完整與去除雜訊所造成的影響

5.2.2 VECTOR MODEL 在場景光線明顯變化下之偵測結果

實驗目的：得知 VECTOR MODEL 在場景光線明顯變化影像的前景偵測能力

實驗參數：視窗大小 7×7 、門檻值 9 度



T = 1 初始背景圖

	目前輸入影像	利用 VECTOR MODEL，視窗大小 7×7 、門檻值 9 度之前景偵測的結果	更新後之背景影像(目前影像取 $1/5$ ，目前背景取 $4/5$ 後加入背景做更新)
T=2			
T=3			
T=4			

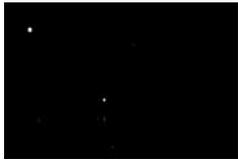
T=5			
T=6			
T=7			
T=8			
T=9			

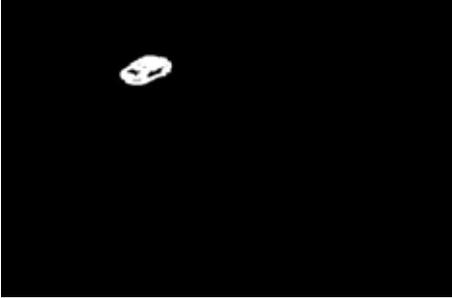
圖 5.4 VECTOR MODEL 在場景光線明顯變化影像的前景偵測結果

結論：由以上的結果可以看出 VECTOR MODEL 的確能夠不受光線變化的影響

]

5.2.3 切割出經由 VECTOR MODEL 偵測出的前景物體

實驗目的：利用前景偵測演算法切割出前景物體

目前背景影像	
目前輸入影像	
經由 VECTOR MODEL 在角度臨界值 9 度，視窗大小 7×7 的偵測結果	
利用 Component Labeling 去除雜訊後之影像	

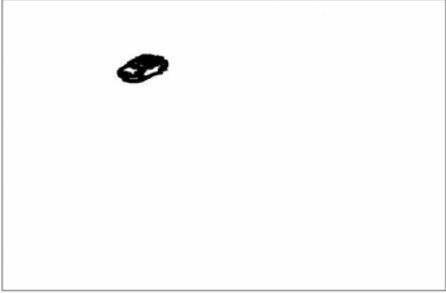
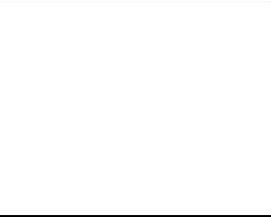
<p>將上圖反相之後的 結果</p>	
<p>將反相之後落在車輛輪廓內的白色區塊加入 Component Labeling 的結果中做修補</p>	
<p>切割出的前景物體</p>	
<p>再經過 Erosion 運算後切割出的前景物體</p>	

圖 5.5 車輛偵測演算法流程圖

5.2.4 原始背景被遮蔽之更新

實驗目的：還原原先被遮蔽的背景區域

	目前輸入影像	偵測出的前景	目前背景
T=2			
T=3			
T=4			
T=5			
T=6			
T=7			

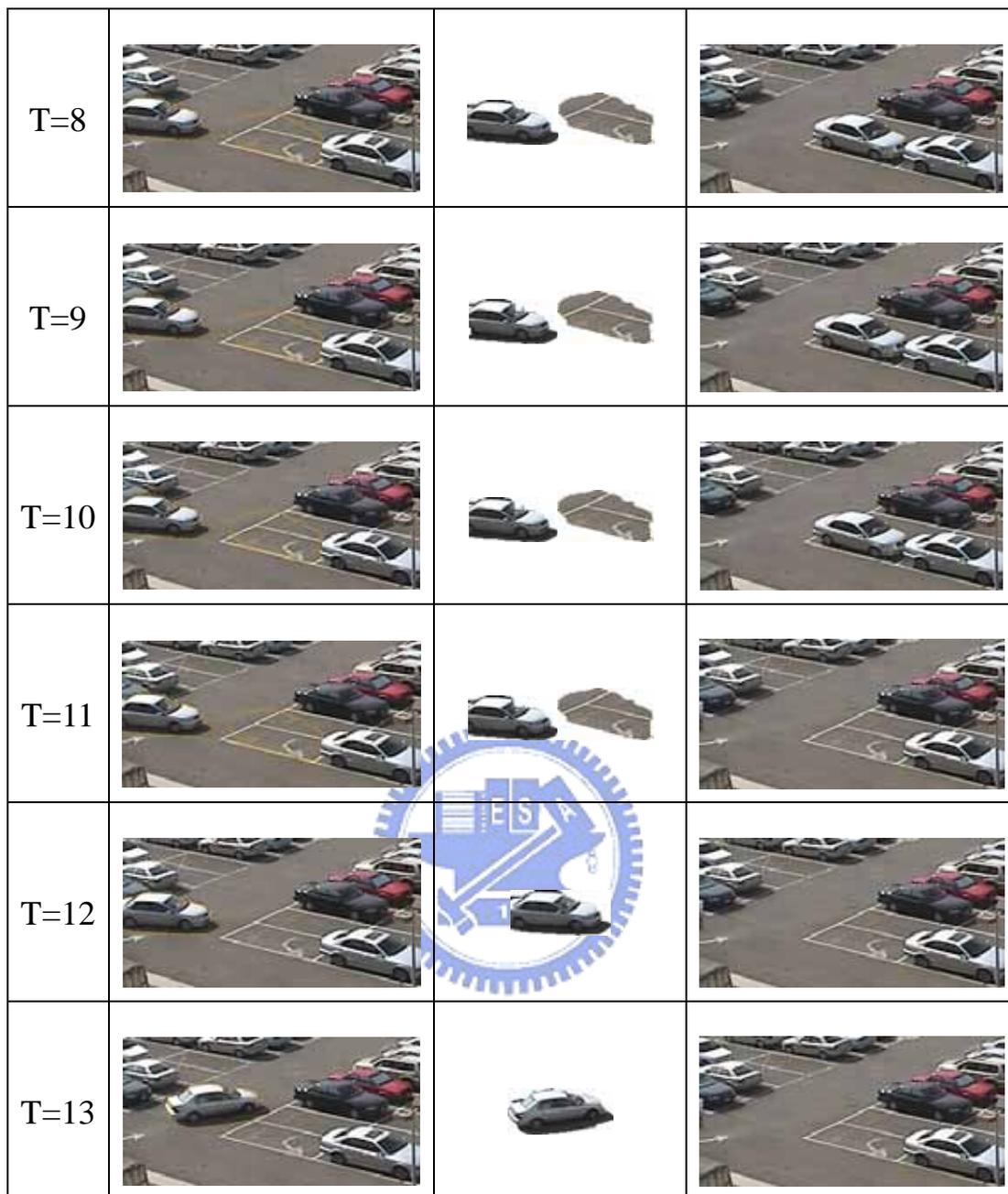


圖 5.6 原先被遮蔽的背景區域還原過程

5.3 車輛追蹤與車輛交錯之結果

5.3.1 基本車輛追蹤

	追蹤結果
T=1	
T=2	
T=3	
T=4	

T=5	
T=6	
T=7	
T=8	
T=9	

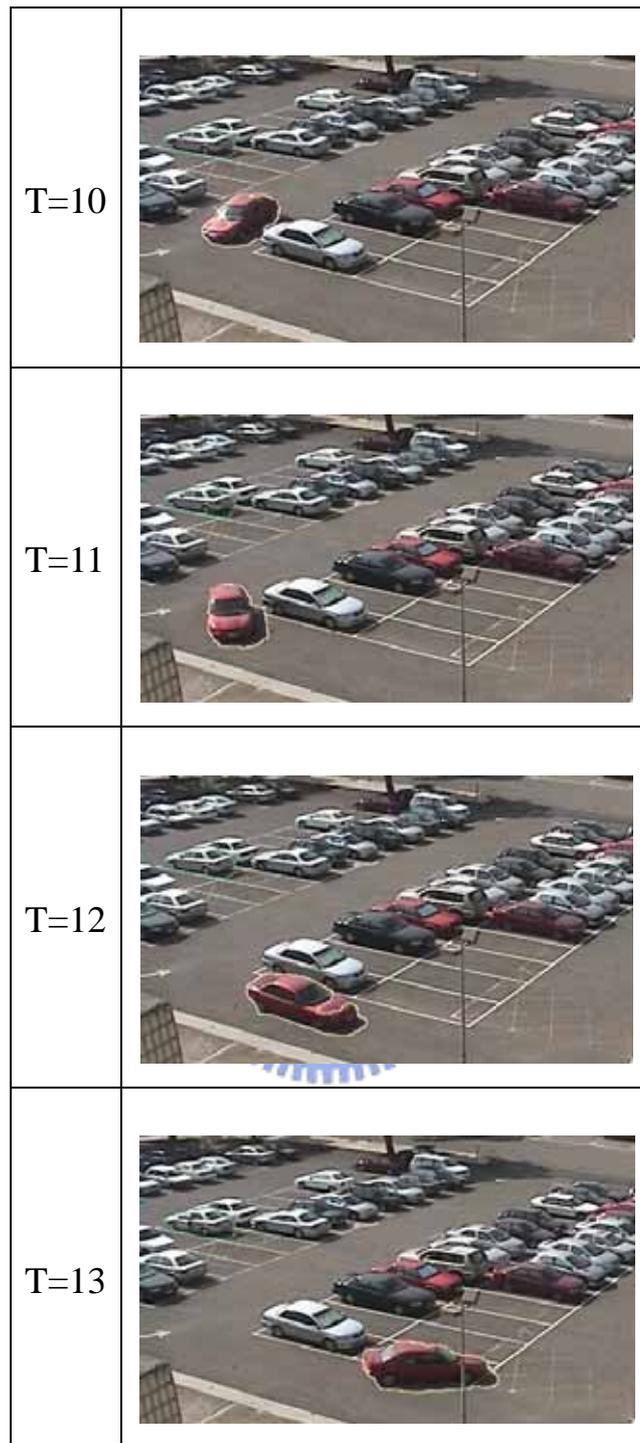


圖 5.7 基本車輛追蹤圖

5.3.2 車輛交錯之追蹤

時間	追蹤結果	時間	追蹤結果
T=1		T=8	
T=2		T=9	
T=3		T=10	
T=4		T=11	
T=5		T=12	
T=6		T=13	
T=7		T=14	

T=15	
T=16	
T=17	
T=18	
T=19	
T=20	
T=21	

圖 5.8 車輛交錯影像序列

5.4 車輛事件偵測與資料庫間的互動

5.4.1 判斷是否正確停入停車格

實驗目的：偵測車輛是否未正確停放停車格

實驗參數：車輛覆蓋停車格面積之比例臨界值為 75%

車輛長軸與停車格較長邊之夾角臨界值為 15 度



圖 5.9 車輛違規停放判定示意圖一

車輛覆蓋停車格面積之比例：43.5%

車輛長軸與停車格較長邊之夾角：30 度

首先，停車方向與停車格方向夾角車超過所定的臨界值，表示車輛停放位置有所偏移與停車格方向不一致，且車輛覆蓋停車格面積之比例不到事先給定的臨界值，因此判定該車輛並沒有正確停放停車格內。



圖 5.10 車輛違規停放判定示意圖二

車輛覆蓋停車格面積之比例：80.3%

車輛長軸與停車格較長邊之夾角：45 度

首先，雖然車輛覆蓋停車格面積之比例高達 80%，超過了臨界值，但停車方向與停車格方向夾角卻車過所定的臨界值，表示車輛停放位置有所偏移與停車格方向不一致，因此判定該車輛並沒有正確停放停車格內。

5.4.2 停車場完整事件流程圖：



圖 5.11 車輛進入停車場，經由門口攝影機拍攝車輛含車牌影像，並將此影像透過網路傳遞給中央管理系統。中央管理系統開始針對此張影像進行車牌辨識，並將辨識所得到的車牌號碼傳給資料庫管理系統

停車場資料

停車編號： 1

車牌號碼： 1

進入時間： 2004-07-09 00:08:35

您的停車格為下圖有顏色的部份

01	02	03	04	05	06	07	08	09	10	11
12	13	14	15	16	17	18	19	20	21	22
			23	24	25	26	27			
			28	29	30	31	32			

圖 5.12 資料庫管理系統根據得到的車牌號碼建立新的帳戶，並發給使用者停車資訊單據，記載進入時間與建議停放位置



圖 5.13 攝影機偵測到有車輛進入停車場，開使進行追蹤



圖 5.14 持續的追蹤車輛中



圖 5.15 偵測到使用者將車輛停入停車場停車格中，因此呼叫資料庫管理系統開始處理使用者資料



圖 5.16 資料庫中記載目前停車場車位的資訊，停車格編號 17 已被使用者停放



圖 5.17 另一輛白色的車進入停車場



圖 5.18 偵測出白色的車輛沒有正確停入停車格



圖 5.19 資料庫中記載目前停車場車位的資訊，其中紀錄下車輛停放超過停車格



圖 5.20 偵測到綠色車輛要離開停車場資料庫開始統計時間、與金額：

停車場資料
 停車編號： 1
 車牌號碼： 1
 進入時間： 2004-07-09 00:53:45
 離開時間： 2004-07-09 00:53:19
 所需繳交金額： 30



圖 5.21 資料庫系統釋放 17 號停車格



六、結論與未來工作方向

本篇論文目標在提供一個無人管理，卻有效率，迅速，而安全的自動化停車場管理系統。主要是藉由視覺監控系統(Visual-based surveillance technique)達成，系統大致可分為三個部份：資訊收集、資訊傳輸及資訊處理。本篇論文所著重的，在於資訊處理部份，即將接收到的影像資訊，透過本篇論文發展出的技術，提供停車場管理者與使用者最佳的協助。

依停車場管理的流程可以分為車輛辨識，車輛偵測與追蹤及車輛監控三個階段

(4) 車輛辨識階段

- (a) 偵測車輛之進入：偵測是否有車輛即將要進入停車場，當車子完全進入攝影機(停車場)後，開始進行車牌辨識。
- (b) 車牌辨識：對於進入停車場的車輛進行車牌辨識，包括以下步驟：車牌定位、字元切割與車牌 OCR (Optical Character Recognition) 辨識。



(5) 車輛偵測與追蹤階段

- (a) 車輛偵測與追蹤：偵測進入停車場內的車輛，並持續追蹤該車輛，以便隨時掌握該車輛在停車場內的狀態。

(6) 車輛監控階段

- (a) 監控車輛有無正確停放在停車格內：
- (b) 監控車輛是否發生碰撞：監控在停車場內移動的車輛，是否有碰撞到其他早已停放在停車場內的車輛。
- (c) 監控異常情況：車主將車輛停入停車場格後，是否就待在車上而沒有下車；車主前來取車時，是否進入車內後，卻遲遲未將車開離。
- (d) 監控車輛是否要離開：高空攝影機偵測到有車輛離開停車格後，便通知資料庫管理系統作處理，資料庫管理系統會做適當的處理，並釋放該停車格。

未來工作除了改善現有一些停車場處理管理功能外，包括擴展二維影像處理技術為三維影像處理技術，以取得車輛與停車格三維空間的資訊提昇處理準確度，還要建立多攝影機監控系統，透過網路及分散式系統連接，做大範圍的人與車輛的自動偵測、追蹤、辨識、管理。



參考文獻

1. E. Durucan and T. Ebrahimi, "Change Detection and Background Extraction by Linear Algebra," Proceedings of the IEEE, vol.89, no.10, pp.1368-1381, Oct. 2001.
2. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfunder: Real-time tracking of the human body. Technical Report 353, MIT Media Lab Perceptual Computer Section, 1995.
3. B. Coifman, D. BEymer, et al., "A Real-time computer vision system for vehicle tracking and traffic surveillance," Transportation Research Part C: Emerging Technologies, Vol. 6, No. 4, August, 1998, pp. 271-288
4. C. Stauffer, W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," Proceedings of CVPR 1999, June 1999, pp. 246-252
5. G. S. K. Fung, N. H. C. Yung, G. K. H. Pang, "Vehicle shape approximation from motion for visual traffic surveillance," Proceedings of 2001 IEEE Intelligent Transportation Systems ,2001, pp.608-613
6. J. C. Rojas, J. D. Crisman, "Vehicle detection in color images," Proceedings of IEEE Int. Conf. on Intelligent Transportation System, 1997.
7. K. Skifstad and R. Jain, "Illumination independent change detection for real world image sequences," CVIP, vol.46, no.3, pp.387-399, 1989.
8. T. Aach et al., "Statistical model-based change detection in moving video," Signal Process., vol.31, pp.165-180, 1993.
9. M. Fathy and M. Y. Siyal, "A Window-based Edge Detection Technique for Measuring Road Traffic Parameters in Real-Time", Real-Time Imaging, Vol. 1, pp.297-305, 1995.
10. A. V. Oppenheim, R. W. Schafer, and T. G. Stockham Jr, "Nonlinear filtering of multiplied and convolved signals," Proc. IEEE, vol. 56, pp. 1264-1291, Aug 1968.
11. Milan Sonka, Vaclav Hlavac, Roger Boyle, "Image Processing, Analysis, and Machine Vision", second edition.
12. Greg Welch; Gary Bishop; "An Introduction to the Kalman Filter" University of North Carolina at Chapel Hill Department of Computer

Science Chapel Hill, NC 27599-3175

- 13.黃聿彰, ”三維空間多攝影機視覺監控系統”, 國立交通大學, 碩士論文, 民國 91 年

