

私立東海大學資訊工程研究所

碩士論文

指導教授：林正基 博士

An Object Segmentation System Design using  
Group-based Color Algorithm

分群式背景色彩演算法之移動物件切割系統研究

研究生：謝秉荃

中華民國 一百零二 年 七月

## 摘要

影像背景切割為擷取移動物體資訊中相當重要的技術，為了得到正確的背景資訊，必須創一個背景模型來切割前景與背景。傳統的背景切割演算法在處理複雜的背景資訊有不錯的效果，但是也帶來比較多的處理時間和較大的記憶體空間。本篇論文提出了一個全新的演算法來偵測前景中的移動物體，利用區塊式處理影像中的像素，再將其像素們一一分群後來取得背景色彩資訊。區塊式與像素分群的處理使我們可以大量減少建立背景時所需的運算，並加速取得背景資訊。之後透過這些分群式背景資訊就可以擷取出移動物體資訊，再提供給使用者相關的警示，達到維護社會安全的目的。

**關鍵詞：**主體色彩、移動物體偵測、背景資訊、分群式計算。

## **Abstract**

Background segmentation is a very important technology to get information about moving objects. In order to get the right background information, you must create a background model to segment the foreground and background. The traditional segmentation algorithm have a good effect on dealing with complex background information, but it also brings more processing time and larger memory space. This paper presents a new algorithm to detect moving objects in the foreground, the use of block-based processing of pixels in the image, and then one by one grouping whose pixels are subsequently acquires the background color information. The block-based and pixel grouping process can significantly reduce the time requirement to create a background model and accelerate get background information. Through grouping color after background information you can extract information about moving objects, and then provided to the user-related alerts, to maintain social security purposes.

**Keywords ;** background segmentation, background model, block-based process, group color

## 致謝

兩年的碩士生活即將結束，感謝指導教授一路上指導與教學，以及許多包容才能夠讓我完成此篇論文。另外也要感謝口試委員的指導和建議，能夠讓此篇論文更加的完整。感謝同屆的碩班同學，一起寫論文的日子是我們永生難忘的回憶。感謝實驗室的學長姐以及學弟妹，讓我的碩士班生活多采多姿。感謝我的父母親以及兩位姊姊，沒有你們的支持我不會走上研究所這條道路。另外感謝我的女友，在我的碩士班生涯裡互相扶持，在各自的道路上持續努力不懈。

最後感謝系上的老師和系辦公室裡的學長姐，有你們的陪伴讓我碩士生涯走的順順利利。

## 目錄

摘要 .....	1
Abstract .....	2
致謝 .....	3
第一章 緒論 .....	8
1.1. 研究動機與背景 .....	8
1.2. 研究目的 .....	9
1.3. 論文架構 .....	10
第二章 背景知識與相關研究 .....	11
2.1. 背景註冊法(Background Registration) .....	13
2.2. 高斯混合模型(Gaussian Mixture Model) .....	17
2.3. 編碼簿演算法(Codebook Algorithm) .....	19
2.3.1 編碼簿理論介紹 .....	20
2.3.2 編碼簿背景資訊建立 .....	20
2.3.3 色彩飽和度 .....	23
2.3.4 移動物體偵測 .....	25
2.4. 區塊式主體色背景模型(Block-based Major Color Algorithm) .....	26

2.4.1.	背景模型建立 .....	26
2.4.2.	前景偵測移動物體 .....	29
第三章	分群式背景色彩演算法物件切割系統 .....	31
3.1.	分群式背景色彩模型建立 .....	31
3.2.	前景移動物體偵測 .....	35
第四章	實驗結果 .....	39
4.1.	實驗環境與測試影片 .....	39
4.2.	測量方式 .....	39
第五章	結論與未來展望 .....	46
參考文獻	.....	47

## 圖目錄

圖 2.1 背景註冊流程圖.....	14
圖 2.2 物件遮罩在移動物體像素上 .....	16
圖 2.3 Codebook 色彩模型 .....	24
圖 2.4 原始影像 .....	29
圖 2.5 偵測目標 .....	29
圖 2.6 編碼簿偵測結果 .....	29
圖 4.1 在 Waving_Tree 上的前景偵測移動物體結果 .....	41
圖 4.2 分群式背景色彩演算法在 Waiving Tree 第 243 幀到第 258 幀的 Similarity 比較 .....	42
圖 4.3 三種演算法在 Waiving Tree 第 243 幀到第 258 幀的 Similarity 比較.....	44

## 表目錄

表 1 偵測移動物體之 Similarity 比較..... 43

表 2 三種演算法 Running time 與 Memory usage 比較..... 45

# 第一章 緒論

## 1.1. 研究動機與背景

社會安全一直是非常受到重視的議題，有如今年發生的波士頓和高鐵的炸彈客事件，無辜的百姓無法在發生案件之前免除災難；或是前年的日本大海嘯，因為監控系統無法快速的告知居民逃難而造成許多的傷亡；還有高齡化社會所帶來的老人居家監護問題，使得電子式監控系統在現今的社會中已經受到非常大的重視。如果以上的問題都能在第一時間偵測到涉及社會安全的資訊給警方或是相關的單位，就有機會可以減少災害的產生。於是我們希望能夠透過電子式的監控系統來偵測出這些危險的訊號，達到快速的通知給外界來救援。因此在現代有許多的都會區架設了監視器來做為保全系統，再搭配上 24 小時輪班制的人力，可以有效的解決許多社會安全的問題。電子式監控系統已經大量運用在各個地方，例如停車場、百貨公司、火車站等等多種移動物體資訊的地方都會用到。傳統的電子式監控系統是會讓一至二位監控人員透過多螢幕的放式偵測監視器的影像，經過人為的判斷來解讀是否發生危害社會安全的情況。但如果沒有配合著智慧型的監控系統下往往會發生人為的疏忽。例如影像中有可疑的人在某塊區域移動時，或是有不明的物體放置在陰影處下[1-3]等等，這些情況我們都

可以用電子監控系統來偵測出來。利用智慧型的監控系統可以在偵測出危險形況發生時第一時間提供給監控人員警示，減少人為上的疏失。

## 1.2. 研究目的

電子式型監控系統的透過監視器所錄下的畫面中偵測背景資訊來取得可疑的移動物體資料，然後再把得知移動物體的資訊發出警訊給監控人員，讓他們可以更快速的通報給外界。因此許多監視系統會錄製大量背景相關的資訊，而常常造成運算速度變慢或是記憶體不足[4-8]。許多背景切割的演算法雖然抓取移動物體的成效不錯，但因為需要大量的背景資訊來建置，造成過多運算量和消耗記憶體空間，隨之監控系統的成本也跟著提高。而此本篇論文提出了一個新的背景切割演算法，可以快速的偵測前景物體的同時又減少所消耗的記憶體，還能夠準確的抓取前景中的移動物體，如此一來可以用運算速度較低的處理器和較少的記憶體來達到減少電子式監控系統成本的目的。

### 1.3. 論文架構

本篇論文主要分為五章。每章概述如下：第一章為研究背景與目的的說明；第二章為背景模型建構之相關研究，介紹傳統及最近的背景模型建構演算法；第三章為使用分群背景演算法流程介紹；第四章為實驗結果，與其他演算法做效能比較；第五章為結論以及未來展望。

## 第二章 背景知識與相關研究

背景建構在移動物體偵測上是非常重要的環節，因為會影響到抓取移動物體的準確率。但是在建構背景模型會遇到許多問題，Toyama[5]等人就提出比較著名的問題點：

### 1. 靜態的移動物件：

當移動物件靜置於一段時間後，我們應該把它視為背景資訊。例如在停車場中如果有車輛從停車場外停進停車場內後，應該過一段時間就該把該車輛視為背景[6]。

### 2. 光影的變化：

白天和夜晚的光影變化會造成像是像素值等有所改變，而一個好的背景濾除演算法必須克服此現象，把有週期性變化的光影變化視為背景資訊裡，而不是前景資訊[7-8]。

### 3. 強烈的光線

在室內的時，常常遇到有關燈或是開燈的情況。如果光影變化的程度太大時會造成背景資訊的誤判，所以必需利用到梯度資訊來克服。因為梯度資訊和色彩資訊是沒有關聯的，所以光影變化大的時候不會受到影響。

#### 4. 動態背景

在戶外的監控系統中常常會遇到下雨或是颶風的情況，所以會造成像是飄逸的樹葉、水的漣漪、搖晃的樹等等情形。而這些原本是背景資訊的物件因為有移動所以常常會被誤判，所以在訓練背景資訊時要把以上的情況歸類在背景模型當中。

#### 5. 偽裝

當移動物體上有許多和背景中顏色相似的地方可能會造成誤判。例如在戶外時身上穿著的衣服和路邊的花草樹木顏色相近時，在透過監視器的螢幕上可能會造成重疊的狀況，所以背景模型中應該不能把移動物體中的色彩與背景中的色彩混淆。

#### 6. 陰影

在光線下往往會讓物件造成影子，而影子應該是屬於背景中的資訊。但是在做移動物體偵測時就會造成把陰影判斷成前景資訊，所以在前景移動物體偵測時把陰影移除是必要的[10]。

#### 7. 訓練背景資訊的適應性

一個好的背景濾除演算法能夠適用於各種場景的變化，而非單獨幾種狀況下可以使用。例如在室內中固定的背景可以準確地抓取移動

物體，在戶外中動態的背景也應當可以做到相同的效果。

本章介紹幾種背景模型建構的方法，第一節主要是介紹傳統的背景模型建構方法，其中包含了背景註冊法和高斯混合模型演算法。第二節是介紹編碼簿演算法，主要是用來處理複雜背景或動態背景的移動物體偵測。第三節是介紹以主體色(Major Color)為基礎的背景擷取方法，利用區塊式的尋找影像序列中的背景像素，再加上門檻值的篩選，得到正確的背景資訊。

## 2.1. 背景註冊法(Background Registration)

背景註冊法是由 S. Y. Chien[21]等人所提出，其概念為利用背景註冊的方式去統計在訓練期間幀中的可能的移動物體像素，可以得知哪些像素重複出現的次數高低。此方法主要是為了解決室內中靜態背景裡的陰影和雜訊。為了提升背景切割效果，於是在訓練張數期間加入背景註冊法來改善效果。

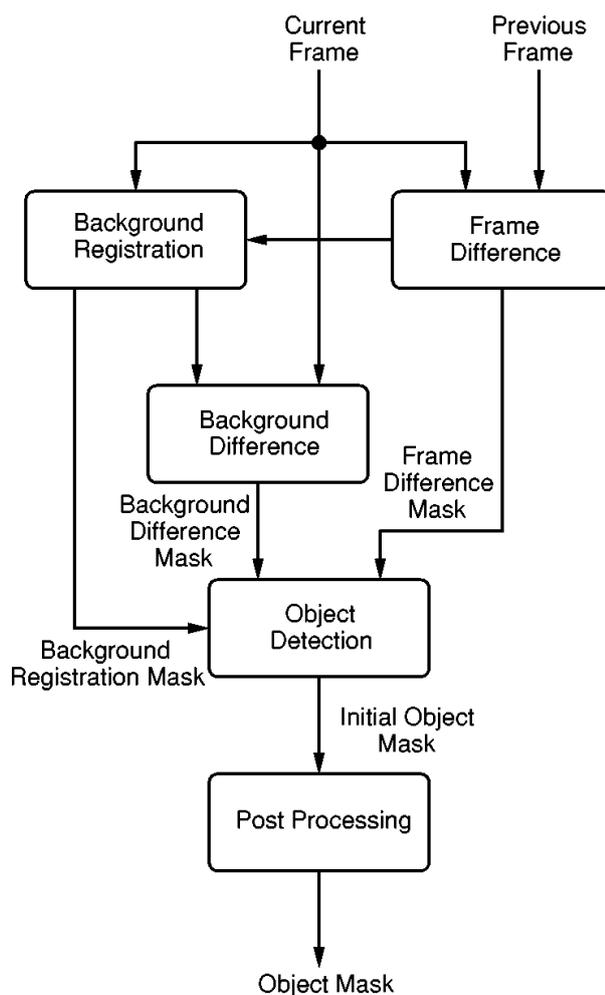


圖 2.1 背景註冊流程圖

圖 2.1 中的流程圖是背景濾除流程圖，可以分為以下幾個步驟：

### 步驟一：Frame Difference

在訓練期間，把目前的幀與前一張幀相同位置的像素做相減，可以得到一組差異值，如式(1)所示。得到差異值後做門檻值( $TH$ )計算，當差異值大於或等於門檻值的時候我

們認為此區有可能為前景移動物體，所以給值 1 來統計，  
如式(2)所示。

$$FD(x, y, t) = |I(x, y, t) - I(x, y, t - 1)| \quad (1)$$

$$FDm(x, y, t) = \begin{cases} 1 & \text{if } FDm(x, y, t) \geq TH \\ 0 & \text{if } FDm(x, y, t) < TH \end{cases} \quad (2)$$

### 步驟二：Background Registration

背景註冊是步驟一中的 $FDm$ 做比較，如果 $FDm$ 等於 0，則  
 $SI(x, y, t)$ 累加，反之則為 0，如式(3)所示。而累加 $SI(x, y, t)$ 到  
門檻值 $Fth$ 時，我們就認為此像素為背景，並把此像素設為 1  
以便與背景像素做對比，如式(4)所示。

$$SI(x, y, t) = \begin{cases} SI(x, y, t - 1) + 1 & \text{if } FDm = 0 \\ 0 & \text{if } FDm = 1 \end{cases} \quad (3)$$

$$BG(x, y, t) = \begin{cases} 1 & \text{if } SI(x, y, t) = Fth \\ BG(x, y, t - 1) & \text{else} \end{cases} \quad (4)$$

### 步驟三：Object Detection

當訓練的張數結束後，把步驟二所註冊過後的背景資料拿出來和做背景相減，並做門檻值( $TH$ )篩選，如式(5)所示。而相減的值大於或等於門檻值，即視為背景，反之則視為移動物體，如(6)所示。圖 1 則是物體偵測的結果。

$$BD(x, y, t) = |I(x, y, t) - BG(x, y, t - 1)| \quad (5)$$

$$BDM(x, y, t) = \begin{cases} 1 & \text{if } BD \geq TH \\ 0 & \text{if } BD < TH \end{cases} \quad (6)$$



圖 2.2 物件遮罩在移動物體像素上

雖然背景註冊法在靜態背景下的效果不差，但是遇到動態或是光影變化較大的背景時，常常會誤判成前景資訊。為了解決動態背景的問題，於是有人提出了高斯混合背景模型。

## 2.2. 高斯混合模型(Gaussian Mixture Model)

高斯混合背景常見的傳統背景濾除法之一，主要是針對動態背景資訊的判斷，由於傳統的背景資訊註冊法無法在動態背景中得到很好的效果，所以 Stauffer 和 Grimson 等人為了解決此問題，所以提出了此方法。高斯混合模型主要是利用  $k$  個高斯混合模型當作背景資訊，其中  $K$  的大小就代表著不同的背景資訊，所以每個像素中會儲存許多的背景資訊，包含會靜態和動態的背景，像搖晃的樹葉、水的漣漪等情況。

高斯混合背景模型建立：

$$P(X_t) = \sum_{i=1}^k \omega_{i,t} \cdot \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (7)$$

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{i,t}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})\right\} \quad (8)$$

式(7)為高斯分佈模型， $k$  為高斯分佈的個數，一般會將  $k$  值設定 3 到 7 之間。 $k$  值的大小代表著背景資訊的多寡，通常越大代表背景有很多組的參考模型。 $\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$  為第  $i$  個高斯分佈如式(8)， $\omega_{i,t}$  為第  $i$  個高斯分佈的權重值， $\mu_{i,t}$  是第  $t$  張影像像素第  $i$  個高斯分佈的平均值， $\Sigma_{i,t}$  為

其對應的標準差。  $X_t$  為第  $t$  張影像像素值。

物體偵測與背景模型更新：

針對新的影像像素值  $X_{t+1}$ ，與背景模型該位置的  $k$  個高斯分佈尋找符合的高斯分佈，其判斷條件為式(9)如下：

$$|X_{t+1} - \mu_{i,t}| \leq T \cdot \Sigma_{i,t} \quad (9)$$

其中  $T$  為自訂門檻值。若該高斯分佈與比較的像素相符合，則更新高斯混合背景模型，如式(10)：

$$\begin{aligned} \omega_{i,t+1} &= (1 - \alpha)\omega_{i,t} + \alpha \\ \mu_{i,t+1} &= (1 - \rho)\mu_{i,t} + \rho X_{t+1} \\ \sigma_{i,t+1}^2 &= (1 - \rho)\sigma_{i,t}^2 + \rho(X_t - \mu_{i,t})^2 \\ \text{where } \rho &= \alpha \cdot \eta(X_{t+1}, \mu_{i,t}, \sigma_{i,t}) \end{aligned} \quad (10)$$

$\alpha$  為學習速率， $\rho$  為高斯分佈的更新速率。 $\alpha$  值越高可以越快得到背景資訊，但是也因為提高了學習速度會帶來更多容易誤判的資訊，所以  $\alpha$  的大小需要看使用者手動調整到最佳效果。再來需要注意的就是權重值  $w_k$  也要取門檻值計算，如式(11)：

$$B = \operatorname{argmin}\{\sum_{k=1}^b w_k > T\} \quad (11)$$

上式的 $B$ 代表著高斯的群組數，而 $T$ 代表著決定更新高斯的門檻值。 $T$ 的大小和式(10)中的 $\alpha$ 有同樣的問題就是設定的值越大會造成太多的高斯背景模型，反之則會只出現太少可以判斷背景資訊的高斯背景模型。而式(11)的 $b$ 代表著要最少的高斯權重值才能大於門檻值 $T$ 。

高斯混合背景模型針對亮度漸漸變化，樹木搖晃或是水的漣漪等動態的背景模型有明顯的效果，不容易造成判斷前景錯誤的情況產生。但因為高斯混合背景模型是以像素做基底(pixel-based)的計算所取得，隨之會產生許多的背景資訊。為了減少記憶體消耗及處理時間，將每個像素的高斯分佈數量 $k$ 篩選為3到7組左右[14], [22]，但這樣做會使得誤判前景與背景機率增加，尤其當背景資訊比較豐富而超過高斯分佈的數量或是移動物體長時間停留會被誤認為背景[29]。

### 2.3. 編碼簿演算法(Codebook Algorithm)

上一節的高斯混合模型演算法有效的解決動態背景的議題，但判斷比較豐富的背景資訊或是長時間的停留在背景中的移動物體有著些許的落差。於是編碼簿演算法增加了更嚴謹的篩選背景模組方式以及計

算像素出現的頻率來改善高斯混合模型演算法不足的地方。

### 2.3.1 編碼簿理論介紹

編碼簿演算法是由 Kim[16]等人所提出，其演算法是利用計算每個像素中的變化程度去作統計而得到背景資訊。由於此演算法的背景資訊是由像素作為基礎取得，所以會產生大量的背景模型，隨之也佔據大量記憶體記憶體空間，對於低成本的電子監控系統會比較大的麻煩。

### 2.3.2 編碼簿背景資訊建立

首先需要建立一個陣列  $\chi = \{x_1, x_2, \dots, x_N\}$  來儲存每個幀裡面像素的 RGB 值，其中  $N$  代表的是幀的數量。再來還有要一個陣列來儲存編碼簿的變化程度的陣列  $C = \{c_1, c_2, \dots, c_L\}$ 。每個編碼  $C_i$ ， $i = 1, \dots, L$ ，內含像素 RGB 向量  $V_i = (\overline{R}_i, \overline{G}_i, \overline{B}_i)$  和  $aux_i = \langle \check{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i \rangle$ ，其中陣列裡的參數如下：

$\check{I}, \hat{I}$ ：影像像素編碼的最小及最大亮度。

$f$ ：影像像素此編碼出現的頻率。

$\lambda$ ：此編碼出現的間隔時間長度。

$p, q$ ：影像像素此編碼第一次符合時間及最後一次符合時間。

編碼簿建立步驟如下：

步驟一：

設  $\chi = \{x_1, x_2, \dots, x_N\}$ ， $\chi$  為  $N$  個單一影像像素 RGB 向量組成的訓練序列。

設  $C = \{c_1, c_2, \dots, c_L\}$ ，為影像像素的編碼簿，總共有  $L$  個編碼，因為剛開始創建所以目前為空。

步驟二：

取得  $N$  個像素的 RGB 值和亮度，並作下列判斷：

1.1 在編碼簿  $C = \{c_i | 1 \leq i \leq L\}$  尋找符合的編碼  $c_m$ ，並且滿足條件(a)和(b)

(a)  $colordis(x_i, v_m) \leq \varepsilon_1$ ， $\varepsilon_1$ ：自訂門檻值

$$(b) \text{ brightness}\left(I, \left(I_m^{\vee}, I_m^{\wedge}\right)\right) = \text{true}$$

$\text{color}dis(x_t, v_m)$  :  $x_t, v_m$  相減後取絕對值(純量)

$\text{brightness}\left(I, \left(I_m^{\vee}, I_m^{\wedge}\right)\right)$  : 目前像素的亮度是否在判斷的範圍

內

1.2 如果  $C = \emptyset$  或沒有任何編碼符合，則  $L \leftarrow L+1$ ，並建立新的編碼

$c_L$ 。

$$v_L \leftarrow (R, G, B)$$

$$\text{aux}_L \leftarrow \langle I, I, 1, t-1, t \rangle$$

否則，則更新符合的編碼  $c_m$ ，包含  $v_m = (\overline{R}_m, \overline{G}_m, \overline{B}_m)$  和

$\text{aux}_m \leftarrow \langle I_m^{\vee}, I_m^{\wedge}, f_m, \lambda_m, p_m, q_m \rangle$ ，更新如下：

$$v_m \leftarrow \left( \frac{f_m \overline{R}_m + R}{f_m + 1}, \frac{f_m \overline{G}_m + G}{f_m + 1}, \frac{f_m \overline{B}_m + B}{f_m + 1} \right)$$

$$\text{aux}_m \leftarrow \left\langle \min \left\{ I, I_m^{\vee} \right\}, \max \left\{ I, I_m^{\wedge} \right\}, f_m + 1, \max \left\{ \lambda_m, t - q_m \right\}, p_m, t \right\rangle$$

步驟三：對每個編碼  $c_i, i=1, \dots, L$ ， $\lambda_i \leftarrow \max \left\{ \lambda_i, (N - q_i + p_i - 1) \right\}$

其中步驟二和三中更新編碼簿目的在增加判斷讓背景資訊的準確度，而步驟三則是把出現頻率少的編碼給刪除以便節省記憶體。

### 2.3.3 色彩飽和度

為了解決區域及全域亮度變化的問題，一般方法是採用色彩比率，但是在許多陰影處的區域效果不是很理想，於是編碼簿演算法決定用一種新的亮度判斷模型來判讀每個像素的光影變化是否為背景。

設有一個像素  $x_i = (R, G, B)$  和一個編碼  $c_i$ ， $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ ，如式(12)：

$$\begin{aligned}\|x_i\|^2 &= R^2 + G^2 + B^2 \\ \|v_i\|^2 &= \bar{R}_i^2 + \bar{G}_i^2 + \bar{B}_i^2 \\ \langle x_i, v_i \rangle^2 &= (\bar{R}_i R + \bar{G}_i G + \bar{B}_i B)^2\end{aligned}\quad (12)$$

利用色彩差異  $\delta$  計算如式(13)，來判斷目前像素與編碼簿的編碼資訊色彩及亮度範圍是否符合，作為我們確定目前像素是否為背景像素或前景像素的依據。

$$\begin{aligned}p^2 &= \|x_i\|^2 \cos^2 \theta = \frac{\langle x_i, v_i \rangle^2}{\|v_i\|^2} \\ \text{colordist}(x_i, v_i) &= \delta = \sqrt{\|x_i\|^2 - p^2}\end{aligned}\quad (13)$$

色彩差異方法主要是把編碼向量正規化到輸入像素的亮度中，為了檢測亮度變化，儲存了 $\check{I}, \hat{I}$  統計值。亮度變化必須限制在一定範圍內，範圍為 $[I_{low}, I_{hi}]$ ，對於每個編碼，定義如式(14)所示：

$$I_{low} = \alpha \hat{I} \quad , \quad I_{hi} = \min \left\{ \beta \hat{I}, \frac{\check{I}}{\alpha} \right\} \quad (14)$$

$\alpha: 0.4 \sim 0.7$ ， $\beta: 1.1 \sim 1.5$ ， $I_{low}$ ：最小亮度， $I_{hi}$  最大亮度

亮度函數定義如式(15)所示：

$$brightness\left(I, \left\langle \check{I}, \hat{I} \right\rangle\right) = \begin{cases} \text{true} & \text{if } I_{low} \leq \|x_t\| \leq I_{hi}, \\ \text{false} & \text{otherwise.} \end{cases} \quad (15)$$

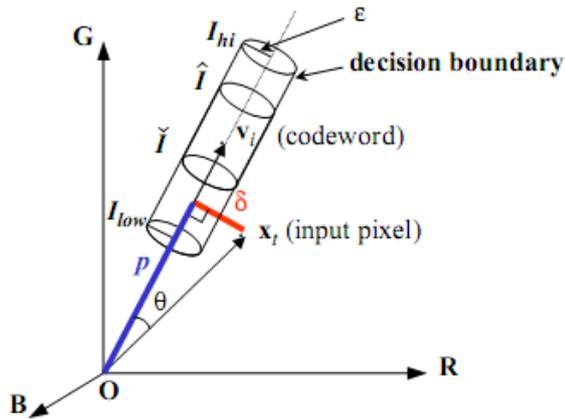


圖 2.3 Codebook 色彩模型

目前像素的向量與亮度資訊都符合判斷式後，我們就把此像素當作背景，反之我們就當作前景資訊。

### 2.3.4 移動物體偵測

利用先前建立好的編碼簿背景模型，在按照上一節提到的色彩差異和亮度範圍來對目標影像作前景偵測。因此目標影像會跟編碼簿背景模型作色彩差異和亮度範圍的比對，若色彩差異小於我們設定的門檻值並且亮度在背景模型編碼的範圍內，則我們認定該像素為背景像素，否則則認定為前景像素。

移動物體偵測步驟：

步驟一：初始化像素背景資訊

$$x = (R, G, B), I \leftarrow \sqrt{R^2 + G^2 + B^2}$$

步驟二：開始比對訓練後的幀中的像素資訊與編碼簿模型

$$\cdot \text{colordis}(x, c_m) \leq \varepsilon_2, \varepsilon_2: \text{自訂門檻值}$$

$$\cdot \text{brightness}\left(I, \left\langle \hat{I}_m, \hat{I}_m \right\rangle\right) = \text{true}$$

步驟三：如果目前影像像素與編碼簿模型相符合，即是背景資訊，反之則視為前景中的移動物體

$$BGS(x) = \begin{cases} \text{foreground} & \text{if there is no match} \\ \text{background} & \text{otherwise.} \end{cases}$$

$BGS(x)$ ：經過編碼簿背景模型比對之後的結果，也就是偵測到物體後的資訊

## 2.4. 區塊式主體色背景模型 (Block-based Major Color Algorithm)

本節要介紹 Tsai[9]等人的以區塊式的主體色為基礎的前景偵測演算法。主體色演算法藉由區塊式的運算可以達到增加取得背景資訊的速度，讓背景資訊拆開為許多區塊，然後區塊內會有許些的背景主體色來當作背景模型。相較於傳統用像素基底的演算法，主體色演算法所耗費的記憶體就來得少許多。

### 2.4.1. 背景模型建立

步驟一：創立群組

先創建一個主體色的群組 $M$ ，裡面包含的有主體色的種類數 $s$ 、主體色像素 $m$ 與影像內像素的符合次數 $MC$ 。我們先創立一個群組來存放主體色資料，內容包含了主體色的像素、與背景的符合次數與主體色的數量，如式(16)所示。

$$M = (m(r), mc(r)) \quad 1 \leq r \leq s \quad (16)$$

## 步驟二：取得主體色資訊

讀取第一張幀的像素資訊後，我們開始比較區塊內像素的差異值，由第一個像素作為初始點，其他像素做比較，如式(17)所示，最後可以得到  $n \times n - 1$  個差異值( $dist$ )，其中  $n$  代表區塊大小。得到這些差異值後，我們會取最小的差異值做門檻值( $th$ )篩選，如式(18)所示。如果大於等於門檻值，則更新主體色資訊，如式(19)所示；反之則新增一組主體色群組，如式(20)所示。

$$dist(r) = \sqrt{I(x, y) - I(x, y + 1)} \quad (17)$$

$$BGM(p) = \begin{cases} \min(dist(r)) \geq th & \text{update BGM} \\ \min(dist(r)) < th & \text{new a major color} \end{cases} \quad (18)$$

$$m(p) = \alpha \times m(p) + (1 - \alpha) \times I(x, y)$$

$$MC(p) = MC(p) + 1 \quad (19)$$

$$s = s + 1, m(s) = I(x, y), MC(s) = 1 \quad (20)$$

步驟三：更新主體色資訊

有了第一張幀的主體色的資料後，我們將持續同步驟二取得並比較下一張幀中的背景資訊，直到訓練張數結束。

等到訓練的張數結束後，我們的主體色背景模型就建置完成。為了要增加背景資訊的準確性和記憶體空間，我們在訓練結束後把符合次數比較少的主體色資訊給刪除，如式(21)所示。

$$M = \begin{cases} MC(r) \leq TH & M = M - m(r) \\ otherwise & \end{cases} \quad (21)$$

## 2.4.2. 前景偵測移動物體

在 2.3.1 節中，我們已經得到主體色的背景模型，就可以開始和訓練過後的幀開始偵測前景中的移動物體資訊。當背景資訊都取得之後，即可開始偵測物體資訊。

步驟一：

我們先建立一紀錄主體色與幀中像素符合的次數記數表  $BMC$ 。

步驟二：

開始讀取訓練後幀中的像素值，如果像素和主體色的差異值 ( $dist$ ) 小於或等於門檻值 ( $th$ )，我們就會增加記數表的次數，如式(22)所示。

$$if \ dist(r) \leq th\_d, BMC = BMC + 1 \quad (22)$$

步驟三：

如果  $BMC$  的數目大於或等於門檻值  $TH\_B$ ，我們就把此區塊內的像素當作背景，為了要與移動物體的像素做對比我們會把此像素變為 0；反之，其他小於門檻值的像素我們就做為移動物體像素，並把其像素設定為 1，如式(23)所示。

$$FO = \begin{cases} 1 & \text{if } BMC \geq TH\_B \\ 0 & \text{else} \end{cases} \quad (23)$$



圖 2.4 原始影像

圖 2.5 偵測目標

圖 2.6 主體色偵測結果

圖 2.4 是幀中的原始影像，而圖 2.5 是原始影像中的移動物體，圖 2.6 是主體色演算法所偵測的移動物體。

主體色的演算法雖然可以快速的建置背景模型資訊，但是還是產生出不少的背景模型資訊，而本文中所提出的新主體色演算法就有效的減少背景資訊的資料量，且也能夠準確的抓取前景中的移動物體資訊。

### 第三章 分群式背景色彩演算法物件切割系統

此章節會介紹分群背景演算法如何切割出前景的移動物體。第二章當中提到編碼簿演算法雖然能有效的解決光影變化和動態背景的問題，但是所需要的背景資訊實在太過龐大，常常造成計算量過大以及記憶體不足的問題；而主體色演算法雖然可以快速的擷取出背景模型，但是建置出的背景模型資料還是不小。分群背景演算法則是達到快速計算背景資訊和擷取移動物體上有不錯效果，以下會介紹分群背景演算法如何擷取前景當中移動物體的像素

#### 3.1. 分群式背景色彩模型建立

傳統的演算法都是針對影像中每個像素點來得到背景特徵等相關資訊，而分群式背景演算法則用區塊式的計算該區域內的像素的背景資訊。選擇區塊式計算的原因有兩個：第一，我們認為像素的變化往只會在某些區域裡改變，與其一個個的紀錄像素的變化程度，還不如區域性的記載代表性的變化向量。第二，區塊式的運算可以加速我們在取得背景資訊，並建置出背景模型。所以在一開始我們採用 $N \times N$ 個區塊來採集背景資訊，而 $N$ 的大小可以是4或是8，因切割的效果而定。依此方式每一個幀可以分割為 $i$ 個行與 $j$ 個列的區塊，因此每一幀會有

$i * j$ 個區塊。接著就在區塊內用色彩分群的方式來取得背景資訊，我們把像素分為 $M$ 群， $M$ 可以分為8、16、32等，也是因實驗過後的效果好壞來決定多少，接著就開始來一一統計其像素的分布情況。從統計的情況就可以得知像素主要的色彩為何，而當作我們分群背景色彩資訊。以下幾個步驟會說明如何建置背景模型：

步驟一：將目前幀中區塊內的像素值做二值化

一開始我們先把幀中區塊內的像素值二值化，也就是RGB值轉成灰階值，如式(24)。

$$I(i \times N + x, j \times N + y) = 0.2839 \times R + 0.5870 \times G + 0.1140 \times B \quad (24)$$

$(x, y)$ ：目前區塊內的像素位置      $R, G, B$ ：像素中R、G、B值

步驟二：統計幀的像素分布情況

我們把區塊中的像素分為 $M$ 群，所以每個群組內會有 $(256/M)$

個像素，圖 6 的範例是用 8 群的色彩像素做分群。

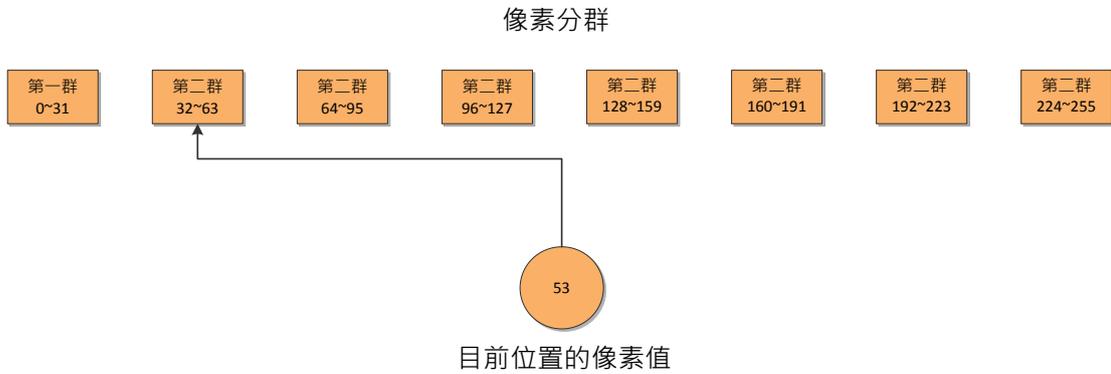


圖 3.1 以 8 群像素分類為例

把區塊內的像素分群後，開始進行各群的像素統計，把像素的數目一一累加，如式(25)。

```

for x = 0 to (N - 1)
  for y = 0 to (N - 1)
     $g = \left[ \frac{I(x,y)}{M} \right] + 1$ 
     $m(i,j,g) = m(i,j,g) + 1$ 
  end
end
end

```

(25)

$g$  : 群組的編號     $m$  : 區塊內像素的分群集合

步驟三：取得分群背景背景資訊

統計完第一張幀中區塊內的色彩分布後，我們將繼續重複步驟一和二直到訓練幀數結束為止。建置背景模型的過程中我們會利用訓練幀的方式來取得背景像素，藉由訓練的方式可以參考訓練中的幀的群組分佈情形，用一定數量的像素來取得背景資訊，使建構背景模型的過程能夠更加準確。得到訓練期間所有幀中像素分布的總數量後我們會把這些總數量除以訓練幀數的數目，就可以得到一組訓練期間的分群的像素數，也就是平均分佈在訓練張數的像素群組，此群組就會當作分群背景色彩演算法的背景模型，如式(26)所示。

*for*  $g = 1$  *to*  $M$

$$m(i, j, g) = \frac{m(i, j, g)}{t}$$

*end*

(26)

$t$  : 訓練影像的幀數

## 3.2. 前景移動物體偵測

得到分群色彩背景模型後，我們就可以開始擷取移動物體資訊。透過背景像素與要偵測影像中的像素做比較，就可以找到移動物體資訊，以下步驟會說明擷取移動物體資訊的過程：

步驟一：將目前幀中區塊內的像素做分群

與初期建置背景模型一樣，一開始先把幀中區塊內的像素做二值化，如式(24)。再來就要把幀中每個區塊的像素做分群，我們把目前幀中區塊的群組分佈與背景像素的分佈做比較，並統計其符合背景像素的次數 $MC$ 。如果區塊內像素分布的色彩分群與背景像素的分群相同，也就是說目前區塊內的像素分群情況與背景像素的分群情況和相符，並做次數的累加，如式(27)。

*for*  $x = 0$  *to*  $(N - 1)$

*for*  $y = 0$  *to*  $(N - 1)$

$$g = \left\lfloor \frac{I(x,y)}{M} \right\rfloor + 1$$

*if*  $m(i, j, g) > 0$

$$MC(i, j, g) = MC(i, j, g) + 1$$

```

        end
    end
end

```

(27)

*MC*：區塊內像素分群與背景像素分群符合的次數

統計完*MC*的值後，區塊內的像素的分佈情況會與背景像的分佈情況再做一次比較，我們將*MC*的值與背景像素中分群裡的值取較小一方做累加，取較小的值做累加用意在避免計算到多餘的符合次數，也就會更精確的取得分群的情況，如式(28)。

$$FMC(i, j) = \min( MC(i, j, 1), m(i, j, 1) )$$

*for g = 2 to M*

$$FMC(i, j) = \min( MC(i, j, g), m(i, j, g) ) + FMC(i, j)$$

*end*

(28)

*FMC*：區塊內 *MC* 與背景分群比較後正確吻合背景值的累加次數

### 步驟三：取得前景中移動物體資訊

從步驟二中可以得到區塊內像素與背景像素的群組符合次數  $FMC$ ，而我們會將這些符合次數做門檻值的計算。我們用手工的門檻值  $th_b$  做篩選(文中的演算法設定為 6)，如果  $FMC$  大於門檻值，我們就將其區塊內的像素視為背景，反之則視為前景中的移動物體。而為了與前景做比對所以會把此區塊像素設定為黑色，也就是將像素值設定為 1；反之符合次數小於或等於門檻值，我們就視為移動物體資訊，並把區塊內的像素設定為白色，像素值設定為 0，如式(29)所示。

```

for i = 0 to ( $\frac{row}{N} - 1$ )
    for j = 0 to ( $\frac{col}{N} - 1$ )
        if  $FMC(i,j) > th_b$ 
             $FO(i,j) = 0$ 
        else
             $FO(i,j) = 1$ 
        end
    end
end
end
end

```

(29)

*row* : 幀的行數    *col* : 幀的列數    *th<sub>B</sub>* : 手動門檻值    *FO* : 前景移動物體

結束與背景像素的比對後，我們就可以擷取出前景的移動物體資訊。

## 第四章 實驗結果

### 4.1. 實驗環境與測試影片

本研究之實驗環境使用 Matlab 2011R 於 Windows XP 作業系統來開發程式。硬體設備為 Intel Core i5 2.67GHz 處理器以及 2G 的記憶體。測試影片是從 PETS 資料庫擷取(<ftp://pets.rdg.ac.uk>)，我們使用的是 Waving\_Tree。影片內容是有人走過在風中搖擺的樹，這個場景相當適合給處理動態背景演算法來測試。影片大小為 160x120 像素並以 15 fps 的速度撥放，影像序列為 286 張影像，BMP 格式。

### 4.2. 測量方式

本文中採用 Similarity、執行速度(running time)與記憶體使用量(memory usage)三種方式來判斷抓取移動物體的效果好與壞。Similarity 包含了 True Positive(TP)、False Positive(FP)、True Negative(TN)和 False Negative(FN)等。

TP：預測正確，實際上結果是正確

FP：預測錯誤，實際上結果是正確

TN：預測正確，實際上結果是錯誤

FN：預測錯誤，實際上結果是錯誤

而 Similarity 是判斷背景以及前景中移動物體成功的機率，如式 (30) 所示。

$$\text{Similarity} = \frac{\text{TP} + \text{FN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (30)$$

算出抓取移動物體後的幀的 Similarity 值就可以得知其演算法品質的效果如何，圖 4.1 會列出各種演算法偵測前景移動物體的結果。執行速度的計算方式是讓影片中每個幀都做前景的移動物體偵測，並且以黑色及白色來分別代表背景以及前景的像素來標記，而所有過程的運算時間。記憶體使用量的計算方式是以背景資訊的使用量來當作基準，我們以像素為基底來算出每個在幀中的像素必須要儲存記憶體多寡來作比較，如果是區塊式為基底的演算法則需算出區塊內平均每個像素會儲存多少背景資訊來作為評估記憶體使用量的大小。

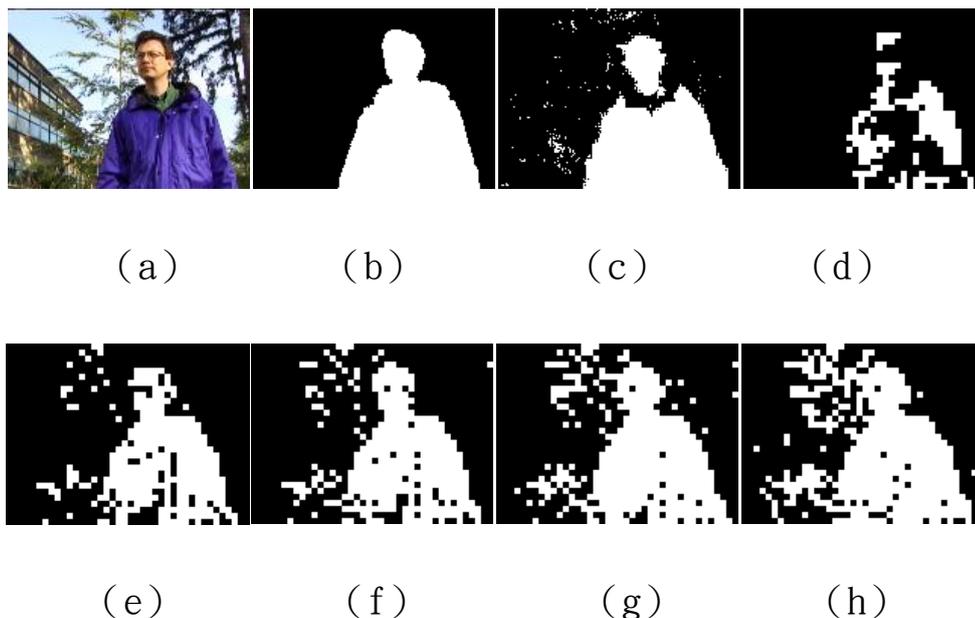


圖 4.1 在 Waving\_Tree 上的前景偵測移動物體結果 (a)Frame#249 (b)Ground Truth (c)Codebook (d)Major Color (e)Group-based Color 16 sets with 4x4 block (f)Group-based Color 8 sets with 4x4 block (g)Group-based Color 16 sets with 8x8 block (h)Group-based Color 8 sets with 8x8 block

圖 4.1(c)和(d)是編碼簿演算法以及主體色演算法對 Waving\_Tree 影片裡第 249 幀所執行過後的結果，採用的原因是第 249 幀是影片中出現移動物體佔據背景最大面積的時間點。圖 7(e)和(f)分別由 16 分群以及 8 分群在 4x4 區塊所執行的結果，圖 7(g)和(h)分別由 16 分群以及 8 分群在 8x8 區塊所執行的結果。

我們將圖 4.1(e)、(f)、(g)與(h)的結果來做測量，讓數據來得知哪一種分群式演算法擷取移動物體的效果較好。

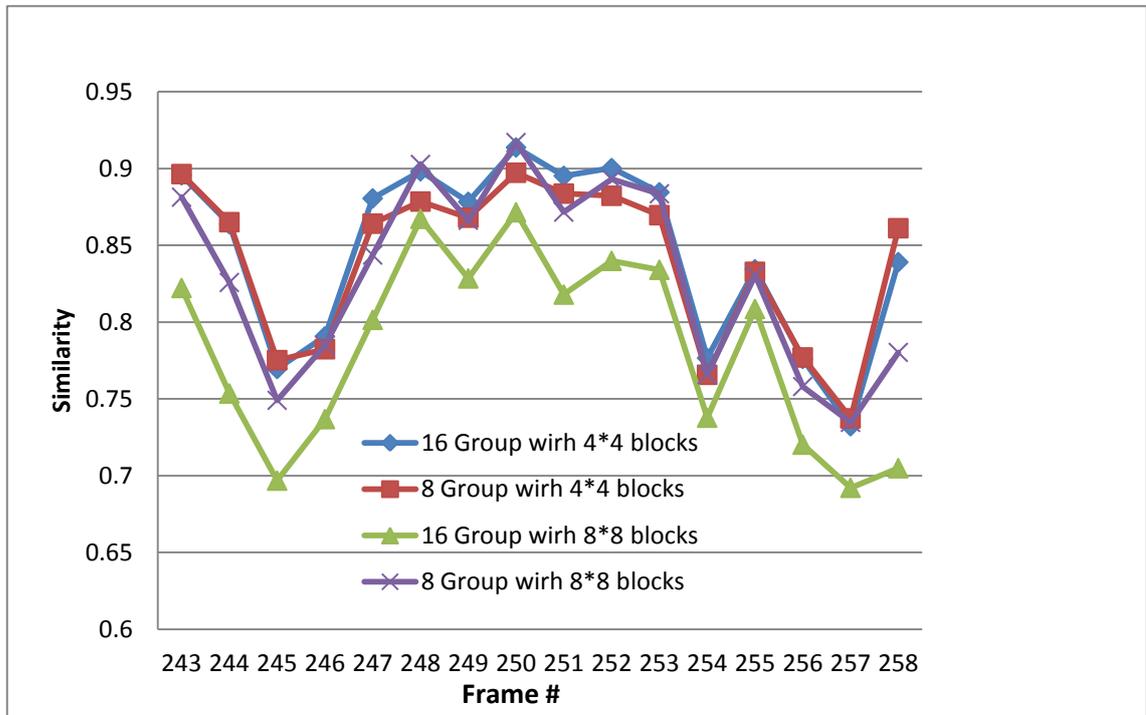


圖 4.2 分群式背景色彩演算法在 Waiving Tree 第 243 幀到第 258 幀的 Similarity 比較

圖 4.2 可以得知在偵測移動物體時 Similarity 的變化，我們採 Waiving Tree 中第 243 張幀到第 258 張幀來統計其原因是在影片中的這段時間才有移動物體出現。而經過統計後可以發現，8 分群配合著 4x4 區塊和 16 分群配合著 4x4 區塊的效果較佳，如表 1 所示。

表 1

分群式背景色彩演算法測量結果比較

	Group-based color(16 sets with 4x4 block)	Group-based color(8 sets with 4x4 block)	Group-based color(16 sets with 8x8 block)	Group-based color(8 sets with 8x8 block)
Similarity average	84.55%	83.97%	78.33%	78.03%
Running time	4.01sec	3.25sec	4.23sec	3.67sec
Memory usage	2bytes/pixel	1byte/pixel	2bytes/pixel	1byte/pixel

從表 1 可以得知，4x4 區塊大小的 Similarity average 比較 8x8 區塊大小較佳，而又以 8 分群的執行速度和記憶體使用量都表現的最好。綜合圖 4.2 與表 1 的結果，我們將會用 8 分群配合 4x4 區塊大小的分群式背景色彩演算法來和編碼簿演算法以及主體色演算法來比較，首先我們比較的是偵測移動物體的 Similarity，如圖 4.3 所示。

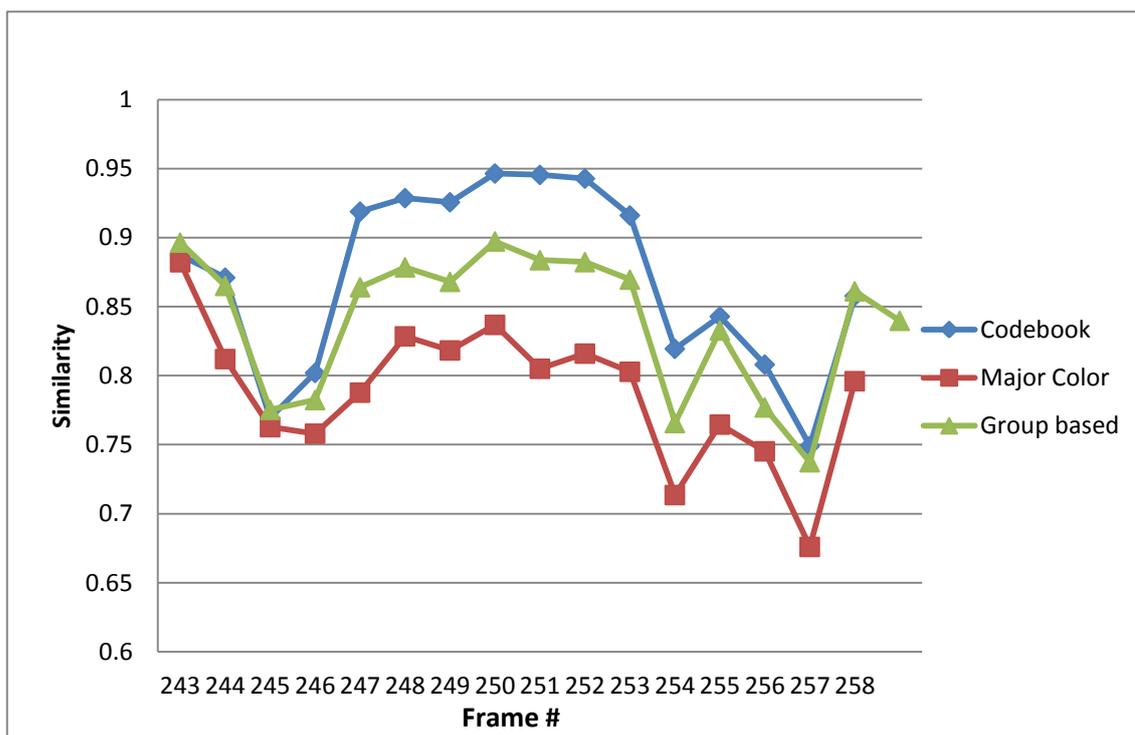


圖 4.3 三種演算法在 Waiving Tree 第 243 幀到第 258 幀的 Similarity 比較

從圖 4.3 可以看出編碼簿演算法的效果最好，其次是分群式演算法，最後才是 Major Color 演算法。表 2 中則可以看出執行速度與記憶體使用量在分群式演算法有顯著地優於編碼簿演算法及主體色演算法。

總結以上圖表可以知道，本文的分群式演算法在執行速度和記憶體使用量的優勢，並且有不錯的擷取移動物體的結果。

表 2

三種演算法 Running time 與 Memory usage 之比較

	Codebook	Major Color	Group-based color(8 sets with 4x4 block)
Similarity average	85.77%	79.60%	83.97%
Running time	10.23sec	5.13sec	3.25sec
Memory usage	60 bytes/pixel	6 bytess/pixel	2 bytes/pixel

## 第五章 結論與未來展望

在本篇論文中我們可以得知使用 8 分群配合著 4x4 區塊大小的分群式演算法在運算速度以及記憶體使用量的表現為最好的組合，而分群式演算法可以達到快速而且節省記憶體原因在於我們在建立背景模型所需要的運算量比較少。分群式演算法在建立背景的過程中只需要分類像素所在的群組，而非統計像素值的變化或是時間戳記等；而傳統的演算法則需要另外新增一組群組來統計單一像素或是區塊內的像素變化或是光影變化，甚至需要更新背景資訊等等需求來建構背景模型。本文的分群式演算法為了講求建構背景模型的簡易性所以沒有採用背景更新，但如果在硬體設備有足夠的記憶體空間下可以考慮加入，藉此增加擷取移動物體的準確性。本文中的演算法的實現平台是在個人電腦上，未來希望可以實作在嵌入式系統或是 IP camera 等市場比較主流的平台，來達到解決社會安全的與減少社會成本的目的。

## 參考文獻

- [1] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, “Improving Shadow Suppression in Moving Object Detection with HSV Color Information,” IEEE Int’l Conference on Intelligent Transportation Systems, pp. 334–339, 2001.
- [2] J. Menendez and S. A. Velastin, “A method for obtaining neural network training sets in video sequences”, Proc. 3rd IEEE Int. Workshop. Visual Surveillance, 2000.
- [3] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting Moving Objects, Ghosts and Shadows in Video Streams,” In IEEE Trans. Patt. Anal. Mach. Intell, Vol 25, pp. 1337–1342, 2003.
- [4] T. Horprasert, D. Harwood, and L. S. Davis, “A statistical approach for real-time robust background subtraction and shadow detection,” IEEE Int. Conf. on Computer Vision, Frame Rate Workshop, pp. 1–19, 1999.
- [5] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: principles and practice of background maintenance,” International Conference on Computer Vision, pp. 255–261, 1999.
- [6] A. Elgammal, D. Harwood, and L. S. Davis, “Non-parametric model for background subtraction,” European Conference on Computer Vision, pp. 751–767, 2000.
- [7] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: real-time tracking of the human body,” IEEE Trans. Pattern Anal. Machine Intell, Vol. 19, No. 7, pp. 780–785, 1997.
- [8] P. Viola and M. Jones, “Robust Real-Time Object Detection,”

- International Journal of Computer Vision, 2001.
- [9] Wen-Kai Tsai, Ming-Hwa Sheu, and Chung-Chi Lin “Block-Based Major Color for Foreground Object Detection on Embedded SoC Platform ” IEEE Embedded Systems Letters, pp.49-52, 2012.
- [10]T. Huang, J. Qiu, T. Sakayori, S. Goto and T. Ikenaga, “Motion Detection Based On Background Modeling And Performance Analysis For Outdoor Surveillance,” IEEE international Conference on Computer Modeling and Simulation, 2009.
- [11]C. Stauffer and W.E.L. Grimson, “Learning Patterns of Activity Using Real-Time Tracking,” IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 747-757, 2000.
- [12]P. Dickinson and A. Hunter, “Scene modeling using an adaptive mixture of Gaussians in color and space,” in Proc. IEEE Adv. Video Signal Based Surveillance, pp. 64 - 69, 2005.
- [13]H. Wang and D. Suter, “A re-evaluation of mixture-of-Gaussian background modeling,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Vol. 2, pp. 1017 - 1020, 2005.
- [14]Y. L. Tian, M. Lu, and A. Hampapur, “Robust and efficient foreground analysis for real-time video surveillance,” in Proc. IEEE Comput. Vision Pattern Recognition, Vol. 1, pp. 1182 - 1187, 2005.
- [15]D. Turdu and H. Erdogan, “Improving Gaussian Mixture Model based Adaptive Background Modeling using Hysteresis Thresholding,” Signal Processing and Communications Applications, pp. 1- 4, 2007.
- [16]K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Real-Time Foreground - Background Segmentation Using Codebook Model,” Real-Time

- Imaging, Vol. 11, No. 3, pp. 172–185, 2005.
- [17]F. Porikli, “Detection of Temporarily Static Regions by Processing Video at Difference Frame,” *Advanced Video and Signal Based Surveillance*, pp. 236–241, 2007.
- [18]J. Ding, M. Li, K. Huang and T. Tan, “Modeling Complex Scenes for Accurate Moving Objects Segmentation,” *Computer Vision*, pp. 82–94, 2010.
- [19]C. C. Chiu, M. Y. Ku, and L. W. Liang, “A Robust Object Segmentation System using a Probability-based Background Extraction Algorithm,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, No. 4, 2010.
- [20]A. Prati, R. Cucchiara, I. Mikic, and M.M. Trivedi, “Analysis and Detection of Shadows in Video Streams: A Comparative Evaluation,” *IEEE International Conf. Computer Vision and Pattern Recognition*, Vol. 2, No. 2, pp. 571–576, 2001.
- [21]D. Wang, T. Feng, H. Shum, and S. Ma, “A novel probability model for background maintenance and subtraction,” *The 15th International Conference on Vision Interface*, 2002.
- [22]S. Chien, S. Ma, and L. Chen, “Efficient Moving Object Segmentation Algorithm Using Background Registration Technique,” *IEEE Transaction Circuits And Systems For Video Technology*, Vol. 12, No. 7, 2002.
- [23]D. S. Lee, J. J. Hull, and B. Erol, “A Bayesian framework for Gaussian mixture background modeling,” *IEEE Int. Conf. on Image Processing* 2003.
- [24]F. Porikli and O. Tuzel, “Human body tracking by adaptive background

- models and mean-shift analysis,” IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, pp. 1-9, 2003.
- [25]A. Monnet, A. Mittal, N. Paragios, and R. Visvanathan, “Background modeling and subtraction of dynamic scenes,” Proc. IEEE Int. Conf. Computer Vision, vol. 2, pp. 1305 - 1312. 2003.
- [26]S. Y. Chien, Y. W. Huang, B. Y. Hsieh, S. Y. Ma, and L. G. Chen, “Fast Video Segmentation Algorithm with Shadow Cancellation, Global Motion Compensation, and Adaptive Threshold Techniques,” IEEE Trans. on Multimedia, Vol. 6, No. 5, pp. 732-748, 2004.
- [27]M. Piccardi, “Background Subtraction Techniques: A Review,” IEEE Int. Conf. on Systems, Man and Cybernetics, Vol. 4, pp. 3099-3104, 2004.
- [28]D. C. Tseng, C. W. Lin , and C. M. Ling, “Motion Object Detection and Tracking Based on Adaptive Background Subtraction,” 18th IPPR Conf. on Computer Vision, Graphics and Image Processing, 2005.
- [29]L. Maddalena and A. Petrosino, “A self-organizing approach to background subtraction for visual surveillance applications,” IEEE Trans. Image Processing, Vol. 17, No. 7, 2008.
- [30]S.T. Su and Y. Y. Chen, “Moving Object Segmentation Using Improved Running Gaussian Average Background Model,” Digital Image Computing: Techniques and Applications, pp. 24-31, 2008.