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

在複雜背景下使用主體色彩演算法之移動物體切割系統

An Object Segmentation System Design using Color Analysis with Complex Background

研究生：蔡宗穎

中華民國 一百 年 七 月

摘要

本文提出一個將影像以區域方式分割的方法,並且針對每個區域內的每個像素相互作同質性計算，來得到該區域的的主體色彩資訊，利用區域主體色彩資訊建構背景模型以偵測複雜背景下的移動物體。影像序列初期要建構背景模型，需要單張或數張只有背景的影像來擷取區域的色彩資訊以建立背景模型,並且利用影像的紋理資訊來處理複雜的背景，可以因應背景的變化來更新背景模型，克服複雜背景下的變化。傳統的編碼簿演算法在處理複雜背景情況下，雖然效果不錯，但是需要較多時間及計算複雜度。我們的方法跟其他背景模型物件分割技術相比,都能夠在複雜的背景環境下更為快速地分割出移動物體。

關鍵詞：主體色彩、移動物件分割、紋理分析、背景模型。

Abstract

This paper presents a block-based image segmentation method. The homogeneity of all the pixels in a block are calculated, and the color information of the block can be obtained. Thus, the background model can be built, and a moving objects can be detected accordingly in a complex background. In an image sequence, some background images are used to capture blocks' color information to create the background model. Their texture information is applied to process a complex background. The background model can be updated while background changes. The traditional codebook background method useds complex algorithms, and requires more time and memory. Compared to other segmentation methods, our approach with the background model can segment a moving object rapidly and successfully in a complex background environment.

Key words：major color、moving object detection、texture analysis、background model.

目錄

摘要.....I

Abstract.....II

第一章　緒論.....1

1.1　研究背景與動機.....1

1.2　論文架構.....3

第二章　背景模型建構方法.....4

2.1.　傳統的背景模型建構方法.....4

2.1.1　連續影像相減法.....4

2.1.2　高斯混合背景模型建構方法.....7

2.2　編碼簿演算法（Codebook Algorithm）.....9

2.2.1　編碼簿理論介紹.....9

2.2.2　編碼簿建立及更新.....10

2.2.3　色彩和亮度.....12

2.2.4　移動物體偵測.....14

2.3　以機率為基礎的背景擷取演算法（Probability-Based Algorithm）.....15

2.3.1　背景模型建立.....15

2.3.2　色彩分割與背景更新.....18

第三章　在複雜背景下使用主體色彩演算法之移動物體切割系統.23

3.1 主體色彩背景模型 23
3.2 主體色彩背景模型訓練及更新 27
3.3 物體偵測 30
3.4 區域紋理資訊 35
3.4.1 含區域紋理之背景模型 38
3.4.2 含區域紋理之背景模型訓練及更新 39
3.4.3 物體偵測 40
第四章 實驗結果 43
4.1 實驗環境與測試影片 43
4.2 選擇區域大小 44
4.3 效能比較 49
第五章 結論以及未來展望 57
參考文獻 58

表目錄

表 1 主體色彩演算法選擇 4*4 區域、8*8 區域對測試影片(訓練及偵測)所需時間及每秒可執行幀數 47

表 2 主體色彩演算法對測試影像序列選擇 4*4 區域、8*8 區域的平均錯誤率.....49

表 3 四種演算法對測試影像序列(訓練及偵測)所需時間及每秒可執行幀數.....55

表 4 四種演算法對測試影像序列的移動物體偵測平均錯誤率 56

圖目錄

圖 1 Codebook 色彩模型 13

圖 2　Waving_Tree 影像序列高低轉折點色彩差異直方圖 21

圖 3　搖晃的樹木固定某個區域在時間點不同下的影像 (a)Frame#001(b)Frame#005 24

圖 4 目標影像分割為 k 個 $N*N$ 區域 25

圖 5 本文演算法對測試影片建立的背景模型中初始背景和訓練過的背景(a) Waving_Tree 的初始背景(b) Waving_Tree 訓練後的背景(c) Parking_Area 的初始背景(d) Parking_Area 訓練後的背景 30

圖 6 本文演算法對 Waving_Tree 影片執行後的移動物體偵測結果 (a)original image (b)ground truth (c) proposed 35

圖 7 以 4*4 像素區域為例，用直角座標系劃分影像區域。 37

圖 8 主體色彩演算法對 Waving_Tree 影片選擇 4*4 區域、8*8 區域所執行的結果(a)Frame#249(b)Ideal Result(c)8*8 Block Result(d)4*4 Block Result 45

圖 9 主體色彩演算法對 Parking_Area 影片選擇 4*4 區域、8*8 區域所執行的結果(a)Frame#1290(b)Ideal Result(c)8*8 Block Result(d)4*4 Block Result 45

圖 10 主體色彩演算法對 Waving_Tree 影像序列（Frame#242~259）選擇 4*4 區域、8*8 區域的錯誤率 48

圖 11 主體色彩演算法對 Parking_Area 影像序列（Frame#912~990）

選擇 4*4 區域、8*8 區域的錯誤率 48
圖 12 四種演算法在 Waving_Tree 影像序列移動物體偵測的結果，依序為 Frame#246、Ideal Result、偵測結果
(a)Codebook(b)Probability-based(c)主體色彩演算法(d)主體色彩+區域紋理資訊演算法 54
圖 13 四種演算法在 Parking_Area 影像序列移動物體偵測的結果，依序為 Frame#258、Ideal Result、偵測結果。(a)Codebook
(b)Probability-based (c)主體色彩演算法 (d)主體色彩+區域紋理資訊演算法.....55
圖 14 四種演算法在 Waving_Tree 影像序列（Frame#242~259）的移動物體偵測錯誤率 55
圖 15 四種演算法在 Parking_Area 影像序列（Frame#912~990）的移動物體偵測錯誤率 56

第一章　緒論

1.1 研究背景與動機

近年來視訊監控設備的普及，相關的研究及應用也快速發展。快速即時的監控系統能夠擷取畫面把資訊傳遞給遠端查詢，也可供事後追蹤之用，可以有效保障我們的生命財產安全。但是監看時間變長之後，容易造成精神不集中，注意力下降等情況。因此自動化即時監控系統可以有效地幫助我們，不會產生漏洞。

自動化監控系統可以透過影像分析方式紀錄，將畫面中的移動物體擷取出來，現今多數研究探討主要是使用固定式攝影機，而本篇研究也是對於固定式攝影機在複雜環境中對移動物體的偵測及追蹤。

背景模型建立為自動化監控系統中第一個重要部份。背景模型建立的效能主要取決背景模型建構的技術。尤其是自然景觀對於建構背景模型會有相當難度，是因為它們通常是動態的，包括光源變換、植物的擺動、水紋波動等。背景模型建立演算法也需要能夠處理從背景中加入新的物體或是移除舊的物體。因此，移動物體的陰影和動態背景也可能造成問題[1-3]。即使在靜態的場景中，也可能因為雜訊跟攝影機抖動而產生變化。所以背景模型演算法需要能夠即時地處理。

最簡單的背景模型為假定一個像素的亮度值能用單一模式的機

率分佈來建構背景模型[4-8]，如高斯背景模型，這種背景模型沒辦法處理多重型態背景，像是擺動的樹枝。另一個為高斯混合背景模型(Gaussian Mixture Model，GMM) [9, 10]用來建構複雜非靜態背景，對於動態背景及緩慢的亮度變化有良好的強健性，但是對於快速變化的背景可能無法精確地偵測，缺點是演算法複雜度過高，若根據學習速率來更新背景變化，較低的學習速率產生的背景模型，在突然變化的背景中錯誤率較高。較高的學習速率產生的背景模型，太慢移動的前景像素又會被加入到背景模型，造成較高的假陰性。

為了解決動態複雜的背景和亮度變化的問題，本篇論文提出一種以色彩為基礎的複雜背景物體切割技術，以影像區域的主體色彩為基礎建立背景模型來處理動態背景和亮度變化，再加上紋理分析來建構背景資訊，可以有效地解決動態背景變化的問題。

1.2 論文架構

本篇論文主要分為五章。每章概述如下：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。第三章為在複雜背景下使用主體色彩演算法流程介紹。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　背景模型建構方法

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。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。第三節是介紹以機率為基礎的背景擷取方法，利用影像序列中背景像素出現的機率會高於移動物體像素的機率來建構背景模型，進而偵測移動物體的方法。

2.1. 傳統的背景模型建構方法

背景模型建構是智慧監控系統中相當重要的步驟，目的在於如何利用背景模型更為精確、快速地偵測移動物體，並且做後續追蹤、辨認…等都需要正確的偵測移動物體，其結果影響到整個監控系統的準確性。

2.1.1 連續影像相減法

連續影像相減法針對連續時間的影像或間隔固定的影像相減，但是不需事先建立無移動物體的背景影像。對影像相對應的像素值相減後取絕對值，若差異值小於門檻值則次數累加，否則次數規零。若累

加次數到達設定的門檻值，則認定該像素為背景像素。等到全部背景像素擷取成功則完成背景建立。接著再將目前影像像素與背景模型像素相減取絕對值，若差異值小於門檻值則認定該像素為背景像素，否則為移動物體像素。

背景模型建立：

步驟 1：假設 I_n 影像序列總共有 n 張，u 和 v 為像素座標。對第 i 張影像像素位置 (u,v) 的像素和第 $i-1$ 張影像像素位置 (u,v) 的像素相減並取絕對值如式(1)，若差異值 $\Delta_i(u,v)$ 小於門檻值 $Threshold_i$ 則累加次數 $CNT_i(u,v)$，否則 $CNT_i(u,v)$ 歸零。

$$\Delta_i(u,v)=\left|I_i(u,v)-I_{i-1}(u,v)\right| \qquad (1)$$

if $\Delta_i(u,v)<Threshold_i$

$CNT_i(u,v)=CNT_i(u,v)+1$

else

$CNT_i(u,v)=0$

end

步驟 2：若第 i 張影像像素位置 (u,v) 的 $CNT_i(u,v)$ 大於門檻值 $Threshold_F$ 則認定該像素為背景像素，直到全部像素位置的背景像素建立後，則背景模型建立成功。

if $CNT_i(u,v) > Threshold_F$ then

$BG(u,v) = I_i(u,v)$

end

物體偵測：

步驟 1：若目前影像像素 $I_i(u,v)$ 與背景模型像素 $BG(u,v)$ 相減後小於門檻值 $Threshold_M$，則為背景像素，否則為移動物體像素如式(2)所示：

$$\Delta_i(u,v) = |I_i(u,v) - BG(u,v)|$$

$$R_i = \begin{cases} Foreground(u,v) & ,\Delta_i(u,v) \geq Threshold_M \\ Background(u,v) & ,\Delta_i(u,v) < Threshold_M \end{cases} \quad (2)$$

R_i 為第 i 張的移動物體偵測結果

連續影像相減法所需運算量很低，速度快，適合室內場景之應用。若在室外場景，無法適應亮度及動態背景的變化，造成移動物體偵測的誤判。

2.1.2 高斯混合背景模型建構方法

Stauffer 和 Grimson 提出了高斯混合背景模型來建構一段時間內的像素強度模型，利用多個高斯分佈來記錄像素特徵值[11],[12]，因為影像像素值在動態背景下是不會固定的，通常為小幅度變動，因此適合用高斯分佈來模型化，但是像素值並不只有一個值來變動，而是某幾個值變動，像是亮度變化、陰影產生的情況而發生，因此採用多個高斯分佈來做背景模型是比較適合的情況。

高斯分佈建構背景模型：

$$P(X_t)=\sum_{i=1}^{k}\omega_{i,t}\cdot\eta(X_t,\mu_{i,t},\Sigma_{i,t}) \qquad (3)$$

$$\eta(X_t,\mu_{i,t},\Sigma_{i,t})=\frac{1}{(2\pi)^{\frac{n}{2}}\left|\Sigma_{i,t}\right|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(X_t-\mu_{i,t})^T\Sigma_{i,t}^{-1}(X_t-\mu_{i,t})\right\} \qquad (4)$$

式(3)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 5 之間，k 值越大，所需的容量也越大。$\eta(X_t,\mu_{i,t},\Sigma_{i,t})$ 為第 i 個高斯分佈如式(4)，$\omega_{i,t}$ 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(5)如下：

$$\left|X_{t+1}-\mu_{i,t}\right| \le T \cdot \Sigma_{i,t} \qquad (5)$$

T 為自訂門檻值。若該高斯分佈與新像素相符合，更新高斯混合背景模型參數如式(6)：

$$\begin{aligned} &\omega_{i,t+1} = (1-\alpha)\omega_{i,t-1} + \alpha \\ &\mu_{i,t+1} = (1-\rho)\mu_{i,t} + \rho X_{t+1} \\ &\sigma^2_{i,t+1} = (1-\rho)\sigma^2_{i,t} + \rho(X_t - \mu_{i,t})^2 \\ &where \quad \rho = \alpha \cdot \eta(X_{t+1}, \mu_{i,t}, \sigma_{i,t}) \end{aligned} \qquad (6)$$

α 為學習速率，ρ 為高斯分佈的更新速率

高斯混合背景模型優點在於記錄像素變化，針對亮度漸漸變化，樹木搖晃等背景建構模型，後來又被廣泛使用到各種背景刪減法中來處理微小重複動作及稍微亮度變化的影像。為了減少記憶體的消耗及處理時間，將每個像素的高斯分佈數量設為固定（通常為 3-5 個）[14],[22]，這可能導致有些背景像素被誤認，當它們的高斯分佈比預期來得多或是它們剛好被移動物體擋住時，並且若移動物體長時間停留會誤認為背景[29]。

2.2 編碼簿演算法（Codebook Algorithm）

上一節提到的傳統移動物體偵測方法，因為背景變化、處理時間問題，使得偵測移動物體的效果不是很理想，因此 Kim, Chalidabhongse 等人提出了 Codebook 方法[15-17]，也就是本節要介紹的編碼簿方法，建構背景模型來記錄背景的變化，包括亮度及陰影的變化，可以有效地處理多重型態背景，但是缺點在於計算量大，所需時間也長。

2.2.1 編碼簿理論介紹

編碼簿演算法採用了群集(cluster)的概念，在一段長時間的連續影像序列中來建構影像背景模型。針對影像中每個像素來個別建立編碼簿，每個影像像素的編碼簿會有一個或多個以上的編碼。利用色彩和亮度資訊來對每個像素群集分組。因此不是每個影像像素會有擁有相同數量的編碼，因為使用群集方式，使得編碼不一定會單一高斯分佈或其他參數分佈。利用群集概念，我們可以利用多個編碼來建構動態背景模型，好處在於我們可以長時間並且有限的記憶體中來擷取動態背景的動作，使得動態背景的變化可以編碼到背景模型中，並且可以適應區域及全域的亮度變化。

偵測目前影像與建構好的背景模型的色彩及亮度差異來判別背景與前景。假如該影像像素滿足下列兩點，則把該影像像素歸類為背景像素，若不滿足則歸類為前景。

（1）該像素的編碼色彩差異低於設定的門檻值。

（2）該像素的編碼亮度值在背景模型的亮度值範圍內。

2.2.2 編碼簿建立及更新

要建立背景模型之前，首先設$\chi=\{x_1,x_2,\mathrm{K},x_N\}$，$\chi$為$N$個單一影像像素 RGB 向量組成的訓練序列。設$C=\{c_1,c_2,\mathrm{K},c_L\}$，為影像像素的編碼簿，總共有$L$個編碼。每個影像像素都有一個編碼簿來表示該像素的變化。每個編碼C_i，$i=1,\mathrm{K},L$，內含像素 RGB 向量$V_i=\left(\overline{R_i},\overline{G_i},\overline{B_i}\right)$和$aux_i=\left\langle \check{I}_i,\hat{I}_i,f_i,\lambda_i,p_i,q_i\right\rangle$，內含亮度資訊和時間變數如下：

$\check{I},\hat{I}$：影像像素編碼的最小及最大亮度。

f：影像像素此編碼發生的頻率。

λ：背景模型訓練期間，此編碼還未重新出現最長時間間隔。

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。

在背景訓練期間，每個影像像素向量x_t在時間t中會互相比較彼此的編碼簿，當比較過後的編碼符合要求時，會當作背景模型的編碼，為了確認哪些編碼才是最佳的，採用色彩差異和亮度範圍作為參考。

編碼簿建立步驟：

1. 設$\chi=\{x_1,x_2,\mathrm{K},x_N\}$，$\chi$為$N$個單一影像像素 RGB 向量組成的訓練序列。設$C=\{c_1,c_2,\mathrm{K},c_L\}$，為影像像素的編碼簿，總共有$L$個編碼，目前為空。
2. 針對N個影像像素，依序對每個像素抓取 RGB 值和亮度。並且作下列判斷：
   A. 在編碼簿$C=\{c_i|1\le i\le L\}$尋找符合的編碼c_m，並且滿足條件(a)和(b)

      (a) $colordist(x_t,v_m)\le\varepsilon_1$，$\varepsilon_1$:自訂門檻值

      (b) $brightness\left(I,\left(\check{I}_m,\hat{I}_m\right)\right)=true$

   B. 如果$C=\phi$或沒有任何編碼符合，則$L\leftarrow L+1$。並且建立新的編碼c_L。

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

      $aux_L\leftarrow\langle I,I,1,t-1,t,t\rangle$

C. 否則，則更新符合的編碼 c_m ，包含 $v_m = \left(\overline{R_m}, \overline{G_m}, \overline{B_m}\right)$ 和 $aux_m \leftarrow \left\langle \check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m \right\rangle$ ，更新如下：

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

$$aux_m \leftarrow \left\langle \min\left\{I, \check{I}_m\right\}, \max\left\{I, \hat{I}_m\right\}, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \right\rangle$$

3. 對每個編碼 $c_i, i = 1, \mathrm{K}, L$ ， $\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i - 1)\}$

2.2.3 色彩和亮度

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，但是在影像較暗的區域效果不甚理想，因為亮度會有比較高的不確定性，所以在偵測黑暗區域會有較多的錯誤。

觀察影像像素在亮度明暗上的時間變化，設計一種色彩模型來評估色彩差異和亮度差異如圖 1 所示，這個模型主要是讓背景像素位於主要軸的編碼沿著高低亮度邊界移動，因為亮度的變化來產生移動。所以當有一個像素 $x_t = (R, G, B)$ 和一個編碼 c_i ， $v_i = \left(\overline{R}_i, \overline{G}_i, \overline{B}_i\right)$ ，如式(7)：

$$\begin{aligned} \|x_t\|^2 &= R^2 + G^2 + B^2 \\ \|v_i\|^2 &= \overline{R_i}^2 + \overline{G_i}^2 + \overline{B_i}^2 \\ \langle x_t, v_i \rangle^2 &= \left(\overline{R_i} R + \overline{G_i} G + \overline{B_i} B\right)^2 \end{aligned} \qquad (7)$$

利用色彩差異δ計算如式（8），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。

$$p^2 = \|x_t\|^2 \cos^2\theta = \frac{\langle x_t, v_i \rangle^2}{\|v_i\|^2}$$

$$colordist(x_t, v_i) = \delta = \sqrt{\|x_t\|^2 - p^2} \qquad (8)$$

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(9)所示：

$$I_{low} = \alpha \hat{I} \quad , \quad I_{hi} = \min\left\{\beta \hat{I}, \frac{\check{I}}{\alpha}\right\} \qquad (9)$$

α: 0.4 ~ 0.7 ； β: 1.1 ~ 1.5

亮度函數定義如式(10)所示：

$$brightness\left(I, \left\langle \check{I}, \hat{I} \right\rangle\right) = \begin{cases} \text{true} & \text{if } I_{low} \le \|x_t\| \le I_{hi}, \\ \text{false} & \text{otherwise.} \end{cases} \qquad (10)$$

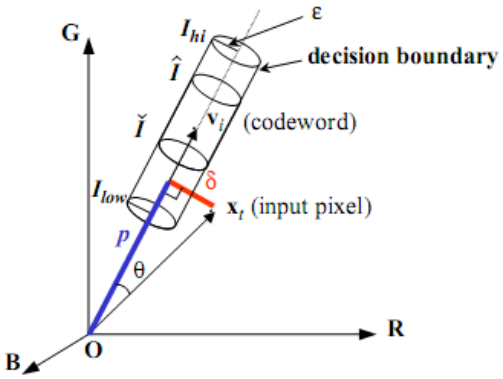


圖 1 Codebook 色彩模型

2.2.4 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：

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

II. 根據以下兩個條件來尋找符合的編碼c_m，並且按照編碼簿建立之演算法步驟2(iii)來更新符合的編碼。

- $colordist(x,c_m) \le \varepsilon_2$，$\varepsilon_2$:自訂門檻值
- $brightness\left(I,\left\langle \check{I}_m,\hat{I}_m \right\rangle\right) = \text{true}$

III. 如果目前影像像素經過編碼簿模型來搜尋配對的編碼成功，則我們認定該影像像素為背景像素，否則為移動物體像素。

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

$BGS(x)$：經過編碼簿背景模型偵測後的結果

2.3 以機率為基礎的背景擷取演算法（Probability-Based Algorithm）

本節要介紹 Chiu 等人的以機率為基礎的背景擷取演算法[9]，他們認為多數的背景影像是靜止的，背景像素在背景擷取過程中不應該變化，不過因為移動物體經過這些背景像素並造成亮度上的變化，因此在影像序列中每個像素將會有不同的色彩資訊來做為背景像素值的選擇，背景像素的色彩機率將會高為移動物體出現的機率。先前的研究[18],[20],[23]使用這種概念來擷取背景，不過它們的記憶體消耗大及處理時間久為主要缺點，而 Chiu 等人的方法則解決了這些問題。

2.3.1 背景模型建立

首先使用像素色彩差異值來分類輸入影像的每個像素直到我們設定的訓練張數F_n，並且計算每張影像像素的色彩機率，另外設一個收斂值讓擷取背景像素色彩最大機率需要大於這個收斂值。根據收斂值公式（15），每張影像的背景像素色彩能夠快速精確地被擷取。

假設 $f_i(x,y)$影像序列的第 i 張，x 和 y 為像素座標，影像大小為 $M*N$。$f_i(x,y)$的每個像素由 $R_i(x,y)$、$G_i(x,y)$和 $B_i(x,y)$組成。經過像素分類後，參數 $C(x,y,n_{xy})$為像素座標 (x,y)位置第 n_{xy} 個像素群組的個數，$R(x,y,n_{xy})$、$G(x,y,n_{xy})$和 $B(x,y,n_{xy})$分別為該位置第 n_{xy} 個像素群組的 *RGB* 資訊。參數 n_{xy} 為像素座標 (x,y)位置的像素群組數。詳細步驟如下：

步驟 1：擷取第一張輸入影像 $f_i(x,y),\ i=1$，設

$$\begin{cases} R(x,y,n_{xy})=R_1(x,y) \\ G(x,y,n_{xy})=G_1(x,y) \\ B(x,y,n_{xy})=B_1(x,y) \\ C(x,y,n_{xy})=1 \end{cases}, \qquad \text{where} \begin{cases} x=0\sim M-1 \\ y=0\sim N-1 \\ n_{xy}=1 \end{cases}$$

步驟 2：擷取下一張輸入影像 $f_i(x,y),\ i=i+1$，依序擷取像素 *RGB* 資訊

$f_i(x,y)=\{R_i(x,y),G_i(x,y),B_i(x,y)\}$，$x=0,1,2,\Lambda\ ,M-1$，

$y=0,1,2,\Lambda\ ,N-1$

步驟 3：如果輸入影像 $f_i(x,y)$內的像素都計算完畢則跳回步驟 2，否則依序計算目前影像像素與在像素座標位置 (x,y)像素群組內像素的色彩差異 $D_i(x,y,n_{xy})$如式(11)

$$\begin{aligned} D_i(x,y,n_{xy}) = &\left|R_i(x,y)-R(x,y,n_{xy})\right|+ \\ &\left|G_i(x,y)-G(x,y,n_{xy})\right|+ \\ &\left|B_i(x,y)-B(x,y,n_{xy})\right|,\forall n_{xy} \end{aligned} \qquad (11)$$

步驟 4：找出 $D_i(x,y,n_{xy})$的最小值 $D(x,y,m)$如式(12)

$$D_i(x,y,m)=\min_{\forall n_{xy}} D_i(x,y,n_{xy}),\quad \text{where}\ \ m\subset n_{xy} \qquad (12)$$

步驟 5：如果$D(x,y,m)\le TH_D$，則認定找到最佳像素。輸入像素會被分類到像素座標位置(x,y)像素群組第m組，更新第m組個數和RGB資訊，$C(x,y,m)=C(x,y,m)+1$，並且根據目前影像像素RGB資訊對背景模型像素群組資訊作更新動作。

TH_D:自訂門檻值

if $(R(x,y,m)<R_i(x,y))$ then

$R(x,y,m)=R(x,y,m)+1$

else $R(x,y,m)=R(x,y,m)-1$

if $(G(x,y,m)<G_i(x,y))$ then

$G(x,y,m)=G(x,y,m)+1$

else $G(x,y,m)=G(x,y,m)-1$

if $(B(x,y,m)<B_i(x,y))$ then

$B(x,y,m)=B(x,y,m)+1$

else $B(x,y,m)=B(x,y,m)-1$

否則，輸入像素就會被設為像素座標位置(x,y)新的背景像素群組，新的背景像素群組資訊如下：

$$n_{xy}=n_{xy}+1 \qquad \text{and} \qquad \begin{cases} R(x,y,n_{xy})=R_i(x,y) \\ G(x,y,n_{xy})=G_i(x,y) \\ B(x,y,n_{xy})=B_i(x,y) \\ C(x,y,n_{xy})=1 \end{cases}$$

跳回步驟 3

步驟 6：如果$i \geq F_n$，計算背景像素每個群組的最大機率$P_i^{\max}(x,y)$如式(13)、(14)所示，然後執行步驟 7。

$$P_i(x,y,n_{xy}) = \frac{C(x,y,n_{xy})}{i} \quad (13)$$

$$P_i^{\max}(x,y) = \max_{\forall n_{xy}} P_i(x,y,n_{xy}) \quad (14)$$

若$i < F_n$，則跳回步驟 3。

F_n:自訂的訓練張數

步驟 7：假如$P_i^{\max}(x,y) > TH_E$，像素座標位置(x,y)背景像素群組為最大機率被認為是背景像素。收斂值TH_E將會調整為式(15)所示：

$$TH\_E = \eta * \omega^{(i-F_n)}, \qquad i > F_n \quad (15)$$

η:TH_E的初始收斂值；ω:初始收斂的加權值

步驟 8：若全部的背景像素已被擷取完成則停止計算。否則，回到步驟 3。

2.3.2 色彩分割與背景更新

背景影像經過上述方法建立完成後，移動物體就能由背景影像和輸入影像*RGB*差異值計算來偵測。為了得到最佳效能，需要克服背景

及亮度的變化，而門檻值也必須能夠自動調整，因此使用了色彩分割方法，利用色彩差異直方圖來計算門檻值。

假設 $f_B(x,y)$ 為背景影像，影像大小為 $M*N$，x、y 分別表示為影像像素座標。$R_B(x,y)$、$G_B(x,y)$ 和 $B_B(x,y)$ 為 $f_B(x,y)$ 的 *RGB* 資訊。

色彩分割步驟：

步驟 1：擷取輸入影像 $f_i(x,y)$，i 為輸入影像的索引。$R_i(x,y)$、$G_i(x,y)$ 和 $B_i(x,y)$ 為 $f_i(x,y)$ 的 *RGB* 資訊。

步驟 2：計算色彩差異直方圖 H_{i_R}、H_{i_G} 和 H_{i_B}。H_{i_R}、H_{i_G} 和 H_{i_B} 分別定義為 $\Delta R(x,y)=R_i(x,y)-R_B(x,y)$、$\Delta G(x,y)=G_i(x,y)-G_B(x,y)$ 和 $\Delta B(x,y)=B_i(x,y)-B_B(x,y)$ 的直方圖，$x=0,1,2,\Lambda,M-1$ 和 $y=0,1,2,\Lambda,N-1$。為了減少雜訊，再計算色彩差異直方圖後，H_{i_R}、H_{i_G} 和 H_{i_B} 會使用式（16）來讓直方圖變得平滑。而平滑過後的直方圖被定義為 H_{i_sR}、H_{i_sG} 和 H_{i_sB}。

$$H_{i\_sR}[j]=\frac{\sum_{k=-2}^{2}H_{i\_R}[j+k]}{5}$$

$$H_{i\_sG}[j]=\frac{\sum_{k=-2}^{2}H_{i\_G}[j+k]}{5}$$

$$H_{i\_sB}[j]=\frac{\sum_{k=-2}^{2}H_{i\_B}[j+k]}{5} \qquad (16)$$

$$where \quad j=-253,\Lambda,-1,0,1,\Lambda,253.$$

步驟 3：從第 i 張輸入影像的色彩差異直方圖 H_{i_sR} 、 H_{i_sG} 和 H_{i_sB} 來計算門檻值 SH_i 和 SL_i 。在每張色彩差異直方圖中，背景像素分佈類似於高斯函數的零均值（zero mean），移動物體像素分佈將位於零均值的兩端。因此第一個轉折點會設為高斯函數零均值的兩端作為高轉折點 (HR_i, HG_i, HB_i) 或低轉折點 (LR_i, LG_i, LB_i)，由式（17）、（18）來定義。然後高門檻值 SH_i 設為 $SH_i = HR_i + HG_i + HB_i$ ；低門檻值 SL_i 設為 $SL_i = LR_i + LG_i + LB_i$ 。圖 2 為 Waving_Tree 影像序列第 250 張影像與背景影像相減後的直方圖，利用第一個高低轉折點來找出移動物體。

$$\text{if } \{H_{i\_sR}[k-1] \geq H_{i\_sR}[k] \text{ and } H_{i\_sR}[k] \leq H_{i\_sR}[k+1]\},$$
$$HR_i = k$$
$$\text{if } \{H_{i\_sG}[k-1] \geq H_{i\_sG}[k] \text{ and } H_{i\_sG}[k] \leq H_{i\_sG}[k+1]\},$$
$$HG_i = k$$
$$\text{if } \{H_{i\_sB}[k-1] \geq H_{i\_sB}[k] \text{ and } H_{i\_sB}[k] \leq H_{i\_sB}[k+1]\},$$
$$HB_i = k$$
$$k = 1,2,\Lambda\ ,252 \qquad (17)$$
$$\text{if } \{H_{i\_sR}[k'-1] \geq H_{i\_sR}[k'] \text{ and } H_{i\_sR}[k'] \leq H_{i\_sR}[k'+1]\},$$
$$LR_i = k'$$
$$\text{if } \{H_{i\_sG}[k'-1] \geq H_{i\_sG}[k'] \text{ and } H_{i\_sG}[k'] \leq H_{i\_sG}[k'+1]\},$$
$$LG_i = k'$$

if $\{H_{i\_sB}[k'-1] \geq H_{i\_sB}[k']$ and $H_{i\_sB}[k'] \leq H_{i\_sB}[k'+1]\}$,

$LB_i = k'$

$k' = -1, -2, \Lambda, -252$ （18）

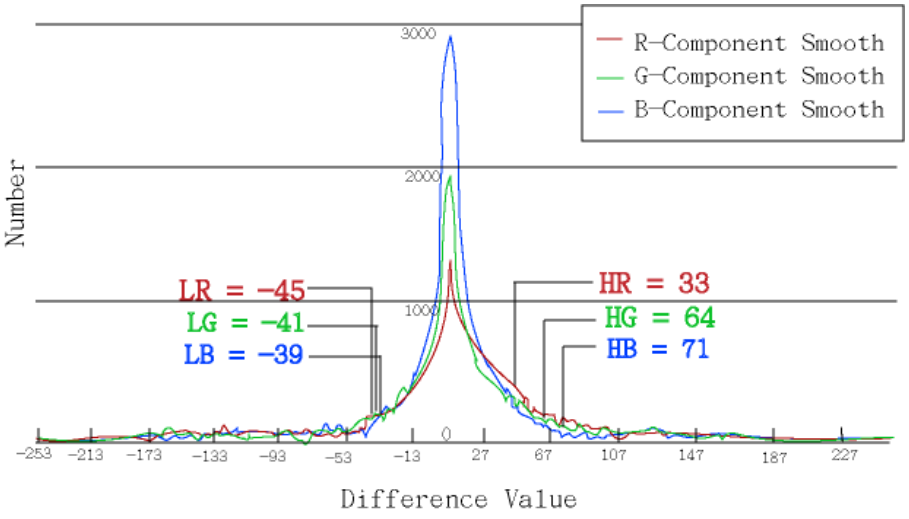


圖 2　Waving_Tree 影像序列高低轉折點色彩差異直方圖

移動物體和背景影像會根據式（18）來分割。參數 $D_{FB}(x,y)$ 為輸入影像和背景影像 RGB 資訊差異值總和，參數 SH_{i-1} 和 SL_{i-1} 為色彩差異直方圖計算用的門檻值，H_{i-1_sR}、H_{i-1_sG} 和 H_{i-1_sB} 從第 $i-1$ 張影像來計算。

$D_{FB}(x,y) = |\Delta R(x,y)| + |\Delta G(x,y)| + |\Delta B(x,y)|$

if $(D_{FB}(x,y) \leq SH_{i-1}$ or $D_{FB}(x,y) \leq |SL_{i-1}|)$

the pixel belongs to the static image

$Q_s(x,y) = (R_i(x,y), G_i(x,y), B_i(x,y))$

else

the pixel belongs to a intrusive object image

$$Q_o(x,y)=(R_i(x,y),G_i(x,y),B_i(x,y)) \quad (19)$$

步驟 4：將上個步驟分割出來的靜態影像與背景影像作更新如式(20)。參數$R_S(x,y)$、$G_S(x,y)$和$B_S(x,y)$分別定義為$Q_S(x,y)$的*RGB*資訊。

$$\begin{cases} R_B(x,y)=\dfrac{R_B(x,y)*(2^n-1)}{2^n}+\dfrac{R_s(x,y)}{2^n} \\ G_B(x,y)=\dfrac{G_B(x,y)*(2^n-1)}{2^n}+\dfrac{G_s(x,y)}{2^n} \\ B_B(x,y)=\dfrac{B_B(x,y)*(2^n-1)}{2^n}+\dfrac{B_s(x,y)}{2^n} \end{cases}$$

$$x=0\sim M-1,\ y=0\sim N-1 \quad (20)$$

n:自訂門檻值，設為 3

第三章　在複雜背景下使用主體色彩演算法之移動物體切割系統

第二章提到的背景模型建構方法，其中介紹的傳統模型建構方法雖然運算量少，但是效果不是很好，對於背景變化及亮度變化就沒有辦法有效地處理，而編碼簿演算法雖然能夠有效地處理背景變化和亮度變化，但是運算量大，所需要的時間較長，而以機率為基礎的背景擷取演算法雖然所需時間短，但是對於背景變化適應效果不是很好。因此我們在本章節提出了在複雜背景下使用主體色彩演算法之移動物體切割系統，之前的方法大多都是以像素為基礎來處理，本文則將影像分割為區域方式計算，減少計算量及處理時間。為了要解決動態背景的問題，並且提出了區域紋理資訊來處理背景及亮度的變化。

3.1 主體色彩背景模型

一般演算法在建構背景模型都是針對影像像素來擷取特徵點資訊，但是本篇論文主要是採用區域方式來計算該區域的主體色彩資訊，選擇區域方式是因為我們認為當影像像素因為背景變化而移動時，其實只是在該像素點周圍附近移動而已，因此我們如果選擇區域方式來擷取特徵點，可以有效地處理動態背景的像素變化。我們從圖

3 搖晃的樹木影像，固定一個區域來觀察可以發現，區域內的樹枝因為時間點的不同而晃動，但是並沒有離開過該區域，可以發現有些背景變化其實只在附近區域移動而已。因此採用區域方式來解決動態背景的問題，我們把影像分割成 $N*N$ 的區域依序來計算主體色彩，我們在第四章會針對不同的區域大小進行實驗，研究區域大小會對建構背景模型的效能有何影響。

(a)

(b)

圖 3　搖晃的樹木固定某個區域在時間點不同下的影像

(a)Frame#001(b)Frame#005

將測試影像分割成 k 個 $N*N$ 區域如圖 4 所示，並且針對每個區域內的像素互相做同質性計算，以便得到建構背景模型所需的主體色彩資訊，並且建立動態背景模型，儲存這些區域主體色彩資訊，等到背景模型完全建立成功後，即可作為移動物體偵測之用。

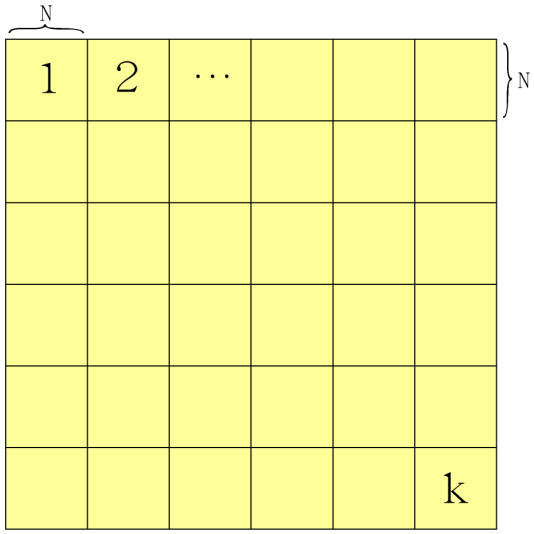


圖 4 目標影像分割為 k 個 $N*N$ 區域

主體色彩背景模型建立步驟：

I. 設定每個區域的左上角像素點作為初始位置，由左往右、由上往下依序計算，針對前一個像素點，分別對 R、G、B 三個值相減後取絕對值，再將三個差值作 norm 計算。

II. 將計算過後的值與設定的門檻值相減過後，如果小於門檻值，則認定這兩個像素有同質性，將符合個數 *matchcount* 累加一，若大於門檻值，則認定這兩個像素沒有同質性，新增該像素為主體色彩元素。

Major Color Background Model Construction Algorithm

I. $L \leftarrow 0^1, C \leftarrow \phi$ // L: 主體色彩像素個數; C: 主體色彩像素

II. 對區域內 m 個像素計算區域主體色彩資訊

for $t=1$ to m do

if $t=1$

$C_1 \leftarrow (R_1, G_1, B_1)$ ， $L \leftarrow 1$ ， $matchcount_{C_1} \leftarrow 1$

// $matchcount$: 符合個數

else

for $i=1$ to L do

$P_t = (R, G, B)$ ， $P_{sub} = \left\| p_t - C_i \right\|$

// P_t : 目前區域像素

$P_{norm} = [P_{subR}, P_{subG}, P_{subB}]$

if $P_{norm} < th_1$ // th_1 ： 自訂門檻值

$matchcount_{C_i} = matchcount_{C_i} + 1$

else

$L \leftarrow L+1$ .Create new element C_L

$C_L \leftarrow (R, G, B)$ ， $matchcount_{C_L} \leftarrow 1$

end

end

end

end

3.2 主體色彩背景模型訓練及更新

在上一節針對每個影像區域作同質性計算來得到每個影像區域的主體色彩資訊，進而建立背景模型初始化。本節是利用剛建立好的背景模型，依照連續影像序列的長度來選擇訓練的張數，在訓練過程中來記錄動態背景及亮度上的變化，並且更新背景模型。

為了處理動態背景的變化，即時的背景更新也是必須的。背景模型在訓練過程中，會將目前影像背景變化記錄更新起來，變化的速度取決更新頻率的設定，更新速度的過快或過慢也可能導致錯誤偵測。

主體色彩背景模型訓練及更新步驟：

1. 將目前影像的每個區域由左往右、由上往下依序與先前建立的背景模型區域互相比對主體色彩資訊。將目前影像區域像素 R、G、B 值分別與背景模型區域主體色彩元素 R、G、B 值相減並取絕對值，再將三個差值作 norm 計算。
2. 將計算過後的值與設定的門檻值相減過後，若小於門檻值，則認定可能為背景區域，因此累計符合個數 $matchcount_k$，並且更新背景模型主體色彩資訊如式（21）：

$$C_i = \alpha P_t + (1-\alpha) C_i \qquad (21)$$

α:學習速率

C_i:背景模型區域主體色彩資訊

P_i:目前影像區域主體色彩資訊

Major Color Background Model Training Algorithm

for i = 1 to L do

$P_t = (R,G,B)$ ， $P_{sub} = \|p_t - C_i\|$

$P_{norm} = [P_{subR}, P_{subG}, P_{subB}]$

if $P_{norm} < th_2$ //th_2:自訂門檻值

$matchcount_{C_i} = matchcount_{C_i} + 1$

//對符合的區域作背景更新

$C_i = \alpha P_t + (1-\alpha)C_i$

//α：更新速率

end

end

圖 5 是在 PETS 資料庫(ftp://pets.rdg.ac.uk)中挑選的 Waving_Tree 影片和 Parking_Area 影片，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模型演算法來測試，而這個影片大小為 160*120 pixels，15 frames/sec。Parking_Area 影片內

容在校園停車場，攝影機中央有一棵樹會隨風搖蕩，場景會因為陽光有明暗的變化，相當適合檢測動態背景和亮度的變化。影片大小為480*360 pixel。圖 5（a）是本文演算法對 Waving_Tree 影片開始建立背景模型的初始背景，之後經過訓練過程，會將動態背景的變化記錄更新起來，以便處理之後的動態背景變化。圖 5（b）是本文演算法對 Waving_Tree 影片經過訓練過程後的背景模型，可以發現原本靜止不動的樹，把樹枝搖晃的動作給記錄更新起來了。圖 5（c）是本文演算法對 Parking_Area 影片開始建立的背景模型的初始背景，圖 5(d）是本文演算法對 Parking_Area 影片經過訓練過程後的背景模型，主要是記錄亮度的變化。

（a）

（b）

（c） （d）

圖 5 本文演算法對測試影片建立的背景模型中初始背景和訓練過的背景(a) Waving_Tree 的初始背景(b) Waving_Tree 訓練後的背景(c) Parking_Area 的初始背景(d) Parking_Area 訓練後的背景

3.3 物體偵測

經由上述的背景模型建立及訓練過程後，記錄了動態背景的變化，包括色彩及亮度的變化。由於動態背景模型已經建立完成，因此對於移動物體偵測上不須花費大量時間。由於本文採用區域方式來計算資訊，因此會有 block effect 的問題，本節會介紹解決的方法。

物體偵測步驟：

1. 將目前影像的每個區域與先前建立的背景模型區域互相比對主體色彩資訊。目前影像區域像素的 R、G、B 值與背景模型區域編碼 R、G、B 值相減並取絕對值，再將三個差值作 norm 計算。

2. 將計算過後的值與設定的門檻值相減過後，若小於門檻值，則認定可能為背景區域，並且累計符合個數 $matchcount_k$。如果符合個數累計大於設定的門檻值，則定義該目前影像區域為 $imgBLK_k$，並且給值為零，標記為背景區域，否則給值為一，標記為前景區域。

if $matchcount_k < th_3$

$imgBLK_k = 1$ //表示目前影像區域為前景

else

$imgBLK_k = 0$ //表示目前影像區域為背景

end

因為本篇論文是以區域方式來計算，因此會有 Block Effect 的問題，因此要處理這種情況。每張影像總共有 k 個 $N*N$ 大小區域，不考慮影像周圍最外圈的區域。當該區域九宮格範圍內的區域屬於前景區域，則會將 BLK_{Count} 累加一。若 BLK_{Count} 大於設定的門檻值，則標記該區域九宮格範圍內的區域皆為前景，否則只標記該區域為背景。

BLK_{Count} :計算 $imgBLK_k$ 九宮格範圍內的區域屬於前景的數量

if $imgBLK_k == 1$

if $imgBLK_{k-N-1} == 1$

$BLK_{Count} = BLK_{Count} + 1$

if $imgBLK_{k-N} == 1$

$BLK_{Count} = BLK_{Count} + 1$

if $imgBLK_{k-N+1} == 1$

$BLK_{Count} = BLK_{Count} + 1$

if $imgBLK_{k-1} == 1$

$BLK_{Count} = BLK_{Count} + 1$

if $imgBLK_{k+1} == 1$

$BLK_{Count} = BLK_{Count} + 1$

if $imgBLK_{k+N-1} == 1$

$BLK_{Count} = BLK_{Count} + 1$

if $imgBLK_{k+N} == 1$

$BLK_{Count} = BLK_{Count} + 1$

if $imgBLK_{k+N+1} == 1$

$BLK_{Count} = BLK_{Count} + 1$

end

// $NewBLK_k$: 更新後的 $imgBLK_k$

if $BLK_{Count} \geq th_3$ // th_3 : 自訂門檻值

$NewBLK_k = 1$ //前景

$NewBLK_{k-N-1}=1$

$NewBLK_{k-N}=1$

$NewBLK_{k-N+1}=1$

$NewBLK_{k-1}=1$

$NewBLK_{k+1}=1$

$NewBLK_{k+N-1}=1$

$NewBLK_{k+N}=1$

$NewBLK_{k+N+1}=1$

else

$NewBLK_k=0$ //背景

end

3. 將經過步驟 2 得到的新的背景區域，執行背景更新，如式（21）

if $NewBLK_k==0$ & $matchcount_k \geq th_5$

$C_i=\alpha P_t+(1-\alpha)C_i$

end

4. 不考慮影像最外圍區域，只針對每個區域周圍九宮格範圍區域來判定，如果周圍八個區域皆被標記為背景區域，才標記該區域為背景區域，否則為標記為前景區域，如式（22）所示：

$$NewBLK_k=\left(NewBLK_{k-N-1}\middle|NewBLK_{k-N}\middle|NewBLK_{k-N+1}\middle|NewBLK_{k-1}\middle|NewBLK_{k+1}\middle|NewBLK_{k+N-1}\middle|NewBLK_{k+N+1}\right) \quad (22)$$

5. 針對處理過後標記為前景的影像區域，所以可能會遇到這個區域被標記為前景，但是區域內有幾個像素為背景像素，而使得結果出現誤差。因此還會對標記前景區域內的像素，再與背景模型區域的主體色彩資訊計算，將區域內的像素與背景模型區域編碼相減作 norm 計算，若小於設定的門檻值，則認定為背景像素。若該區域像素都沒有小於設定的門檻值，則認定為前景像素。

if $NewBLK_k = 1$ // $NewPiexl$: 為新區域 $NewBLK_k$ 內的像素

for $t = 1$ to m do

for $i = 1$ to L do

$P_t = (R, G, B)$ ， $P_{sub} = \left\| p_t - C_i \right\|$

$P_{norm} = \left[P_{subR}, P_{subG}, P_{subB} \right]$

if $P_{norm} < th_1$

$NewPiexl = 0$ //更新為背景像素

else

$NewPiexl = 1$ //更新為前景像素

end

end

end

end

6. 完成步驟 5 之後得到的前景與背景像素則是移動物體偵測後的結果。

3.4 區域紋理資訊

在實驗過程中發現雖然本文提出的方法可以有效地偵測移動物體，但是在移動物體周圍輪廓偵測會有少許的錯誤，圖 5 是本文提出的方法在影像序列中移動物體偵測的結果。圖 6（a）為原始影像序列第 247 張影像，圖 6（b）是圖 5（a）的 ground truth，而圖 6（c）則是本文提出的演算法所得到的結果，可以發現還是有一小部份動態背景被誤認為前景偵測出來。在第四章會針對這個來進行討論，因此在本節提出了區域紋理資訊，利用直角座標系概念來分析區域紋理特性，加入到動態背景模型，分析移動物體及背景區域紋理，使得偵測移動物體輪廓錯誤率下降，並且能夠解決動態背景變化的問題。

(a) (b) (c)

圖 6 本文演算法對 Waving_Tree 影片執行後的移動物體偵測結果(a)original image (b)ground truth (c) proposed

利用直角座標系的概念將區域內的像素分別區分成四個象限區域，並且分別將各個象限內的像素亮度值累加起來，另外設一個中間

區域分別對四個象限計算，將計算過後的值用一個 4Bit 值來表示該區域的紋理資訊。

區域紋理資訊分析步驟：

1. 將區域內每個像素取亮度值，$I \leftarrow \sqrt{R^2 + G^2 + B^2}$
2. 利用直角座標系概念，將區域分為四個象限，並且另設一個中間區域。分別將屬於四個象限和中間區域的像素亮度值累加，如式（23）。以圖 7 為例，4*4 像素區域劃分為四個象限，每個象限共有 4 個像素，第一象限 $Zone_1$：編號第 1、2、5、6 個像素；第二象限 $Zone_2$：編號第 3、4、7、8 個像素；第三象限 $Zone_3$：編號第 9、10、13、14 個像素；第四象限 $Zone_4$：編號第 11、12、15、16 個像素；中間區域 $MiddleZone$：編號第 6、7、10、11 個像素。

$$Zone_1 = I_3 + I_4 + I_7 + I_8 \text{ ，}$$

$$Zone_2 = I_1 + I_2 + I_5 + I_6 \text{ ，}$$

$$Zone_3 = I_9 + I_{10} + I_{13} + I_{14} \text{ ，}$$

$$Zone_4 = I_{11} + I_{12} + I_{15} + I_{16} \text{ ，}$$

$$MiddleZone = I_6 + I_7 + I_{10} + I_{11} \text{ 。} \quad (23)$$

$Zone_{1\sim4}$: 將目前影像 $N*N$ 區域分為四個象限區域

$MiddleZone$: 為目前影像 $N*N$ 區域中的中間區域

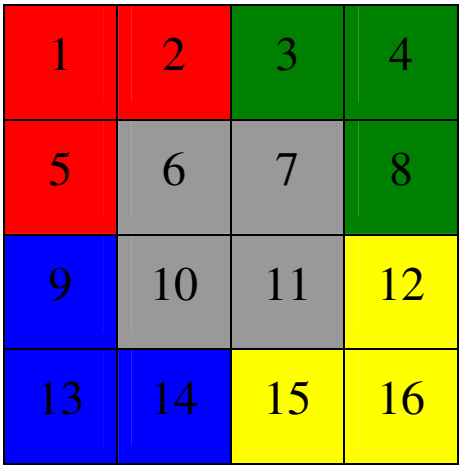


圖 7 以 4*4 像素區域為例，用直角座標系劃分影像區域。

3. 將中間區域像素亮度值總和與其他四個象限區域像素亮度值總和相減後，用一個 4Bit 值來表示該區域的紋理資訊。

for i = 1 to 4 do

$Zone_{Sub} = MiddleZone - Zone_i$

if $Zone_{Sub} \geq 0$

$Bit[i] = 1$

else

$Bit[i] = 0$

end

end

3.4.1 含區域紋理之背景模型

本節提出了區域紋理資訊的概念來幫忙處理動態背景的問題，在開始使用以色彩為基礎之複雜背景物體分割演算法建立動態背景模型的時候，加入區域紋理資訊到背景模型中。

含區域紋理之背景模型建立步驟：

1. 每個區域的左上角像素作為初始位置，由左往右，由上往下依序計算，對前一個像素分別對 R、G、B 三個值互減後取絕對值後，將三個值作 norm 計算。
2. 將計算過後的值與設定的門檻值作比較，如果小於門檻值，則認定這兩個像素有同質性，將符合個數累加一，若大於門檻值，則認定這兩個像素沒有同質性，新增該像素到主體色彩元素內。
3. 將區域內的每個像素取亮度值，使用直角座標系象限概念，另外設一個中間區域，將區域內分別屬於四個象限和中間區域的像素亮度值累加，將中間區域像素亮度值總和與其他四個象限區域像素亮度值總和相減後，用一個 4Bit 值來表示該區域的紋理資訊，將這個 4Bit 值加入到影像區域主體色彩中，作為建立背景模型所需。

3.4.2 含區域紋理之背景模型訓練及更新

本節是介紹將區域紋理資訊加入到動態背景模型訓練及更新的過程，背景模型主體色彩資訊增加了區域紋理資訊後，訓練過程中可以針對動態背景及亮度變化上更為精確地記錄，增加偵測移動物體上的準確率及降低錯誤的誤判。

含區域紋理之背景模型訓練及更新步驟：

1. 將目前影像的每個區域由左往右、由上往下依序與先前建立的背景模型區域互相比對主體色彩資訊。將目前影像區域像素 R、G、B 值分別與背景模型區域主體色彩元素的 R、G、B 值相減並取絕對值，再將三個差值作 norm 計算。
2. 將目前影像區域內的每個像素取亮度值，使用直角座標系象限概念，另外設一個中間區域，將區域內分別屬於四個象限和中間區域的像素亮度值累加，將中間區域像素亮度值總和分別與其他四個象限區域像素亮度值總和相減後，用一個 4Bits 值來表示該目前影像區域的紋理資訊。

3. 步驟 1 計算過後的值與設定的門檻值相減過後，若小於門檻值，則認定可能為背景區域，因此累計符合個數 $matchcount_k$，並且更新背景模型主體色彩資訊。
4. 將目前影像區域計算後得到的 4Bits 值與背景模型區域主體色彩資訊的 4bits 值比對，如果完全相同則累計次數，否則則新增該目前影像區域的 4Bits 值，更新到背景模型主體色彩資訊中。

3.4.3 物體偵測

經由新增的區域紋理資訊加入到動態背景模型訓練及更新後，完整的背景模型已經建立成功。新的動態背景模型，加上了區域紋理資訊，可以有效地記錄動態背景及亮度上的變化，對於複雜背景下的移動物體偵測有相當好的效果，但是於本文採用區域方式來計算資訊，因此會有 Block Effect 的問題須解決，本節會描述如何處理該問題。

物體偵測步驟：

1. 將目前影像的每個區域與先前建立的背景模型區域互相比對主體色彩資訊。將目前影像區域像素的 R、G、B 值與背景模型區域主體色彩元素 R、G、B 值相減並取絕對值，再將三個差值作 norm 計算。

2. 將目前影像區域內的每個像素取亮度值，使用直角座標系象限概念，另外設一個中間區域，將區域內分別屬於四個象限和中間區域的像素亮度值累加，將中間區域像素亮度值總和分別與其他四個象限區域像素亮度值總和相減後，用一個 4Bit 值來表示該目前影像區域的紋理資訊。

3. 將步驟 1 計算過後的值與設定的門檻值相減過後，若小於門檻值，則認定可能為背景區域，並且累計符合個數 $matchcount_k$。假如相似次數小於設定的門檻值，並且沒有與背景模型區域主體色彩資訊的 4Bits 值完全相同，則認定該目前影像區域為前景，否則為背景。

if $matchcount_k < th_3 \,\&\, Zone_{4Bits} == 0$

$imgBLK_k = 1$ //表示目前影像區域為前景

else

$imgBLK_k = 0$ //表示目前影像區域為背景

th_3 : 自訂門檻值

$Zone_{4Bits}$: 目前影像區域紋理資訊與背景模型區域紋理資訊相同累加次數總和

4. 為了處理 Block Effect 問題，每張影像總共有 k 個區域，不考慮影像周圍最外圈的區域。當該區域九宮格範圍內的區域屬於前景

區域的數量大於設定的門檻值，則標記該區域九宮格範圍內的區域皆為前景，否則只標記該區域為背景。

5. 針對每個目前影像區域周圍九宮格範圍區域來判定，如果周圍九個區域皆被標記為背景區域，才標記該區域為背景區域，否則為標記為前景區域，如式（22）所示。

6. 針對處理過後標記為前景的影像區域，所以可能會遇到這個區域被標記為前景，但是區域內有幾個像素為背景像素，而使得結果出現誤差。因此還會對標記過區域內的像素，再與背景模型區域的主體色彩門檻值計算，將區域內的像素與背景模型區域主體色彩元素相減作 norm 計算，若小於設定的門檻值，則認定為背景像素。若該區域像素都沒有小於設定的門檻值，則認定為前景像素。

7. 完成上述步驟後所得到的前景像素與背景像素則是移動物體偵測後的結果。

第四章　實驗結果

本章首先針對選擇區域大小來探討建構背景模型的效能差異，接著對本篇論文所提出的在複雜背景下使用主體色彩演算法之移動物體切割系統、在複雜背景下使用主體色彩含區域紋理資訊演算法之移動物體切割系統與編碼簿演算法和 Chiu & Ku 提出的以機率為基礎的背景擷取演算法對 PETS 資料庫中的 Waving_Tree 影像序列和 Parking_Area 影像序列作效能比較。

4.1 實驗環境與測試影片

本研究之實驗環境使用 Matlab 7.0 於 Windows XP 作業系統來開發程式。硬體設備為 Intel Core2 Q9550 2.83GHz 及 4G 記憶體。測試影片一共有兩組序列，皆是從 PETS 資料庫擷取(ftp://pets.rdg.ac.uk)。第一段影片是 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 160*120 pixel，15 frames/sec，影像序列為 286 張影像，BMP 格式。第二段影片是 Parking_Area，影片內容在校園停車場，攝影機中央有一棵樹會隨風搖蕩，場景會因為陽光有明暗的變化，相當適合檢測動

態背景和亮度的變化。影片大小為 480*360 pixel，影像序列為 2823 張影像，BMP 格式。

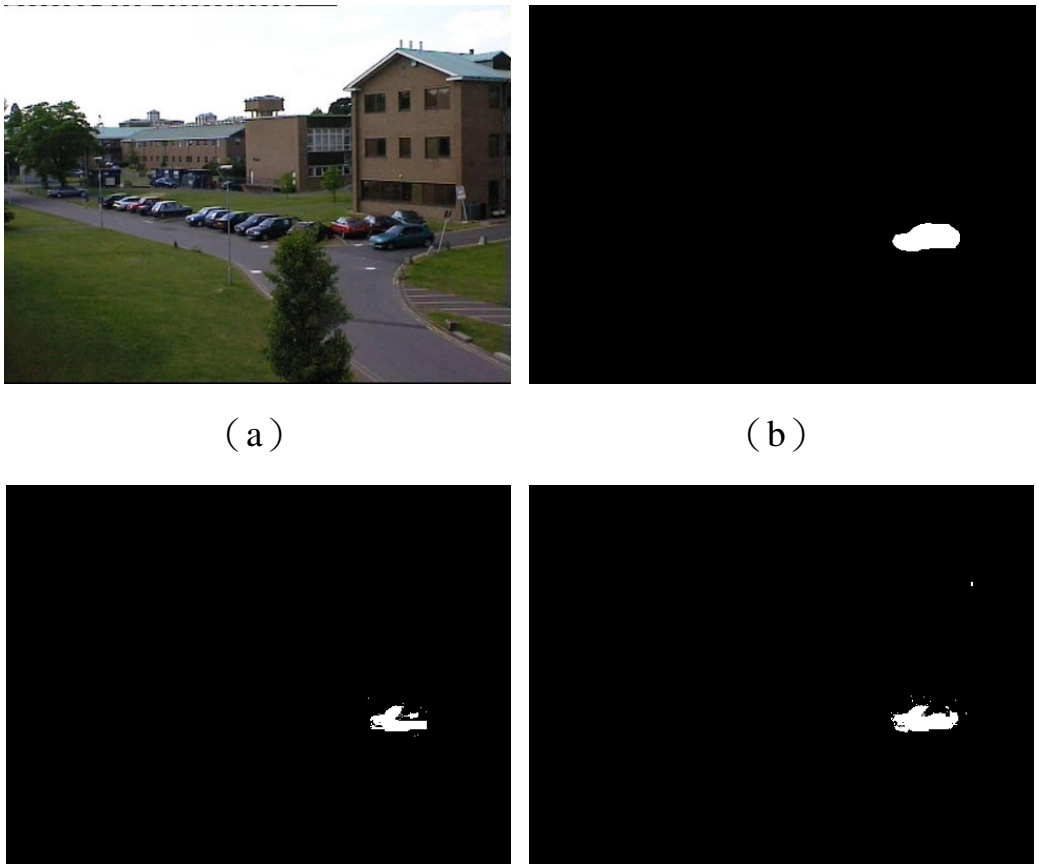
4.2 選擇區域大小

首先針對區域大小進行實驗，對於選擇不同的區域大小會對偵測移動物體的效果有何影響。圖 8 是說明主體色彩演算法對 Waving_Tree 影片執行過後的結果，選擇 frame249 作為結果顯示。圖 8(b)為理想結果，圖 8(c)是選擇 8*8 區塊所執行後的結果，圖 8(d)是選擇 4*4 區塊所執行後的結果。從圖 8 發現選擇 8*8 區塊所執行完的結果，相較於 4*4 區塊的結果，在偵測移動物體的輪廓邊緣不是很理想。而圖 9 是本文提出的演算法對 Parking_Area 影片執行後的結果，選擇 frame1290 作為結果顯示。圖 9(b)為理想結果，圖 9(c)是選擇 8*8 區塊所執行後的結果，圖 9(d)是選擇 4*4 區塊所執行後的結果。從圖 9 可以發現選擇 8*8 區塊跟選擇 4*4 區塊的結果相差不多，但是還是 4*4 區塊的效果比較好，在移動物體(車輛)的車頂都沒有偵測到，原因在於移動物體所在位置在原先背景也是車輛（黑色），顏色太過相近導致偵測錯誤。

（a）　　（b）　　（c）　　（d）

圖 8 主體色彩演算法對 Waving_Tree 影片選擇 4*4 區域、8*8 區域所執行的結果

(a)Frame#249(b)Ideal Result(c)8*8 Block Result(d)4*4 Block Result



（a）　　（b）

（c）　　（d）

圖 9 主體色彩演算法對 Parking_Area 影片選擇 4*4 區域、8*8 區域所執行的結果

(a)Frame#1290(b)Ideal Result(c)8*8 Block Result(d)4*4 Block Result

表 1 是主體色彩演算法對 Waving_Tree 影片和 Parking_Area 影片執行（訓練及偵測）所需時間及每秒可執行幀數，從表 1 來看發現兩個影片執行所需時間，選擇 8*8 區塊都是比較久的，而對測試影片每秒可執行幀數，則是選擇 4*4 區塊比較多，並且使用 ErrorRate 來檢驗演算法效能，如式（24）所示：

$$ErrorRate = \frac{FP + FN}{TotalPixels} \quad (24)$$

FN(*False Negative*):定義該像素實際為背景，結果卻誤認為前景。

FP(*False Positive*): 定義該像素實際為前景，結果卻誤認為背景。

圖 10 為主體色彩演算法對 Waving_Tree 影片選擇不同區域大小偵測移動物體的錯誤率，影像序列為第 242 張到第 259 張，剛好是人物出現在攝影機到離開的片段，圖 11 為本文提出的演算法對 Parking_Area 影片選擇不同區域大小偵測移動物體的錯誤率，影像序列為第 912 張到第 990 張。表 2 為主體色彩演算法對 Waving_Tree 影片和 Parking_Area 影片選擇不同區域大小偵測移動物體的平均錯誤率。從圖 10 發現 Waving_Tree 影像序列因為影像大小為 160*120，而移動物體與影像大小所佔比例很大，因此 4*4 區塊跟 8*8 區塊的錯誤率都比較高，介於 1~7%之間。而從圖 11 發現 Parking_Area 影像序列因為影像大小為 480*360，移動物體與影像大小所佔比例很小，

因此錯誤率會下降很多，介於 0.1%～0.6%之間。在 Waving_Tree 影片中選擇 4*4 區塊的錯誤率比較低，而在 Parking_Area 影片中選擇 8*8 區塊的錯誤率雖然比較低，但是也與選擇 4*4 區塊的錯誤率相差不多，最後看表 2 平均錯誤率發現選擇 4*4 區塊在 Waving_Tree 影片和 Parking_Area 影片的平均錯誤率為 3.4%與 0.38%，而選擇 8*8 區塊在 Waving_Tree 影片和 Parking_Area 影片的平均錯誤率為 5.12%與 0.34%，從平均錯誤率及執行結果顯示來判斷，因此選擇 4*4 區塊來作主體色彩演算法主要的測試。

表 1 主體色彩演算法選擇 4*4 區域、8*8 區域對測試影片(訓練及偵測)所需時間及每秒可執行幀數

主體色彩演算法選擇不同區域對測試影像序列(訓練及偵測)所需時間及每秒可執行幀數				
測試影像序列 / 區域大小	Waving_Tree		Parking_Area	
	所需時間	每秒可執行幀數	所需時間	每秒可執行幀數
4*4 BLK	192 s	1.49 fps	19066 s	0.148 fps
8*8 BLK	284 s	1.01 fps	19189 s	0.147 fps

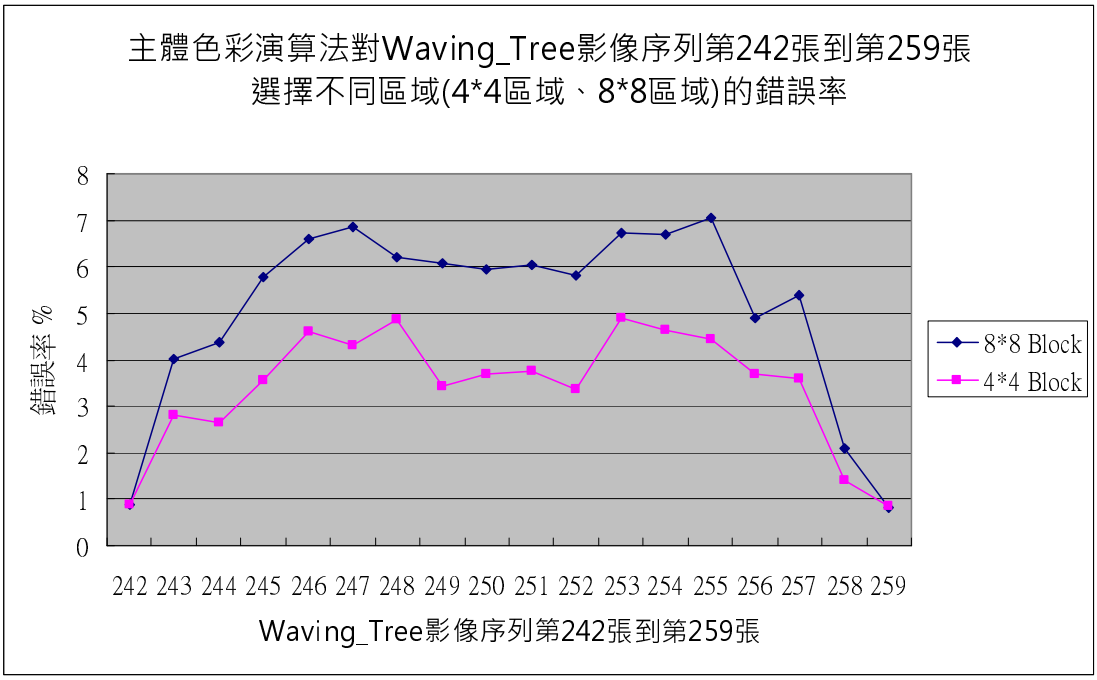


圖 10 主體色彩演算法對 Waving_Tree 影像序列（Frame#242~259）選擇 4*4 區域、8*8 區域的錯誤率

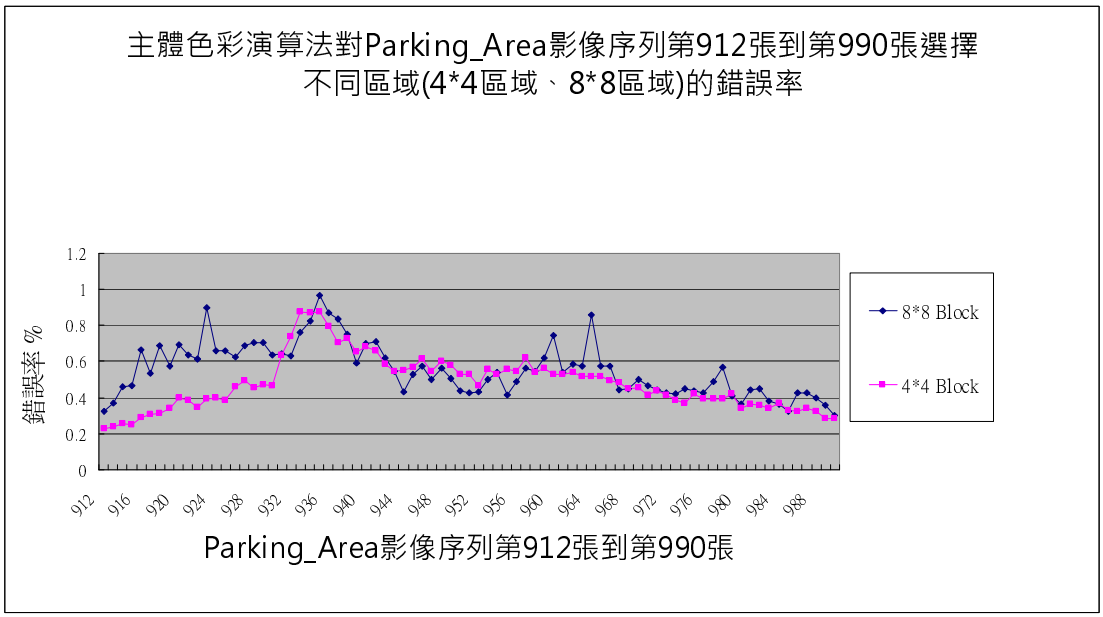


圖 11 主體色彩演算法對 Parking_Area 影像序列（Frame#912~990）選擇 4*4 區域、8*8 區域的錯誤率

表 2 主體色彩演算法對測試影像序列選擇 4*4 區域、8*8 區域的平均錯誤率

主體色彩演算法選擇不同區域對測試影像序列的平均錯誤率		
測試影像序列 / 區域大小	Waving_Tree（Frame#242~259）	Parking_Area（Frame#912~990）
	平均錯誤率	平均錯誤率
4*4 BLK	3.41%	0.38%
8*8 BLK	5.12%	0.54%

4.3 效能比較

在 3.4 節區域紋理資訊中提到主體色彩演算法對於測試影像序列執行後的結果，發現移動物體的輪廓和背景變化都有少許的誤認，因此我們又提出了區域紋理資訊來幫助背景模型的建立，有效地處理背景的變化。使用主體色彩演算法與編碼簿演算法和以機率為基礎的背景擷取演算法作效能比較，並且加入後來提出的區域紋理資訊作為比較，分析加入過後的差異。圖 12 是說明 Codebook 演算法、Probability-based 演算法、主體色彩演算法和主體色彩演算法加上區域紋理資訊對 Waving_Tree 影像序列執行移動物體偵測的結果，圖片依序為 Waving_Tree 影像序列第 246 張、理想結果、實際偵測結果。圖 12（a）為 Codebook 演算法對測試影像序列偵測的結果，可以發現該演算法對於動態背景的變化可以有效地處理，但是還是有少許的錯誤及雜訊，不過該演算法計算複雜度高，所需時間長。圖 12（b）為 Probability-based 演算法對測試影像序列偵測的結果，可以發現背

景的樹葉搖晃被錯誤偵測為移動物體了，因此可以了解該演算法對於動態背景的變化效果不是很好。圖 12（c）為主體色彩演算法，但未加入區域紋理資訊對測試影像序列偵測的結果，可以發現在移動物體邊緣輪廓附近有少許錯誤偵測。圖 12（d）為主體色彩演算法再加上區域紋理資訊對測試影像序列的移動物體偵測結果，可以發現加入了區域紋理資訊後，對於移動物體輪廓及邊緣的錯誤偵測降低了，因此得知區域紋理資訊可以有效地分析該區域的紋理資訊，藉由區域紋理資訊對動態背景及亮度的變化來做有效地處理。圖 13 為四種演算法對 Parking_Area 影像序列偵測移動物體的結果，圖片依序為 Parking_Area 影像序列第 258 張、理想結果、實際偵測結果。圖 13(a)為 Codebook 演算法對 Parking_Area 影像序列的移動物體偵測結果，發現該演算法偵測結果也是與理想結果相差不大，但是在移動物體本體周圍輪廓沒有正確偵測到。圖 13(b)為 Probability-based 演算法對 Parking_Area 影像序列的移動物體偵測結果，可以發現該演算法把背景的亮度變化和動態背景變化都誤認為前景了。圖 13(c)為主體色彩演算法但無加入區域紋理資訊對 Parking_Area 影像序列偵測的結果，可以發現實際結果與理想結果相差無幾，只有移動物體輪廓附近有少許雜訊。圖 13(d)為主體色彩演算法加上區域紋理資訊對

Parking_Area 影像序列偵測的結果，可以發現結果相較於其他三種演算法，偵測移動物體的效果最佳，最接近於理想結果。

表 3 是說明 Codebook 演算法、Probability-based 演算法、主體色彩演算法和主體色彩演算法加上區域紋理資訊分別對 Waving_Tree 影片和 Parking_Area 影片（訓練及偵測）移動物體的結果所需時間及每秒可執行幀數。從表 3 顯示主體色彩演算法在 Waving_Tree 影片執行所需時間最短，所以後來加入了區域紋理資訊後，也不須花費太多時間計算，而 Codebook 演算法因為計算複雜而須花費很多時間，因此最久。而在 Parking_Area 測試影片因為此影片長度及大小都相較於 Waving_Tree 影片來得大，因此可以用來評估演算法的效能，發現也是主體色彩演算法所需時間最短，Codebook 演算法最長。使用式（26）的 Error Rate 來對這四種演算法作效能評估，利用錯誤率大小來檢驗演算法的效能。圖 14 為四種演算法在 Waving_Tree 影像序列第 242 張到第 259 張之間的移動物體偵測錯誤率，可以發現主體色彩演算法的移動物體偵測錯誤率略高於 Codebook 演算法，但是再加上了區域紋理資訊後就低於 Codebook 演算法了，而 Probability-based 演算法因為動態背景變化偵測誤認的關係導致錯誤率都很高。圖 15 為四種演算法在 Parking_Area 影像序列第 912 張到第 990 張之間的移動物體偵測錯誤率，發現主體色彩演算法與主體色彩演算法加上區域紋理資

訊的移動物體偵測錯誤率依舊很低，而 Codebook 演算法在這個影片中的錯誤率變高了許多，是因為對於移動物體偵測不完全，都只有部份偵測到所導致的結果，而 Probability-based 演算法是背景及亮度變化誤認的關係，錯誤率依舊比較高。表 4 為四種演算法在 Waving_Tree 影像序列第 242 張到第 259 張之間和 Parking_Area 影像序列第 912 張到第 990 張之間的移動物體偵測平均錯誤率，在 Waving_Tree 測試影像序列裡主體色彩演算法的平均錯誤率為 3.79%，後來加上區域紋理資訊的平均錯誤率下降為 3.41%，而 Codebook 演算法的平均錯誤率 5.64%，Probability-based 演算法的平均錯誤率為 17.17%，是因為動態背景及亮度的變化導致誤認為移動物體而使錯誤率最高。而在 Parking_Area 測試影像序列裡 Codebook 演算法因為沒辦法把移動物體完全偵測使得錯誤率上升，平均錯誤率為 0.97%，Probability-based 演算法對移動物體偵測沒有問題，只是會將背景變化誤認為移動物體，所以平均錯誤率為 0.62%，而主體色彩演算法的平均錯誤率為 0.52%，加上了區域紋理資訊後，錯誤率下降到 0.48%。從最後結果看來，主體色彩演算法可以有效地記錄動態背景及亮度上的變化，也能夠正確地偵測移動物體，不會將動態背景誤認為移動物體，後來提出的區域紋理資訊，分析區域紋理分佈來幫住記錄動態背景及亮度的

變化來建構背景模型，使得偵測移動物體不會將背景變化誤認為前景，達到我們的要求。

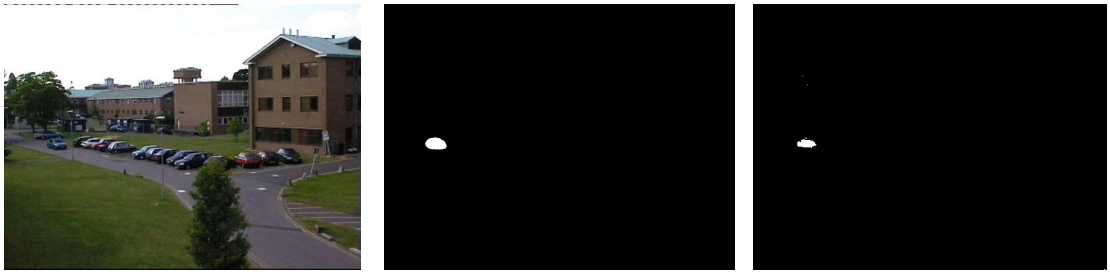
（a）

（b）

（c）

（d）

圖 12 四種演算法在 Waving_Tree 影像序列移動物體偵測的結果，依序為 Frame#246、Ideal Result、偵測結果(a)Codebook(b)Probability-based(c)主體色彩演算法(d)主體色彩+區域紋理資訊演算法



（a）

（b）

（c）

（d）

圖 13 四種演算法在 Parking_Area 影像序列移動物體偵測的結果，依序為 Frame#258、Ideal Result、偵測結果。(a)Codebook (b)Probability-based (c)主體色彩演算法 (d)主體色彩+區域紋理資訊演算法

表 3 四種演算法對測試影像序列(訓練及偵測)所需時間及每秒可執行幀數

四種演算法對測試影像序列(訓練及偵測)所需時間及每秒可執行幀數				
測試影像序列 / 演算法	Waving_Tree		Parking_Area	
	所需時間	每秒可執行幀數	所需時間	每秒可執行幀數
Codebook	1653 s	0.17 fps	26355 s	0.107 fps
Probability-based	1125 s	0.25 fps	23076 s	0.122 fps
主體色彩	116 s	2.47 fps	6721 s	0.422 fps
主體色彩+區域紋理資訊	192 s	1.49 fps	10968 s	0.257 fps

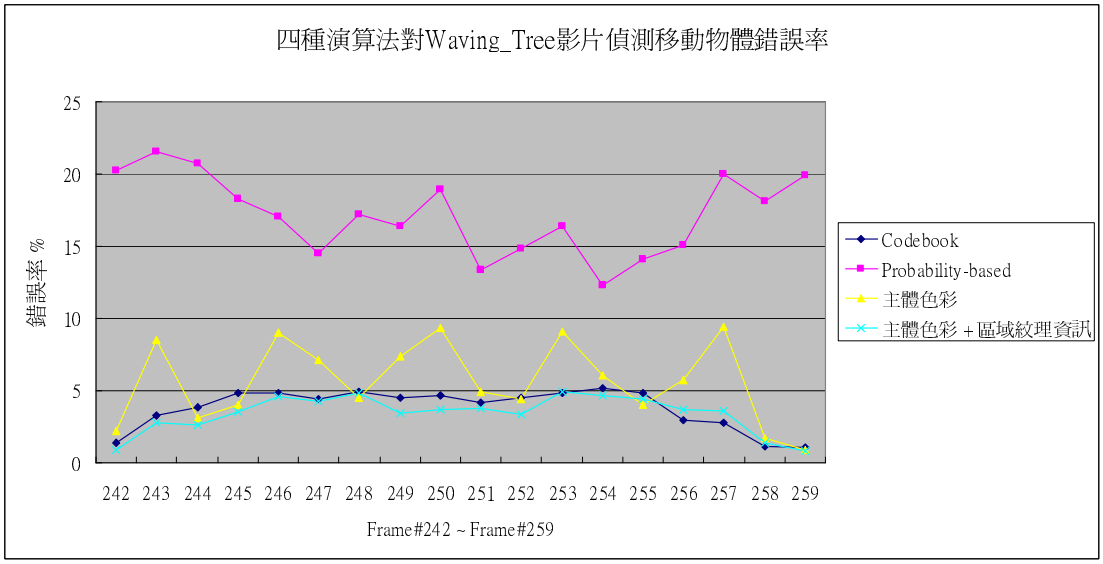


圖 14 四種演算法在 Waving_Tree 影像序列（Frame#242~259）的移動物體偵測錯誤率

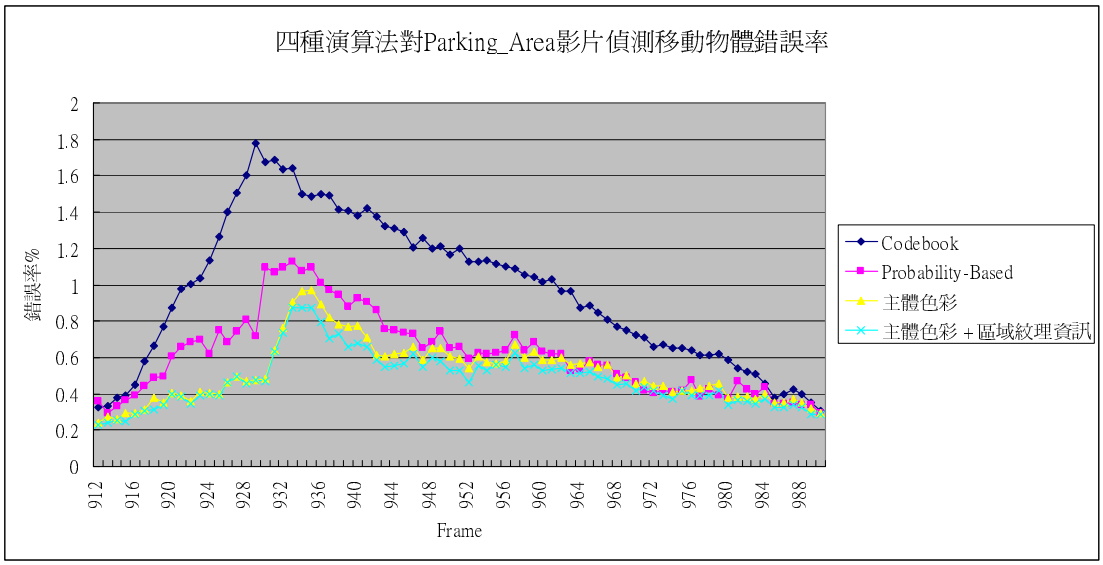


圖 15 四種演算法在 Parking_Area 影像序列（Frame#912~990）的移動物體偵測錯誤率

表 4 四種演算法對測試影像序列的移動物體偵測平均錯誤率

四種演算法對測試影像序列的平均錯誤率		
測試影像序列 / 演算法	Waving_Tree（Frame#242~259）	Parking_Area（Frame#912~990）
	平均錯誤率	平均錯誤率
Codebook	5.64%	0.97%
Probability-based	17.17%	0.62%
主體色彩	3.79%	0.52%
主體色彩 + 區域紋理資訊	3.41%	0.48%

第五章　結論以及未來展望

在本篇論文中我們提出一個使用區域方式來計算影像區域的主體色彩來建構動態背景模型，再加上區域紋理資訊來幫助分析背景及亮度的變化，然後來偵測複雜背景下的移動物體，根據實驗結果發現，我們的方法可以有效地偵測到複雜背景下的移動物體，並且對於複雜背景下的亮度及色彩變化都能有效地處理並且能夠更新背景模型來適應。相較於其他兩個演算法，除了實驗結果比較好之外，花費時間也相對地少，並且我們提出的演算法在處理速度更為迅速。

本研究未來可以改善的地方，對於移動物體偵測準確率更為提昇，移動物體輪廓邊緣的雜訊可以減少或去除，希望能夠讓演算法效能更好、執行速度更佳並且減少處理時間是我們未來期許的目標。

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] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 246-252, 1999.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] F. Porikli, “Detection of Temporarily Static Regions by Processing Video at Difference Frame,” Advanced Video and Signal Based Surveillance, pp. 236-241, 2007.

[17] J. Ding, M. Li, K. Huang and T. Tan, “Modeling Complex Scenes for Accurate Moving Objects Segmentation,” Computer Vision, pp. 82-94, 2010.

[18] C. C. Chiu, M. Y. Ku, and L. W. Liang, “A Robust Object Segmentation System using a Rrobability-based Background Extraction Algorithm,” IEEE Trans. Circuits and Systems for Video Technology, vol. 20, No. 4, 2010.

[19] 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.

[20] 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.

[21] 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.

[22] D. S. Lee, J. J. Hull, and B. Erol, “A Bayesian framework for Gaussian mixture background modeling,” IEEE Int. Conf. on Image Processing 2003.

[23] 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.

[24] 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.

[25] 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.

[26] M. Piccardi, “Background Subtraction Techniques: A Review,” IEEE Int. Conf. on Systems, Man and Cybernetics, Vol. 4, pp. 3099-3104, 2004.

[27] 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.

[28] 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.

[29] 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.