

國立交通大學

多媒體工程研究所

碩士論文

視訊中區塊遞移之研究

A Study of Region Propagation in a Video Sequence

研究生：劉怡伶

指導教授：莊仁輝 教授

中華民國 九十六 年六月

視訊中區塊遞移之研究

A Study of Region Propagation in a Video Sequence

研究 生：劉怡伶

Student : Yi-Lin Liu

指導 教授：莊仁輝

Advisor : Jen-Hui Chuang

國立交通大學
多媒體工程研究所
碩士論文

A Thesis

Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements



Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

視訊中區塊遞移之研究

學生：劉 怡 伶

指導教授：莊 仁 輝 博士

國立交通大學

多媒體工程研究所

摘要

本論文中實作了一個追蹤區塊遞移的演算法，不同於其他相關研究中使用移動向量來預測下一張影像中的區塊位置，我們讓每一張影像中的切割盡可能完整，因而能用較直觀的方式來找到連續影像之間區塊的對應關係。我們提出的方法是基於區塊擴增演算法的基礎，加上一些區塊合併的機制讓相似的區塊合在一起，並透過不同的區塊擴增結果及時間回饋的概念，希望過去好的結果能盡量被保留，讓區塊的大小及形狀更穩定。區塊遞移的判斷利用相同區塊在連續影像中的交集、相似性等容易理解的概念，能達到不錯的結果。最後，我們對區塊遞移的追蹤結果進行分析，以評估所提出的方法是否達到理想中的正確性及穩定性。

A Study of Region Propagation in a Video Sequence

Student : Yi-Lin Liu

Advisor : Dr. Jen-Hui Chuang

Institute of Multimedia Engineering

National Chiao Tung University

ABSTRACT

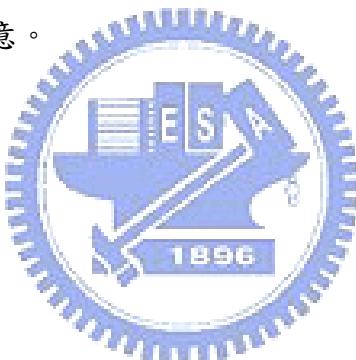
In this thesis, we implement an algorithm for tracking the region propagation in a video sequence. Unlike ordinary objects tracking approaches which detect motion vector of each image block to predict the position of a region in the next frame, we try to focus on reasonable region segmentation in each frame and then track the regions by some simple and straightforward methods. Our segmentation method is based on region growing. To avoid over-segmentation, we use some criteria to merge similar regions. We try to integrate the different segmentation results for same image as well as, by adopting the concept of feedback, from consecutive images. Since developed algorithm can generate stable segmentation results, characteristics of regions' intersection and similarity are used to track these regions of them. Finally, we analyze the tracking results to show that the proposed approach works reasonably well in terms of correctness and stability.

致謝

本論文能夠順利完成，首先必須感謝我的指導教授莊仁輝老師的悉心教導，老師的指導讓我在研究方面受益良多、事半功倍；而老師嚴謹的研究態度，更讓我在研究跟生活上的態度上皆有所啟發。此外，更要感謝三位口試委員，賴飛熊教授、雷欽隆教授以及王才沛教授的意見及指教，讓本篇論文的內容得以更加充實且完善，在此衷心感謝他們。

我還要感謝智慧型系統實驗室的學長、同學、學弟妹們，特別感謝林泓宏學長在課業及研究上的指導；感謝碩二的同學們，一路以來互相幫助；感謝學弟妹們，讓實驗室充滿活力；感謝從大學到現在一起努力的資科 94 級同學；還要感謝在我低潮時給予我很大支持與鼓勵的欣和阿土。兩年的碩士生活裡，因為有你們的陪伴，讓我過得很充實也很愉快，真的非常謝謝你們。

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



目錄

第一章	緒論.....	1
1.1	研究動機與背景.....	1
1.2	相關研究.....	1
1.2.1	切割.....	1
1.2.2	合併.....	4
1.2.3	物體追蹤.....	4
1.3	系統流程.....	6
1.4	本論文各章節介紹.....	7
第二章	影像切割之探討.....	8
2.1	以區塊擴增演算法為基礎.....	8
2.1.1	以點為主的區塊擴增.....	9
2.1.2	色彩模型及距離.....	10
2.2	區塊合併 (merge)	12
2.2.1	平均值和變異量.....	13
2.3	更穩定的切割結果.....	14
2.3.1	增加種子.....	15
2.3.2	移動種子位置的作法.....	17
2.3.3	整合不同種子位置的分割結果.....	18
2.3.4	比較.....	20
2.4	使用時間回饋 (feedback) 的空間整合方法	22
2.5	總結.....	25
第三章	區塊遞移的追蹤.....	27
3.1	交集 (intersection)	27
3.1.1	追蹤的對應關係.....	30
3.1.2	交集的對應結果及討論.....	30
3.2	區塊相似性.....	32
3.3	重新追蹤.....	33
3.4	總結.....	35
第四章	實驗結果與討論.....	36
4.1	成功追蹤的區塊分析.....	36
4.2	區塊穩定性.....	42
4.3	分水嶺演算法.....	43
4.3.1	分水嶺演算法的變形.....	43
4.3.2	分水嶺演算法的合併.....	44
4.4	區域擴增演算法的改進.....	45
4.5	總結.....	48

第五章 結論與未來工作.....	49
參考文獻.....	50

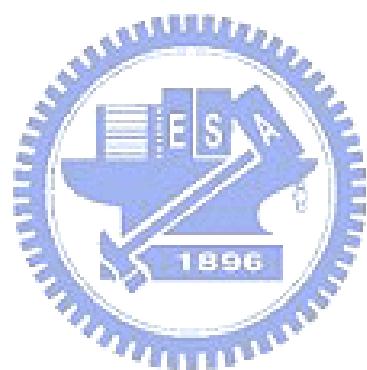


圖 目 錄

圖 1-1	兩階層流程圖。	6
圖 1-2	系統流程圖。	7
圖 2-1	種子間距之比較，(a)間距為 16 個點，(b)間距為 8 個點。	8
圖 2-2	原始影像。	9
圖 2-3	(a) 基本的區塊擴增演算法，(b)區塊擴增演算法有加入門檻。	10
圖 2-4	色彩模型不同對區塊擴增演算法的比較，(a)RGB 色彩模型，(b)HSI 色彩模型。	11
圖 2-5	(a) 第 n 張影像的原圖，(b) 第 $n+1$ 張影像的原圖，(c) 第 n 張影像經過區塊擴增演算法的結果，(d) 第 $n+1$ 張影像經過區塊擴增演算法的結果。	12
圖 2-6	區塊平均值分析：(a)基本的區塊擴增演算法，(b)有加入門檻之區塊擴增演算法。	13
圖 2-7	區塊變異量分析：(a)基本的區塊擴增演算法，(b)有加入門檻之區塊擴增演算法。	14
圖 2-8	連續影像(a)~(f)的區塊合併結果。	16
圖 2-9	在區塊周圍增加種子(a)之前的作法，(b)增加後的種子圖(白點)，(c)區塊擴增的結果，(d)區塊合併的結果。	16
圖 2-10	種子位置示意圖。(a) 種子放置於網格之左上角，(b) 種子放置於網格中間。	17
圖 2-11	移動種子位置所得之結果比較。(a) 種子左上角，(b) 種子放中間。	17
圖 2-12	(a)原圖，(b)種子放於網格左上角之位置圖，(c)區塊擴張，(d)區塊合併的結果，(e)~(g)為種子放於網格中間之各項結果。	18
圖 2-13	不同區塊分割的演變。	19
圖 2-14	整合後之結果。	20
圖 2-15	(a) 加倍的種子位置圖，(b) 種子加倍的區塊擴增及合併結果。	20
圖 2-16	兩種影像切割流程圖。(a)方法一的流程圖，(b)方法二的流程圖。	21
圖 2-17	整合不同 seed 位置的影像，(a)~(e)種子位置在左上角的連續五張影像，(f)~(j)種子位置在中間的連續五張影像，(k)~(o) 連續五張影像經過整合左上角及中間的結果。	23
圖 2-18	時間和空間整合方法流程圖。	24
圖 2-19	移動區塊變化示意圖。	24
圖 2-20	時間和空間整合方法在連續影像上的切割結果比較，(a)~(e) 種子位置在左上角的連續五張影像，(f)~(j) 連續五張影像經過整合左上角及中間的結果，(k)~(o) 連續五張影像經過時間和空間整合方法的結果。	26
圖 3-1	交集示意圖。	27
圖 3-2	合併之後的影像追蹤。	29

圖 3-3 追蹤對應方式之比較，(a)單向追蹤，(b)雙向追蹤。	30
圖 3-4 考慮交集的追蹤方法，原圖(左)，空間整合方法(中)，時間和空間整合方法(右)。	31
圖 3-5 可能遺失追蹤的交集示意圖。	32
圖 3-6 由左至右，原始影像(左)、時間和空間整合方法用交集追蹤方法(中)和材質與交集的追蹤方法(右)所得之結果。	34
圖 3-7 連續五張影像以相似性判斷新的區塊。	35
圖 4-1 各演算法成功追蹤區塊數之變化。	
...37	
圖 4-2 各演算法成功追蹤面積之變化。	37
圖 4-3 原圖(左)、區塊擴增與合併之結果用交集追蹤方法(中)，和利用相似度與交集門檻的追蹤方法(右)，所得之結果。	38
圖 4-4 原圖(左)、空間整合方法用交集追蹤方法(中)和材質與交集的追蹤方法(右)，所得之結果。	39
圖 4-5 相似度追蹤錯誤的例子。	40
圖 4-6 原圖(左)、時間和空間整合方法用交集追蹤方法(中)和材質與交集的追蹤方法(右)，所得之結果。	41
圖 4-7 單一區塊追蹤面積變化圖。	42
圖 4-8 單一區塊追蹤狀況，(a)第 1 張影像，(b)第 33 張影像。	43
圖 4-9 分水嶺演算法初步切割結果。	44
圖 4-10 分水嶺演算法經過區塊合併的結果。	44
圖 4-11 有限制的區域擴增結果。	45
圖 4-12 種子位置示意圖。	46
圖 4-13 動態給予種子的區域擴增結果。	47
圖 4-14 原圖(左)、時間和空間整合方法用相似度追蹤方法(中)和動態給予種子的區域擴增與時間整合以相似度追蹤方法(右)，所得之結果。	48

第一章 緒論

1.1 研究動機與背景

影片 (video) 中物體的追蹤運用十分廣泛，例如壓縮時若能清楚知道重要物體的位置和移動方向，可以對前景及背景採不同程度的壓縮，以減少記憶體使用量。因此目前有很多關於影片追蹤的研究，其中大部分的作法是粗略切成區塊後，估算其移動向量，藉以推算出下一張影像中區塊的分佈情形。在我們之前的研究中，為了提高估算的速度，將單張影像切為許多方格子作為超點 (super-pixel) 來計算移動向量，遇到不規則的物體邊界時，可能需要增加格子密度來更加逼近物體邊緣，但仍然無法十分貼近邊緣。因而我們從中得到一些靈感，希望透過一些切割的演算法讓影像成為許多超點的集合，以便於之後的追蹤。因此本篇論文中，首重在區塊的切割，在此我們所希望的區塊是盡可能讓單一物體屬於同一區塊，等同於希望能直接切出物體，以物體為單位進行追蹤，而我們最終的目標是在連續的影像上，對特定或是有意義的區塊進行追蹤。



1.2 相關研究

在影片的追蹤方面已有許多的論文被發表，使用的方法也不盡相同，多數的方法中必須先將單張影像中的物體切出，或者是切割成更多的區塊來組成一個物體，因此在本節的第一部份中，我們選出一些關於切割 (segmentation) 的相關研究論文。接著，在第二部分介紹目前已有的一些物體追蹤方法，瞭解其優缺。

1.2.1 切割

目前切割的演算法可以分為四種，設定門檻的方式、基於邊界 (boundary) 的方法、基於區塊 (region) 的方法和混合邊界跟區塊的方法。設定門檻的方式是將灰階值落於相同門檻的點認定為同一個區塊，此方法無法處理有雜訊及模糊的邊

界。第二種基於邊界的方法是使用相鄰兩點灰階值變化的大小來判斷，形同對影像作梯度計算，梯度值大的部分就認為是可能的邊界點。第三種基於區塊的方法，則是希望相同區塊的點能夠有相近的灰階值，1.1.1.1 節的區塊擴增演算法（region growing）即是屬於此種。而 1.1.1.2 節的分水嶺演算法（watershed）則是屬於第四種的方法。

1.2.1.1 區塊擴增演算法

Rolf Adams et al.[1]提出一種給予種子的區塊擴增演算法，首先選擇一些種子，將種子分為 n 個集合，表示為 A_1, A_2, \dots, A_n ，集合的數目等於影像切割完成後的區塊數目。每一次在那些集合內的點的鄰居中挑選出下一個要加入的點，這些可以加入的候選點可以表示成：

$$T = \left\{ x \notin \bigcup_{i=1}^n A_i \mid N(x) \cap \bigcup_{i=1}^n A_i \neq \emptyset \right\} \quad (1-1)$$

$N(x)$ 表示點 x 的鄰居。接著必須有個判斷式來選擇哪一點能加入集合中，最簡單的方法是計算 x 點和區塊 A_i 的灰階平均值差，表示為：

$$\delta(x) = |g(x) - \text{mean}_{y \in A_i(x)}[g(y)]| \quad (1-2)$$

並且在所有與 x 相鄰的區塊中，選出 $\delta(x)$ 最小的作為 x 加入之候選區塊。然後才從集合 T 中選出 $\delta(x)$ 最小的點加入集合中，不斷重複上述步驟直到整張影像都加入區塊的集合，即完成影像的切割。

如果還想增加判斷式的準確，可以加入 variance 來考慮，改變判斷式如：

$$\delta(x) = \frac{|g(x) - \text{mean}_{y \in A_i(x)}[g(y)]|}{\text{SD}_{y \in A_i(x)}[g(y)]} \quad (1-3)$$

或者希望不止針對灰階影像，只要稍對判斷式做更改，即可用於彩色影像中。區塊擴增演算法的優點為不受影像中物體形狀的影響且可用於任何維度之影像。

1.2.1.2 分水嶺演算法

分水嶺演算法是一種被廣泛運用的切割演算法，他是屬於區塊擴增演算法的一種[2]。分水嶺演算法在一開始會先偵測影像中梯度的最小值點，以這些最小值為各個區塊的中心，慢慢向周圍梯度值相近的點擴張，當碰到來自於另一個最小值擴張出來的點時，則定義此點為這兩個區塊的邊界。若所有區塊都互相碰到形成邊界後，這張影像就切割完畢了。

使用分水嶺演算法的第一個問題是如何計算影像的梯度，過去經常使用的方法是用型態學的梯度來計算，型態學梯度的定義如下：以 $f(x, y)$ 表示影像輸入， M_n 是一個大小為 n 的平坦化濾器。

$$\text{Erosion} : \varepsilon_n(f)(x, y) = \text{Min}\{f(x + x_0, y + y_0), (x_0, y_0) \in M_n\} \quad (1-4)$$

$$\text{Dilation} : \delta_n(f)(x, y) = \text{Max}\{f(x - x_0, y - y_0), (x_0, y_0) \in M_n\} \quad (1-5)$$

$$\text{Morphological gradient} : g(f)(x, y) = \delta_1(f)(x, y) - \varepsilon_1(f)(x, y) \quad (1-6)$$

不過這個方法會流失許多資訊，尤其是在影像有變化時，很可能造成型態學梯度有些許位移或是邊界太厚的情況。因此，另一種作法是直接採用影像的灰階值來計算兩點的相似度距離，不再多一次額外處理影像的梯度，例如F. Meyer[3]的方法中，以RGB三個頻道的最大差值相減作為顏色距離的量測準則，同時也將分水嶺演算法運用到彩色影像上。



然而分水嶺演算法經常使得影像過度的切割，目前主要有兩種方式來克服過度的切割，第一種是切割之後利用一些條件的限制將相鄰的區塊合併；第二種則是以一些點的集合表示平坦的區塊，取代傳統方法中最小值點角色。第一種方法以不同係數的高斯模糊得到切割程度不同的結果[4]，利用梯度值的極值比例分析控制其切割的細碎程度，如果能夠預期每個區塊平均的臨邊數，便能輕易的推出所需之細碎度，也就是此影像所需的模糊係數。而H. Gao et al.[5]屬於第二種方法，先使用上述之型態學梯度定義，將梯度值較小的認定為平坦區塊，標記這些區塊中的點為 0，其餘的則為 1。得到以二位元形式標記好影像之後，以標為 0 的這些點為中心向外做分水嶺演算法，即可得到較好之切割。另外，因為型態學梯度經常只用於亮度

(luminance)，為了避免亮度很小時的錯誤判斷，這篇論文中採用取亮度和顏色梯度中較大的作為最後的影像梯度。

除了以不同的梯度或準則來控制其切割結果，在許多的相關研究論文中也嘗試使用不同的顏色模型來切割，例如F. Meyer[3]以HSI，H. Gao et al.[5]則是採用CIE L*a*b*，一般而言將色度和亮度分開的顏色模型會得到比較好的切割結果。

1.2.2 合併

切割演算法所造成的過度切割，必須有一些合併的機制來處理。在合併的方法中大致可以分成二種：第一種是利用切割結果的區域或邊界性質來合併[6]，根據區塊邊界的連續性及梯度值決定邊界的強度，並取一個門檻值將比較弱的邊界視為不必要的邊界；此外，也利用區塊的內容算出材質或顏色的特性，將特性差不多的區塊加以合併。

第二種方法主要利用區塊的相關位置來合併[7]，首先將切割結果的區塊表示成相鄰區域圖 (region adjacency graph)，以點 (node) 代表區域，相鄰區域則有邊 (edge) 相連，並將兩區域的相似度作為邊的權重，對於每個點取其最小的邊來建出一個有方向性的最近鄰居圖 (nearest neighbor graph)。然後利用一些性質，例如最近鄰居圖中存在有迴圈則合併。



1.2.3 物體追蹤

在物體追蹤相關的論文中，大致上可以分為五種：(1)基於模組的方法，(2)基於肉眼看的到的資訊的方法，(3)基於物體輪廓的方法，(4)使用物體特徵的方法，(5)混合上述四種的方法。

在第一種基於模組的方法中，需要一個預先知道的物體形狀[8]，根據已知的物體形狀即使物體有部分被遮蓋也能正確的追蹤到，這個方法的缺點是如果物體形狀未知則無法追蹤，因而使用時機受到限制，缺乏一般性。第二種「基於肉眼看的到的資訊的方法」是利用區塊中的資訊追蹤前後兩張圖中相似的區塊，使用的資訊包

括區塊的移動、顏色和特徵。例如D. Wang [9]在使用分水嶺演算法完成初步的切割之後，將擁有相同移動向量且相鄰的區塊視為一個物體。以物體為單位估算出下一張圖中物體的位置，再用每個點的灰階值計算投射誤差：

$$f(x, y, t) - f(x + dx, y + dy, t + 1) \quad (1-7)$$

使得下一張圖能正確的切出每個物體的範圍。這一類型的方法最大的缺點就是當物體變形過多就會無法正確追蹤。

第三種基於物體輪廓的方法不用區塊中的點來追蹤，而是追蹤物體的邊緣[10]，一開始需要知道物體大概的輪廓，計算其內、外部的能量來逼近正確的輪廓，此處的能量是用輪廓的微分和灰階的梯度等資訊求出。因為採用慢慢逼近正確位置的作法，預估的輪廓不能離真正的位置太遠，因此這類的方法不允許物體移動得太多或是物體被遮蔽，但能容許物體微小的變形。有別於第二種方法使用區塊的資訊，第四種方法使用物體的特徵，例如以物體的角作為特徵點來追蹤[11]。不過這個方法很大的困難在於如何決定特徵點屬於哪一個物體。

最後一種方法是希望能混合上述四種的優點來克服缺點，A. Cavallaro et al.[12]使用兩個階層：區塊和物體，如圖 1-1。在每一張圖中切出物體之後分成許多個不重疊的區塊，由第一張圖切出的區塊分別計算其重心、移動向量和灰階值平均及變異量，用這些特徵代表一個區塊，減少逐點計算的複雜度。以區塊為單位估算下一張圖的位置，在和原圖切割出來的區塊做對應，找出正確的切割。最後再切割的結果以區塊為單位做時間方向的特徵點連結，意即利用區塊的特徵決定追蹤的關係。此處以區塊為單位的優點就是，即使因為切割結果不好使得一物體在某些時間點被分成兩個不連接的物體，還是能在特徵點連結這個步驟判定這兩個物體是屬於同一個物體的。

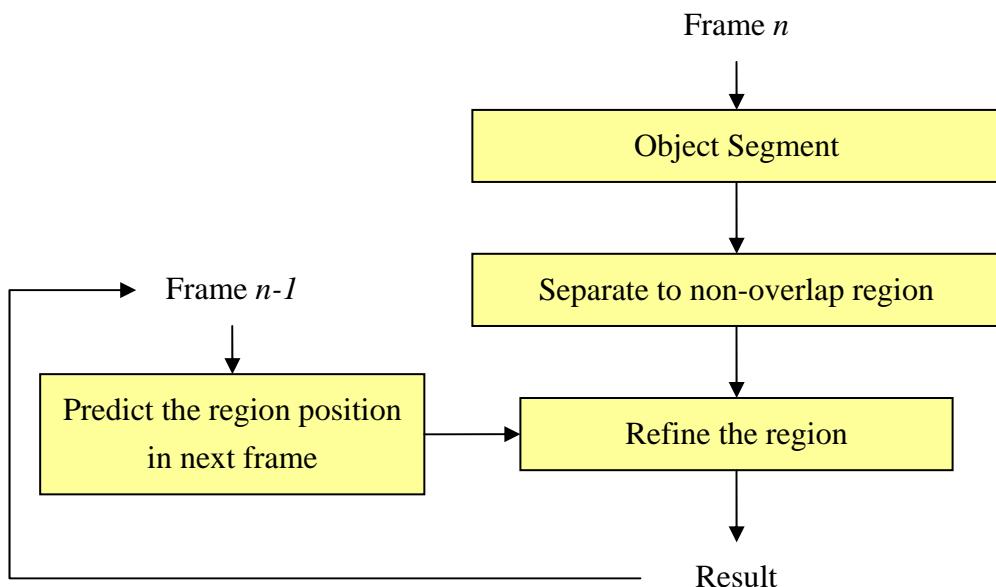
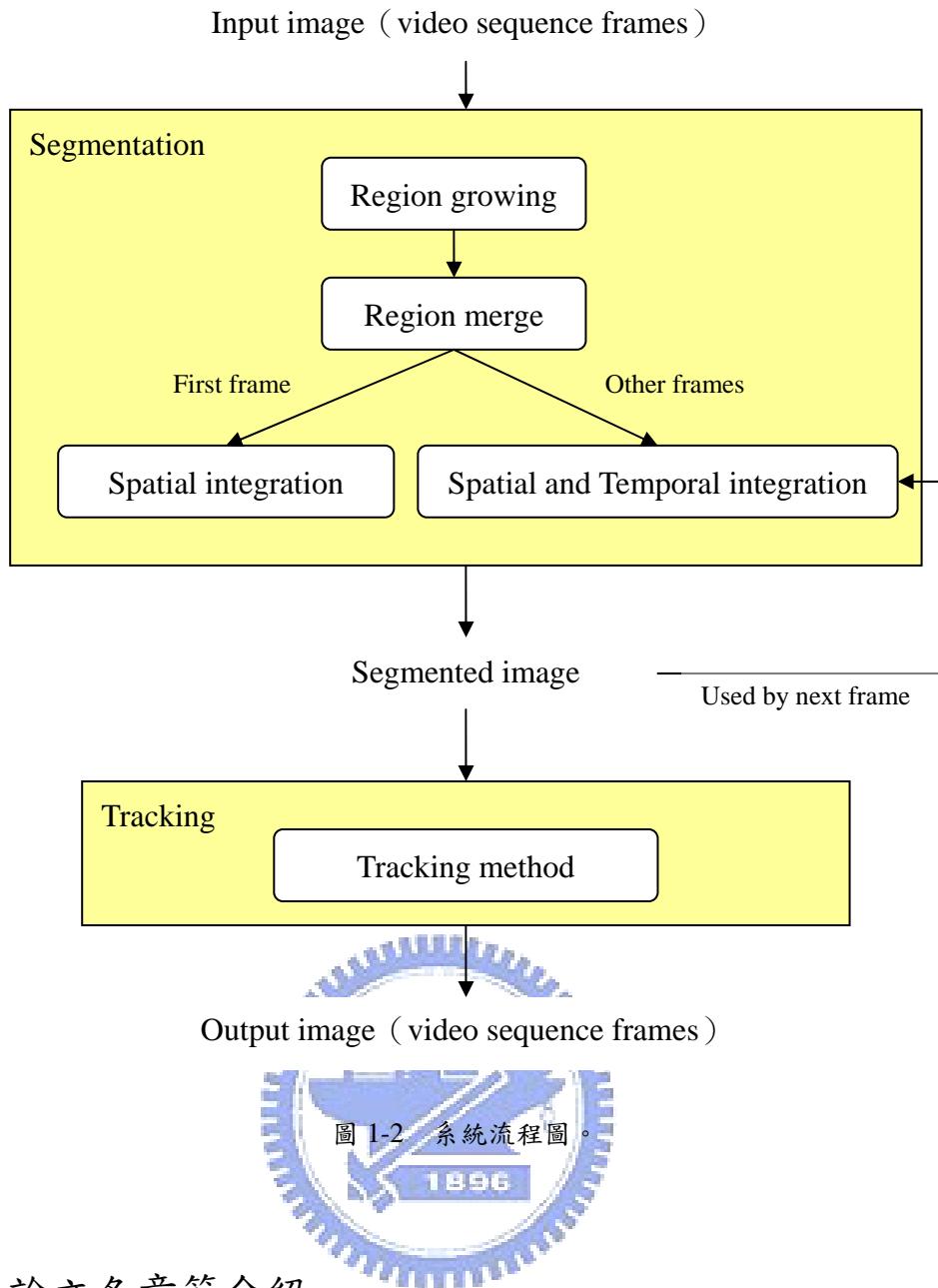


圖 1-1 兩階層流程圖。

1.3 系統流程

圖 1-2為我們所建之系統的流程圖。首先將影片輸出為連續多張的影像，以這些影像作為系統的輸入，接著運用一些方法對影像做區塊的切割，在我們的切割處理中主要是以區塊擴增（region growing）演算法作為基礎，之後在後處理的步驟中透過比對區塊中顏色的分佈情況，以及不同結果中區塊交集的情況，用以決定哪些區塊需要合併成為一個大區塊。在追蹤的部分，我們同樣的利用區塊的顏色、交集來判斷區塊在前後張影像的對應關係，而未計算其移動向量。



1.4 本論文各章節介紹

本篇論文全文共分為四章，除了本章為緒論外，第二章介紹影像中所使用的切割方法，內容包含延伸的區塊擴增演算法、區塊合併機制的方法介紹，此外我們延伸基本的方法，試圖讓切割的結果更好。第三章介紹時間方向上的追蹤。第四章是一些實驗的結果評估。第五章為結論並說明本研究未來的一些研究方向。

第二章 影像切割之探討

在過去的研究中，我們曾經將影像切成許多大小相同的網狀方格子，並以這些方格子代替點來做物體的追蹤，然而由於方格子無法準確的框出物體的形狀，有時就會使得移動的估計產生錯誤。因此，在本論文中我們運用一些區塊切割的演算法希望將影像切成許多的小區塊，以這些區塊作為超點（super-pixel）來估算物體的移動，來增進運動估計的正確性。

在本章中，我們將針對如何切割影像區塊來做探討。2.1及2.2節包含了基本的區塊擴增（region growing）和區塊合併演算法，2.3節中更深入的探討如何進一步改善基本區塊擴增的結果，2.4節中我們提出一個時間方向上的整合分析機制來提高我們切割結果的正確性（影像區塊的完整性）。

2.1 以區塊擴增演算法為基礎

由於區塊擴增演算法是以種子（seed）的數目控制最終區塊的數目，區塊數目太多可能使得追蹤、比對上花費大量的時間，數目太少則可能使得區塊跨過物體邊界，使得切割結果不佳，如圖 2-1(a)紅色框中所示。經過我們測試的結果，這系列的影像採用種子間距為 8 個點時能有較好的結果，如圖 2-1(b)，因此在之後的部分，我們皆採用種子間距為 8 個點來切割。



圖 2-1 種子間距之比較，(a)間距為 16 個點，(b)間距為 8 個點。

在我們希望以區塊作為超點的目標下，如果每張圖能夠形成大小相似、區塊總數差不多的結果，則對於作為超點之表示方式應該是比較好的結果。因此，我們將過去的方法延伸，在每一張影像中放置均勻的格子點，以這些格子點為中心向鄰近點中，色彩距離最接近的點開始擴張區塊，希望能得到類似格子點但形狀更逼近物體的結果。

2.1.1 以點為主的區塊擴增

在過去的區塊演算法中，每一次都納入有最小色彩距離 $\delta(x)$ 的點，然後一直到所有點都計算過，且加入某個區塊為止。這樣的作法能讓每個區塊都是封閉且完整的，如原始影像圖 2-2 區塊擴增後的結果如圖 2-3(a)，其中有相同顏色的部分代表屬於相同的區塊。但是如果有一些雜訊點或者是邊界的部分被勉強加入，將會影響到整個區塊的色彩平均值或變異量，對於我們後處理時的合併或是追蹤，都有造成誤差的可能。



圖 2-2 原始影像。

這時候比起區塊是否完整和封閉，我們更重視在區塊中的點是否非常相似，故我們以點為主優先考慮區塊的相似性，每一次加入之前對判斷式取一個門檻 (threshold)，如果 $\delta(x)$ 大於這個門檻，則不加入任何一個區塊中，因此我們可以將區塊擴增演算法中可加入區塊的候選點改為：

$$T = \left\{ x \notin \bigcup_{i=1}^n A_i \mid N(x) \cap \bigcup_{i=1}^n A_i \neq \emptyset \cap \delta(x) < \text{Threshold}_t \right\} \quad (2-1)$$

如圖 2-3(b)，我們可以發現到物體邊緣都被排除在區塊外的黑色部分，在下一節中，我們將更深入探討、使得分割結果能延伸至超點的應用。



圖 2-3 (a) 基本的區塊擴增演算法，(b)區塊擴增演算法有加入門檻。

2.1.2 色彩模型及距離

彩色影像分析中有好幾種經常使用的色彩模型，其中 RGB Model 是最常使用的，概念上是用 red、green、blue 三原色的組合來表示各個色彩。另外還有其他跟 RGB 模型概念不同的色彩模型，例如 HSI 是將色度(chrominance)和亮度(luminance)分開的一種色彩模型，在 1-2 節相關研究中曾提到，一般在色彩切割時若採用色度和亮度分離的色彩模型會有較好的結果，因此，此處我們採用 HSI 的色彩模型。而經由一些數學的運算，即可將色彩由 RGB 空間轉換到 HSI 空間，轉換的公式如下：

$$H = \begin{cases} \theta & , \text{if } B \leq G \\ 360 - \theta & , \text{otherwise} \end{cases} \quad (2.2)$$

$$S = 1 - \frac{3}{(R+G+B)} * [\min(R, G, B)] \quad (2.3)$$

$$I = \frac{1}{3} * (R+G+B) \quad (2.4)$$

其中 $\theta = \frac{0.5 * [(R-G) + (R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{0.5}}$ 。我們由(2.2)~(2.3)可以觀察得知，HSI 是以色調(hue)、飽和度(saturation)、亮度(intensity)三者作為組合色彩的元素。使用 HSI 色彩表示法，可將色彩的色調、飽和度、亮度分離成三個元素，各自獨立去

觀察，而不同於 RGB 色彩模型是將色彩的色調、飽和度、亮度等資訊混合在一起，因此 HSI 模型對於色彩的變化情形經常可以表現得更為清楚。

區塊擴增演算法中需要計算相鄰兩點的色彩距離，用以作為挑選下一個加入點的判斷式，在參考論文[3]中，採用 RGB 中差值最大的一個作為兩點色彩上的距離，但是這樣的判斷式未用到所有色彩資訊因而不夠精確，所以我們除了將色彩模型改為 HSI，並採用 HSI 三者差值的絕對值相加取代之前 RGB 定義的色彩距離，判斷式可表為：

$$\delta(x) = \left\{ \begin{array}{l} |H(x) - H(y)| + |S(x) - S(y)| + |I(x) - I(y)| \\ y \in A_i \& x \in N(y) \& x \notin A_i \end{array} \right\} \quad (2.5)$$

圖 2-3 是兩種色彩模型的區塊擴增演算法之結果，(a)是RGB色彩模型，我們可以觀察到其結果在一個物體內可能較為細碎，如紅色框標示處，但是對於右下角的火車RGB的表現則比(b)HSI的結果好，是因為HSI在I值極小時H、S的變動可能很大而造成誤判，故在I值極小時我們必須忽略H、S的變動。

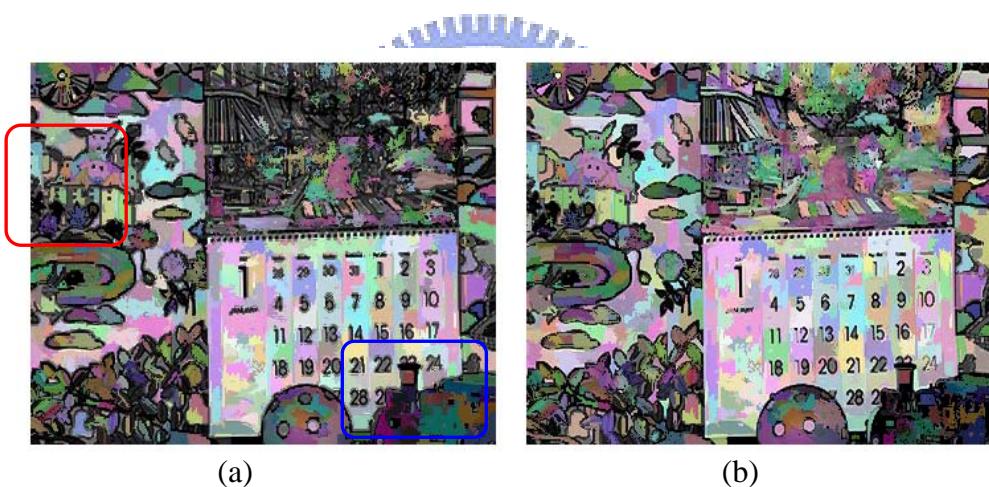


圖 2-4 色彩模型不同對區塊擴增演算法的比較，(a)RGB 色彩模型，(b)HSI 色彩模型。

2.2 區塊合併 (merge)

經過區塊擴增的區塊若是作為超點，我們希望能夠有較穩定的切割結果，也就是希望在連續的兩張影像中，即使物體有些許移動也能夠有差不多大小及形狀的切割結果。圖 2-5(a)(b)是連續兩張影像的原圖，(c)(d)是在種子在同一個位置的切割結果，左半邊是稍微往右移，而右半邊的月曆是往上移。比較切割結果之後我們可以發現即使物體只有些微的移動，在物體邊緣的部分相鄰兩張圖亦有可能切出差距很大的區塊。如圖 2-5(c)中顏色平坦的月曆是切成幾個類似矩形的區塊，在(d)中卻是將左上角切成一角型的帶狀區塊。因此，我們瞭解到使用區塊擴增演算法並沒有很穩定的切割結果，如果以其所切割的區塊作為超點來追蹤並不適合。

從圖 2-5的結果中，我們可以發現雖然單一區塊的穩定性不好，可是對於一個物體整體範圍而言，區塊擴增演算法往往也可以切出不錯的結果。因此我們考慮做一些區塊合併的動作，希望讓屬於同一個物體的區塊能合併在一起，使後續之追蹤演算法能以物體為單位來進行。

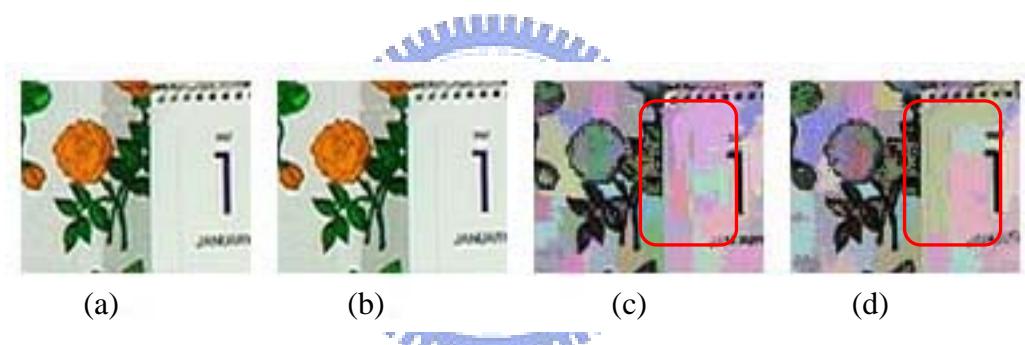


圖 2-5 (a) 第 n 張影像的原圖，(b) 第 $n+1$ 張影像的原圖，(c) 第 n 張影像經過區塊擴增演算法的結果，(d) 第 $n+1$ 張影像經過區塊擴增演算法的結果。

一般而言，考慮兩個區塊是否能夠合併，通常是比較兩個區塊的相似程度或者是區塊邊界的連續性，前段曾提到我們希望以點為主體使得連續圖片中區塊的分割結果在內容上盡可能相似，所以在此我們是優先考慮較能表現區塊材質 (texture) 的平均值和變異量，而不是採用區塊邊界的連續性。

2.2.1 平均值和變異量

透過平均值和變異量我們可以知道區塊內各點的顏色的分佈情形，所以比對這兩個特徵可以瞭解各區塊的相似程度。為了瞭解經由區塊擴增演算法切割好的區塊對應於相同物體者之平均值和變異量的分佈，以下我們試著計算每個區塊和與其最相似鄰近區塊的平均值之差，並以 8 為一個區間，每一個區間給予不同的顏色，試圖觀察同一個物體中的區塊是否能落於同一區間。

圖 2-6(a)是沒有對區塊擴增演算法取門檻的分析結果，其中單一個物體內部點可能因為跨多個區間，而呈現幾種不同顏色的區塊，顯示其物體內的平均值差異較大，從影像左上角的四隻小豬可以明顯看出。圖 2-6(b)是有在區塊擴增演算法中取加入門檻的分析結果，相同物體的區塊其平均值幾乎落於同一個區間內，也就是物體幾乎為同一顏色。這個結果印證了我們2.1.1節中以點為主限制區塊擴增演算法(2-1)的假設。

圖 2-7和圖 2-6類似，但採用區塊變異量的值來顯示其顏色。我們可以發現，屬於相同物體的區塊擁有近似的變異量，因此我們利用這個結果並配合之前的平均值分析來做區塊的合併，作為本論文採用之區塊合併演算法：首先找出相鄰兩區塊中平均值差異最小的，判斷其變異量的差異是否小於我們設的門檻，如果符合條件，則將兩個區塊合併，否則即視為不同物體而不合併，而後一直重複這個動作直到區塊平均值的差異超出我們可以容忍的範圍。



圖 2-6 區塊平均值分析：(a)基本的區塊擴增演算法，(b)有加入門檻之區塊擴增演算法。



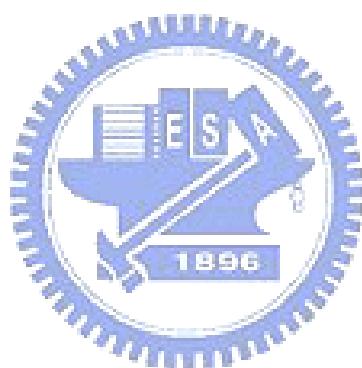
圖 2-7 區塊變異量分析：(a)基本的區塊擴增演算法，(b)有加入門檻之區塊擴增演算法。

2.3 更穩定的切割結果

區塊擴增演算法雖然每一次都是選距離最近的點加入，但是在遇到很平坦的影像時，加入的順序不但和距離有關，也和種子的位置有關。假設有 A 、 B 兩個種子， A 、 B 之間的每一個點都有相同的距離值，此時這些點被檢查到的先後次序會直接影響加入 A 或 B 所擴增的區塊集合之先後順序，亦即越遠的點會越慢加入，因此在平坦的區塊中種子的位置會和區塊擴增演算法的結果有密切的關係，而區塊擴增演算法的結果也會影響到區塊合併的結果。如圖 2-8 是連續幾張影像經過區塊合併演算法的結果，我們盡可能將相似的區塊合併在一起，仍然得到每一張影像最後的切割狀況都不相同（如紅框標示區域），這一連串影像的原圖左半邊是整體一起移動，每個區塊的相對關係和材質幾乎沒變，放置的種子位置也沒有變，以種子的相對位置而言，是左半邊區塊位移使得種子在區塊的相對位置上改變。如此，亦印證了我們前面所認為的，在平坦區塊中，種子位置對區塊擴增的結果有密切關係。在本節以下的討論中，我們希望夠過調整區塊擴增演算法中種子的位置，讓區塊切割的結果能夠更佳穩定，以提高後續物體追蹤之成功率。

2.3.1 增加種子

從前一節的結論可知種子放置的位置會對影像切割的結果產生一些的影響。因此我們第一個想法是，如果在切割好的物體外部再多放置一些種子，是否能讓原本沒有被正確包括進來的區塊再被加進來？於是我們嘗試在圖 2-8(a)下方圓球的外圍部分加一些種子進去，亦即在圖 2-9(a)綠色那塊的球區塊邊緣多放上一些種子，如圖 2-9(b)球的周圍多了一些白點（種子），然後再重新做一次區塊擴增和區塊合併演算法，而得到圖 2-9(c)(d)的結果。我們發現增加種子雖然在區塊擴增時能更接近邊緣，但同時也會造成經過區塊擴增演算法之後的區塊過小，導致其平均值及變異量容易因為少數點而有巨大改變，不能客觀的表現出區塊的材質特性，反而有無法順利的進行合併的情形，使得結果並不如預期、也未優於原本的方法。



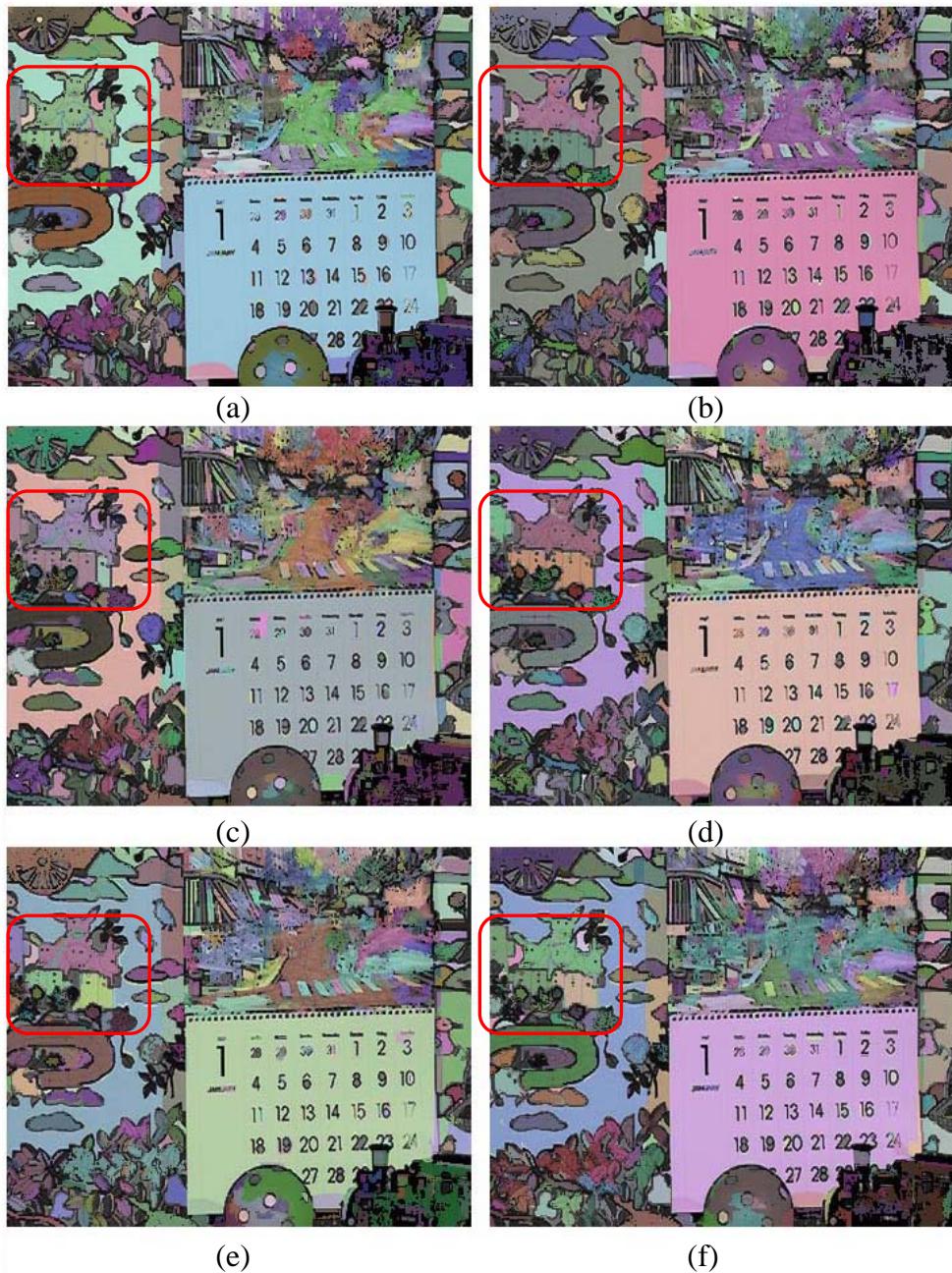


圖 2-8 連續影像(a)~(f)的區塊合併結果。

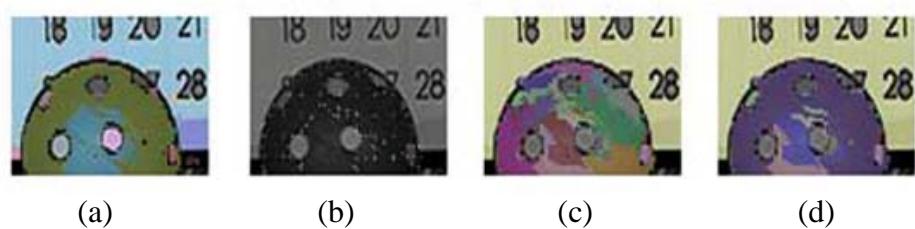


圖 2-9 在區塊周圍增加種子(a)之前的作法，(b)增加後的種子圖（白點），
(c)區塊擴增的結果，(d)區塊合併的結果。

2.3.2 移動種子位置的作法

前一節中我們增加種子數目卻使得區塊太小、不利於合併，而無法有較好的結果，但是從圖 2-8 中可以肯定種子位置會影響區塊擴增演算法的結果，於是在不增加種子數量的前提下，我們嘗試將種子移動到不同的位置。原本我們是將種子放在格子點的左上角，如圖 2-10(a)，此處我們試著將種子放在格子的中間，如圖 2-10(b)。

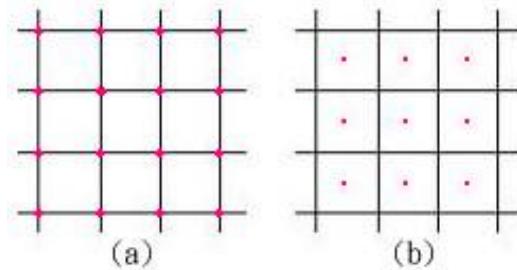


圖 2-10 種子位置示意圖。(a) 種子放置於網格之左上角，(b) 種子放置於網格中間。



圖 2-11 移動種子位置所得之結果比較。(a) 種子左上角，(b) 種子放中間。

圖 2-11(a)、(b)是同一張影像分別將種子放在格子左上角跟中間的位置，經過區塊擴增、合併的結果，我們可以發現切割的結果大致上是相同的，但是在圖 2-11(b)影像中間兩隻鳥及上面的豆子狀物體（紅色框中）被切成兩半，這是由於物體本身有光影的影響，圖 2-12(a)原圖中明暗分隔處顯示其光影的不同。圖 2-12(b)、(c)、(d)分別是種子放置於網格左上角的種子位置、區塊擴增演算法的結果及區塊合併演算

法的結果，而圖 2-12(e)、(f)、(g)則是將種子放置於網格中間所得之子位置、區塊擴增演算法的結果及區塊合併演算法的結果。前者區塊合併之後較沒有受到光影的影響，是因為種子位置落於光影界線附近、或是光影左右都有種子分佈於物體內，使得區塊擴增的結果並沒有完全切割於光影的線上，或者並不會有如圖 2-12(f)鳥的左半部分，有一些屬於物體的地方完全不屬於任一區塊。這樣才能讓區塊之間的平均值、變異量…等條件差異不會過大，而有合併的機會。

當然我們也是可以放寬區塊合併時對於平均值及變異量的門檻限制，但是同時可能會使得一些顏色相近，卻屬於不同物體的區塊合併。此外，若是在區塊擴張階段沒有被歸到任一個區塊的點，就會永遠的遺失。但是我們無法預期種子放置於那個位置會有較好的結果，因此如果能將兩張的優點合併，應能避免種子位置不佳而造成的不穩定狀況。

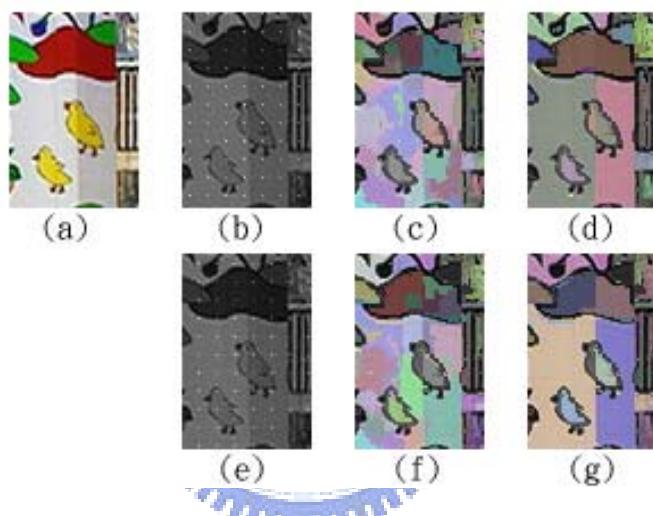


圖 2-12 (a)原圖，(b)種子放於網格左上角之位置圖，(c)區塊擴張，(d)區塊合併的結果，(e)~(g)為種子放於網格中間之各項結果。

2.3.3 整合不同種子位置的分割結果

在2.1.1節中，我們對區塊擴增的判斷式加入一個門檻的限制，因而能夠增加區塊擴增演算法不跨過物體邊界的可信度。在此，我們假設區塊不會跨出物體邊界太多，如果將種子位置不同所得出之兩張區塊分割結果拿來比較，兩者區塊的關係大致分成兩種狀況：

(1) 某一小區塊被涵蓋於另一張圖的某一區，如圖 2-13(b)的區塊每個區塊都被涵蓋在(a)中。這時候我們認為(a)是較好的切割結果，但是(b)的四個區塊合起來後可能在邊緣的部分會略大於(a)，因此某點 x 若是屬於(a)的區塊中或是(b)四個區塊的集合中，就認為點 x 應是屬於這個物體的。

(2) 如圖 2-13(c)和(d)為兩種切割結果，其各自由幾個區塊組成一物體，但是切割結果的區塊大小和形狀皆不同。這時我們必須將(c)的區塊中和(d)同一區塊有交集的區塊盡可能合併在一起，反之對(d)也是一樣。重複做了幾次之後，期待能達到(c)和(d)中區塊合在一起如同(a)的結果。

此外合併的時候為了避免切割時邊界認定的不同，造成屬於不同物體的區塊有微小交集，在合併時還是要對將合併的兩區塊設一個門檻 T_i ，在此我們用的是兩區塊的交集對聯集的比例取門檻，而不是無條件合併的。

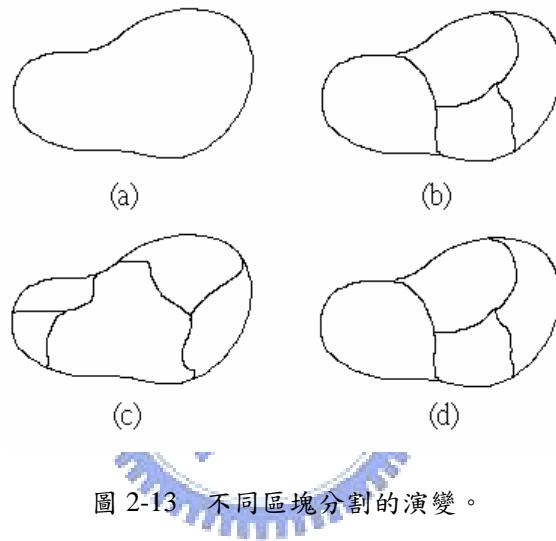


圖 2-14是整合了圖 2-11之後的結果，(a)、(b)各選取了不一樣的門檻值 0.005 及 0.01。與之前的結果相對照，我們可以看到一些光影的影響被消除了。雖然同樣受光影影響的鳥仍未被合併，但是圖 2-11(a)具有物體被區塊涵蓋的範圍較大的優點（藍色圈選處），在之後的追蹤上能提高精準度。而圖 2-11(b)中，對於四隻小豬下方切割較好的部分（紅色圈選處）也被包含在新的結果中。如上述第二種區塊演變情況的球（黃色圈選處），也完整合併成一顆。這些對我們而言，都可以說明整合過後的圖是比較好的。



圖 2-14 整合後之結果。

2.3.4 比較

前節中，我們採用種子放在左上角及中間的兩個經過切割後的結果來整合，如果在比較同時將所有種子放置於同一張圖，如圖 2-15(a)的種子放法（白點表示為種子），經過區塊擴增及合併演算法將得到圖 2-15(b)的結果。我們可以看到，若將種子同時放置於一張圖所得到的區塊會較為細碎，亦即之前曾提到是因區塊過小導致平均值及變異量太敏銳而造成的。這樣細碎的結果也不能說不好，只是我們之後在物體追蹤的應用上，主要的需求是希望能盡可能將區塊合併成一個物體。



圖 2-15 (a) 加倍的種子位置圖，(b) 種子加倍的區塊擴增及合併結果。

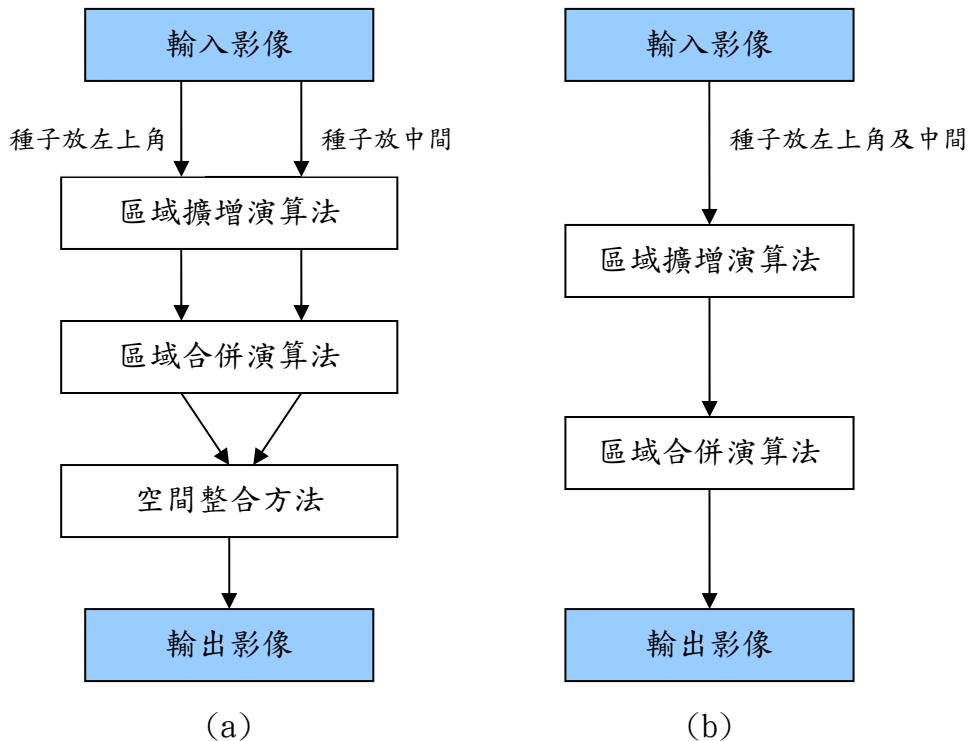


圖 2-16 兩種影像切割流程圖。(a)方法一的流程圖，(b)方法二的流程圖。

圖 2-16(a)(b)是兩種作法的流程圖，而表 2-1 為兩種演算法的執行速度比較。我們可以發現左邊黃色框框的方法一中，需要多做一步整合的處理，但是整合的部分比其他步驟耗費時間極少，即使需要做兩次區塊擴增及合併演算法仍然比方法二快上許多；右邊的方法二主要是在區塊合併花了非常多時間，因為區塊總數會隨著種子數加倍而加倍，在區塊合併的步驟中每一次要比對的區塊數目同樣加倍，而產生出龐大的計算量。

表 2-1 執行時間比較

	區塊擴增	區塊合併	整合	總和
方法一	2*8 sec.	2*17 sec.	0 sec.	50 sec.
方法二	9 sec.	110 sec.		110 sec.

2.4 使用時間回饋 (feedback) 的空間整合方法

在前一節中我們嘗試將兩種種子位置所得到的區塊切割結果做空間上的整合，在圖 2-14 中可以觀察到，整合後的結果確實將不同切割區塊的優點融合在一起。但是如果兩種切割的結果都不是很理想時，空間整合的步驟能夠協助我們來改善嗎？圖 2-17 將種子位置左邊、中間的兩個結果和空間整合後的結果並排表示，並由上而下列了五張連續影像的切割結果。以紅色圈選處舉例，在(l)中，因為(b)、(g)在同一個物體區塊合併後的分界不同，使用空間整合方法中判斷交集的步驟時，因為互有交集而最終能夠合成一個完整的物體。而(m)的結果中最右邊有一小塊屬於同一物體的區塊沒有被包含在一起，而回到(c)和(h)的結果中，可以見到那一塊同樣的沒有被包含在一起。由此我們可以得知，如果用於整合的兩個切割結果不幸在同一個地方出錯時，在空間整合方法的過程中仍然無法克服。

這時候我們的想法還是盡量使得(c)、(h)的切割更好，但是在前面的章節中已經試過許多方法，改善的幅度有限，而前一節的空間整合方法其實也是展現了一些優點，唯其問題是源於整合的來源不佳。另外，我們也希望好的切割能一直延續下去，而不隨著單張切割好壞而影響，因此考慮是否可以在切割第 n 張影像時，融入第 $n-1$ 張可能不錯的結果。據此，圖 2-18 提出一個新作法的流程圖，我們稱之為時間和空間整合方法，在第一張影像時，我們採用由兩張不同位置的種子切割而成的影像經過整合的結果作為最終的輸出，第二張影像開始，我們運用前一張的切割結果和目前的區塊擴增、合併後的結果做整合，以獲得最後的輸出。



圖 2-17 整合不同 seed 位置的影像，(a)-(e)種子位置在左上角的連續五張影像，(f)-(j) 種子位置在中間的連續五張影像，(k)-(o) 連續五張影像經過整合左上角及中間的結果。

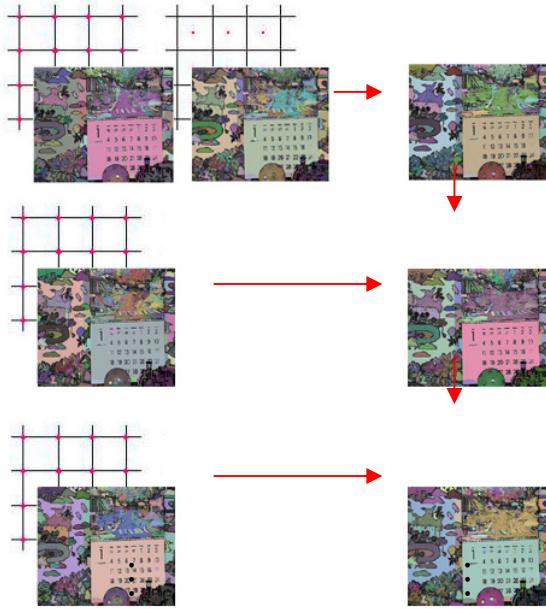


圖 2-18 時間和空間整合方法流程圖。

在時間方向的整合方法上，我們套用2.3.3節空間整合的觀念。如圖 2-19所示，假設第 $n-1$ 張影像對區塊 R_i^{n-1} 切割很完整，在第 n 張時物體移動不大，但區塊切割卻有些零散分成 R_x^n 與 R_y^n 。如果第 $n-1$ 張影像區塊切割很完整的部分會和第 n 張的零散區塊有些重疊，我們便可以透過判斷這些有重疊的區塊使其盡量合併。不同於 2.3.3節整合的部分是，因為比對的兩張圖是不同時間的，物體不見得只有單純的移動，單純考慮交集的情況下如果物體有遮蔽、消失情形時，也會造成合併上的錯誤，因此在時間方向整合的部份我們必須加入區塊在顏色材質上的相似性才能進行合併。

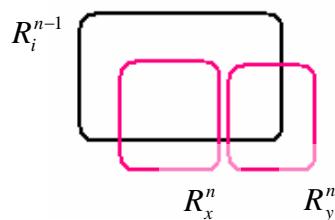


圖 2-19 移動區塊變化示意圖。

反之若是第 $n-1$ 張影像切割的不好，我們可以透過反覆對兩張影像進行合併，如2.3.3節第二種情況，利用第 n 張影像切割較好的特性返回去檢查是否能在擴大區塊的範圍。不過在這裡我們雖然對第 $n-1$ 張影像也有進行區塊合併的動作，卻不回去重新改寫第 $n-1$ 張影像輸出的結果，原因是倒回去改可能要一直追溯到第一張圖，讓每一張圖都合併的更完美，但這樣會造成系統負荷過大。相較之下，如果我們能使得連續影像在追蹤上的結果越來越好，也是一個很好的選擇。

圖 2-20是之前的區塊擴增及合併、空間整合方法和本節之時間和空間整合方法的比較。我們在之前觀察到的紅框中，看到時間和空間整合方法雖然在(l)時合併的不如空間整合方法(g)好，但是到了下一張影像(m)時克服了另兩個方法(c)、(h)切割不好的問題，不僅將前一張的(l)合併好的區塊延續下來，同時也加入(h)將物體左半邊合併在一起的優點，而接下來的(n)與(o)可以看到即使區塊合併演算法輸出的區塊不好、沒有連續，還是能藉由前一張好的合併結果來克服，而使得其結果優於空間整合方法。不過因為時間和空間整合方法中，我們考慮了材質的相似性，在影像色彩亂度較大的地方，在相似性的門檻限制中經常被擋下，而使得區塊不如只考慮交集程度的空間整合方法來的完整，如圖 2-20藍色框起處。這樣的結果是可以接受的，因為我們的目標是希望找到一些有意義及穩定的區塊，因而是可以忽略那些亂度較高的部分。



2.5 總結

在本章中延伸了基本的區塊擴張演算法，我們嘗試以種子位置之選擇來控制其切割的結果，並經由後處理的步驟，使得結果有利於我們之後做物體的追蹤。



圖 2-20 時間和空間整合方法在連續影像上的切割結果比較，(a)-(e) 種子位置在左上角的連續五張影像，(f)-(j) 連續五張影像經過整合左上角及中間的結果，(k)-(o) 連續五張影像經過時間和空間整合方法的結果。

第三章 區塊遞移的追蹤

我們在1.2.2節提到了一些物體追蹤的演算法，大部分的方法只切第一張影像，然後估算物體的移動向量，藉以得到下一張影像中物體的位置。而我們在第二章中運用區塊擴增及區塊合併演算法以及一些延伸的方法將影像中的物體切出，使得每一張圖中的物體位置是已知的，不知道的是兩張圖中物體遞移的關連。本章中我們使用第二章的結果，試著對連續影像上的區塊（物體）做關連對應，希望追蹤區塊在連續影像中移動的過程。

3.1 交集 (intersection)

在一連串的影像中，我們假設物體移動幅度並不大，前後兩張影像的相同物體在位置上必有些許的重疊，雖然重疊的地方是屬於區塊不同的部分，如圖 3-1(a)， R_i 表示第 n 張影像中某個區塊，而 R_j 是第 $n-1$ 張影像中和 R_i 相同物體的區塊， R_i 與 R_j 交集的部分是 R_i 的下半部和 R_j 的上半部，但經由計算交集的步驟，我們可以初步的知道這個物體在下一張影像中可能的位置。

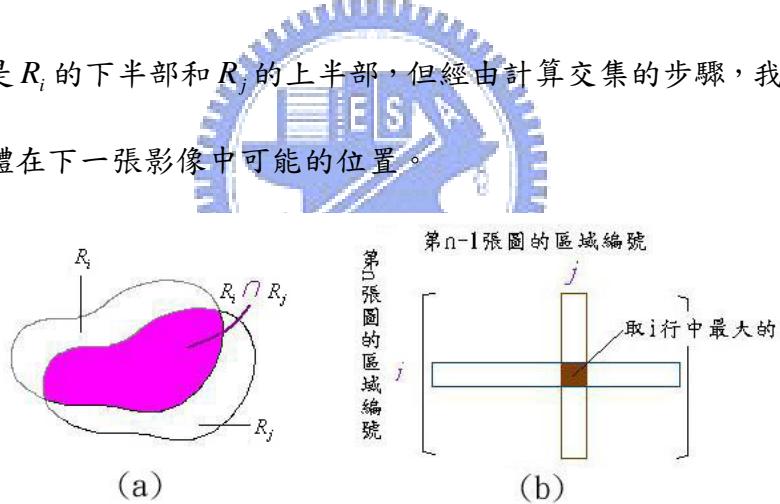


圖 3-1 交集示意圖。

計算交集的過程中，由於每個區塊大小不一，為了使得比較時單位一致，我們計算的是兩個區塊的交集對於聯集 (union) 所佔的比率。假設某個區塊前後影像中的關係如圖 3-1(a)，圖 3-1(b)以一個陣列來儲存前後兩張圖區塊編號的關係，直行的

索引為第 n 張圖的區塊編號，橫列則為第 $n-1$ 張影像的區塊編號，因此陣列第 i 行、第 j 列個元素 ($M[i, j]$) 表示第 $n-1$ 張影像編號為 j ，在第 n 張影像則變成 i 之點的數目，而第 i 行的總和表示第 n 張影像中編號為 i 的總數，也就是 R_i 的大小，對於 j 列也是一樣，所以我們可以用這個陣列將 R_i 和 R_j 的交集聯集定義成：

$$R_i \cap R_j = M[i, j] \quad (3-1)$$

$$R_i \cup R_j = \sum_{k=1}^j M[i, k] + \sum_{k=1}^i M[k, j] - M[i, j] \quad (3-2)$$

前後兩區塊的交集比率為：

$$ratio_{i,j} = \frac{R_i \cap R_j}{R_i \cup R_j} < T_i \quad (3-3)$$

此外為了避免只有微小交集的區塊也被認為是有關連，我們對 $ratio_{i,j}$ 取一個門檻 T_i ，用以決定區塊是否有關連。圖 3-2 左半邊為連續五張影像，以左上角為種子的區塊擴增演算法，經過 2.2 節提到的區塊合併演算法得到的結果。而圖 3-2 右半邊的影像是用左邊的切割結果來計算 3-3 式的 $ratio_{i,j}$ 值，並限定其 $ratio_{i,j}$ 必須大於 0.6 才能定義為是前後有關連區塊。從第二張影像開始有顏色的區塊即為前後有關連的區塊，相同顏色表示互相對應。





圖 3-2 合併之後的影像追蹤。

3.1.1 追蹤的對應關係

在前述方法中的對應方法中，我們是以目前作的第 n 張影像的區塊，去尋找前一張影像中與之交集最大的區塊，因此有可能形成多塊目前的區塊對應到第 $n-1$ 張影像中的相同一塊區塊。當然這樣的情形是取決於我們門檻值 T_i 的大小， T_i 大的時候會使其形成一對一的對應，但同時也會降低成功對應的區塊數目。反之當 T_i 值較小時，提高了成功對應的數目，也會使得第 $n-1$ 張影像對第 n 張影像的關係形成多對一，此時對於亂度較高的區塊追蹤結果看似會不錯，但實際上那些區塊並未合併在一起。

如圖 3-3 是兩種對應方式的比較，(a)是第 n 張影像往前一張影像對應的作法，而(b)則是由第 n 張影像和第 $n-1$ 張影像互相對應的結果。我們可以看到在我們框起來的地方(a)總是比(b)多了一部份的面積，在(a)中看起來是同一區塊的部分，事實上是因為兩者對應到前一張相同的區塊，使得顯示的顏色相同，但我們並未將兩個區域真正的合併。



圖 3-3 追蹤對應方式之比較，(a)單向追蹤，(b)雙向追蹤。

3.1.2 交集的對應結果及討論

圖 3-2 為區塊擴增和合併演算法在切割及追蹤的結果，而圖 3-4 是原圖、空間整合方法以及時間和空間整合方法在考慮交集的追蹤演算法的結果比較。我們首先針對空間整合方法來觀察紅色框圈起處的追蹤狀況，這也是考慮交集的追蹤演



圖 3-4 考慮交集的追蹤方法，原圖(左)，空間整合方法(中)，時間和空間整合方法(右)。

算法經常可能造成的一個問題，圖 3-5為可能遺失追蹤的交集示意圖，(a)到(d)分別表示連續五張的區塊切割結果，(a)到(b)時由兩個區塊變成一個區塊，凡是在我們取追蹤門檻 T_i 大於 0.5 時，(a)中必只有一區會跟(b)的區塊有追蹤上的連結。而(c)的區塊略微縮小，幅度不太所以仍可以追蹤，到了(d)時，又變為兩個小區塊，左邊的對於(c)的區塊交集不夠大於門檻值，而(c)的另一邊也是相同情況，此時這個區塊我們就會追丟了。不管在最先提出的區塊擴增和合併演算法或是空間整合方法，都很容易會發生這樣的情況，因此我們提出希望保留過去好的結果、讓區塊更穩定的時間和空間整合方法。

圖 3-4兩個演算法的比較中，追蹤的區塊數會因為門檻的限制而越來越少，所以我們以第五張圖的區塊觀察，首先注意紅色框起處，在這部份時間和空間整合方法克服了我們在前面提到關於空間整合方法會受到區塊擴增及合併演算法切割好壞的問題，在圖 2-20 中我們也看到時間回饋演算法的切割能夠保留前一張圖的優點，因而在追蹤的時候也能夠避免掉門檻限制而失去追蹤的問題。而在藍色框起處，我們看到時間和空間整合方法在球的部分由於考慮區塊材質相似度時受到光影的影響，切割不是非常完整，使得結果不如空間整合方法來的好，但是其遺失的區塊不多，只要能夠持續的追蹤，我們是可以容忍這樣的不夠完美的。

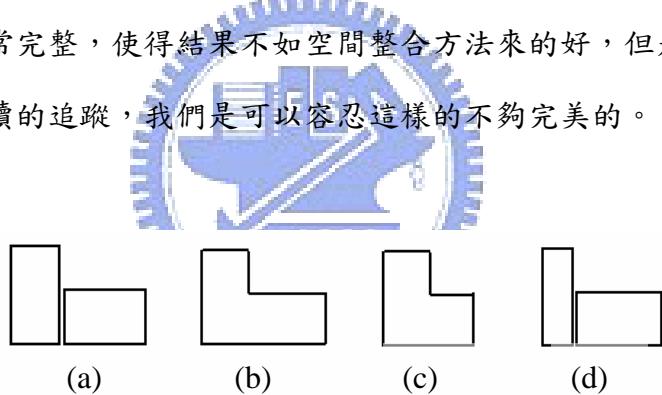


圖 3-5 可能遺失追蹤的交集示意圖。

3.2 區塊相似性

單純以交集程度來追蹤，有時候可能會造成時間方向上對錯區塊的可能，或者如之前提到的因為交集門檻而產生的一些問題，因此我們也可以在比對交集的同時考慮區塊材質的相似性。由於交集的比率和相似性的數值單位上不一致，所以這部

份是承接交集演算法中取一個門檻值 T_i ，在符合 T_i 條件之下，對於所有有交集的區塊，尋找材質相似性最高的一塊區塊來做對應。

圖 3-6分別是原圖，時間和空間整合方法以採用交集追蹤方法、考慮材質與交集的追蹤方法三者比較的結果。我們可觀察到加入比對材質的方法對於顏色相近、亂度較不穩定的部分有較好的追蹤結果。這是由於我們加入區塊材質相似度的比對時，同時降低了對於交集門檻的限制，因為我們只需在交集比率在一個可以接受的範圍，剩下的讓相似度來決定即可，因此在比對區塊材質的追蹤上能有較多數目的區塊成功追蹤。

3.3 重新追蹤

本章前面所述的追蹤方法中，都必須從第一張影像一直不斷的成功追蹤，否則只要有區塊一次沒有成功追蹤，就會永遠遺失這個區塊的追蹤，此外在中途的影像裡新出現的區塊也是無法追蹤的，因此必須額外的處理重新追到的區塊及新出現的區塊。前節提到使用區塊相似度來判斷前後張影像最可能屬於同一物體的區塊，反之，若連續兩張影像中任兩個有交集的區塊 R_i^{n-1} 及 R_j^n 相似度值夠大（我們定義相似度值越小越相像），我們也能夠認為 R_i^{n-1} 及 R_j^n 是不同的區塊，此時從第 n 張影像開始重新對 R_j^n 區塊追蹤。





圖 3-6 由左至右，原始影像(左)、時間和空間整合方法用交集追蹤方法(中)和材質與交集的追蹤方法(右)所得之結果。

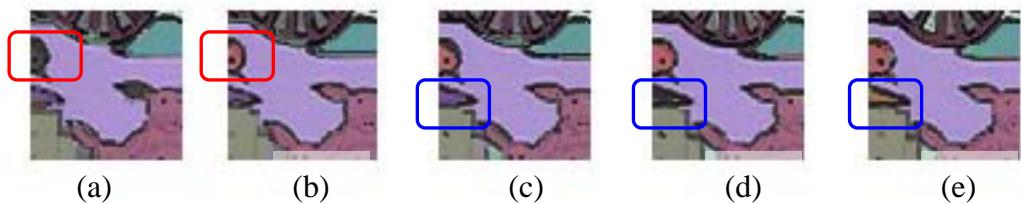


圖 3-7 連續五張影像以相似性判斷新的區塊。

圖 3-7是連續五張影像的部份追蹤結果，(a)中紅色框起處的小鳥頭並未成功追蹤，是因為此時的小鳥頭太靠近邊界且區塊過小造成切割時被忽略。但是(b)中小鳥頭被塗上橘色，顯示在(a)中與之有交集的區塊相似度值過大，而被認定為新出現的區塊。(c)的藍色框中顯示一個原本成功追蹤的區塊，在(d)中也是由於區塊擴增演算法中的種子無法落於區塊內造成的切割不佳，使得無法成功追蹤，(e)中順利的切出區塊重新追蹤成功。

3.4 總結

在本章中我們使用第二章的切割結果來做區塊的追蹤，我們在第二章盡可能的使切割結果完整及穩定，加上我們假設影像中物體移動量不大，因而可以使用比較直觀的方式達到追蹤的目的，而不需估算複雜的移動向量。



第四章 實驗結果與討論

前面章節介紹我們所使用的一些切割及追蹤的方法，本章中我們對結果做量化的分析，探究各方法的優缺及其遺失追蹤或是錯誤追蹤的原因。

4.1 成功追蹤的區塊分析

首先，我們觀察在一連串影像中的成功追蹤結果，連續 40 張的影像區塊的大小、形狀會增減，也有新出現的區塊，但在這裡我們先不考慮新進影像的區塊，因為在第三章中提出以相似性較大作為判別新出現區塊的方法只是大略的概念，穩定性不足，如果加進來分析，恐怕會擾亂結果。此外，我們希望分析那些最穩定的結果，所以在追蹤的時候，除了以目前的第 n 張影像對第 $n-1$ 張影像找交集最大或是最相似的，反過來也用第 $n-1$ 張影像對第 n 張影像找交集最大或是最相似的，如果兩者彼此對應才被我們認為是穩定的追蹤結果，換言之就是要求連續影像的區塊對應關係必須一對一。

圖 4-1 是三種區塊切割方法和兩種追蹤方式對於成功追蹤的區塊數之比較，圖 4-2 則是三種區塊切割方法和兩種追蹤方式對於成功追蹤的區塊面積之比較。首先我們可以看到區塊擴增及合併演算法在一開始成功追蹤的區塊很多，但是後面急遽的下降，尤其是考慮相似度的追蹤下降的更快。圖 4-3 是區塊擴增及合併演算法在第 10、20、30 和 40 張影像中，一對一做交集及相似度追蹤的結果，有顏色的區塊即是成功追蹤的部分。我們可以看到以相似度做追蹤能夠讓切割不是非常穩定但相似性高的區塊保留下來（如紅色框起處），但同時也使得有些看似很穩定的區塊追蹤失敗（如藍色框起處），其原因是來自於切割時所產生的一些破碎的區塊，使得在計算相似度上受到干擾而遺失追蹤。此外細碎的切割也是有一定的機率使得區塊達不到交集的門檻而失蹤，因此切割的完整性對於我們的追蹤是很重要的。

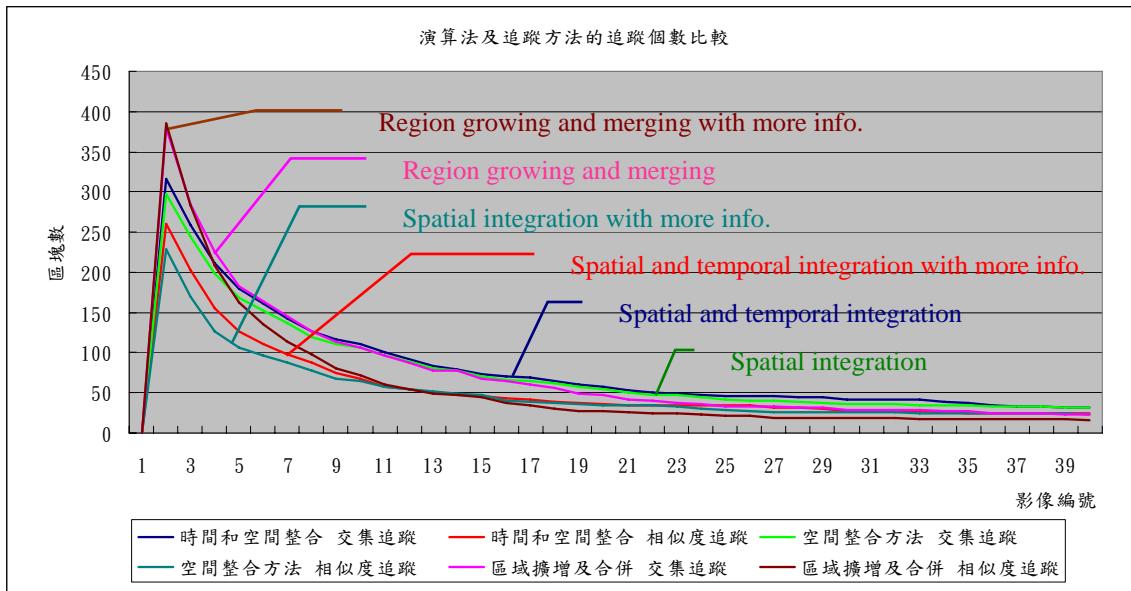


圖 4-1 各演算法成功追蹤區塊數之變化。

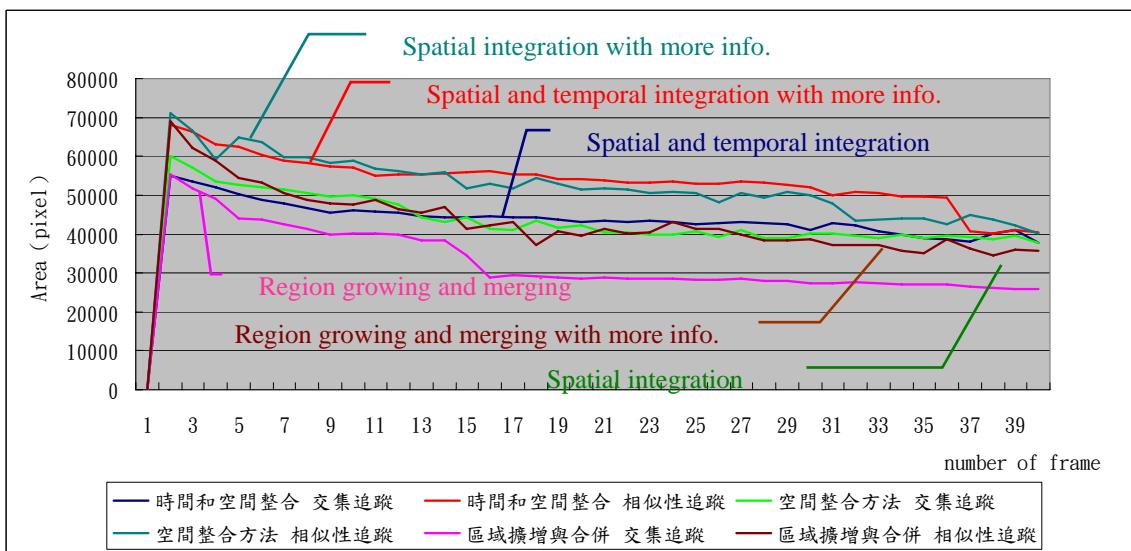


圖 4-2 各演算法成功追蹤面積之變化。

圖 4-4 是空間整合方法在第 10、20、30 和 40 張影像中，一對一做交集及相似度追蹤對應的結果。觀察空間整合方法的追蹤區塊數目及面積，可以發現由於空間整合方法的切割結果相對於區塊擴增及合併方法而言，區塊變的比較完整，因而在追蹤的時候相對應的區塊數會比較少，追蹤到的面積卻比較大。不過隨著區塊不斷遞移，這兩種方法的區塊數目也漸漸趨近一樣，因為空間整合方法受到區塊擴增及合併方法的影響，對於區塊的穩定性不足，在變動頻繁的狀況下容易追蹤失敗。也



圖 4-3 原圖(左)、區塊擴增與合併之結果用交集追蹤方法(中)，和利用相似度與交集門檻的追蹤方法(右)，所得之結果。

由於某些區塊大小變動較大，採用交集的追蹤法常會因此而追蹤失敗，但採用相似性的追蹤則能夠一直追著最相似的區塊而持續找到，如圖 4-4月曆上方藍色圈起處。反之，有些區塊雖然很穩定，但是新進入影像的顏色有一些差異，最後超出相似度追蹤容許的範圍，而導致我們追蹤失敗，如圖 4-4影像左上角紅色圈起處。因而兩種追蹤方式可以說各有優缺。然而這兩種追蹤方式都有追錯區塊的風險，交集追蹤可能因物體小又移動的很快，使得不同物體的區塊有大部分是交集的，相似性

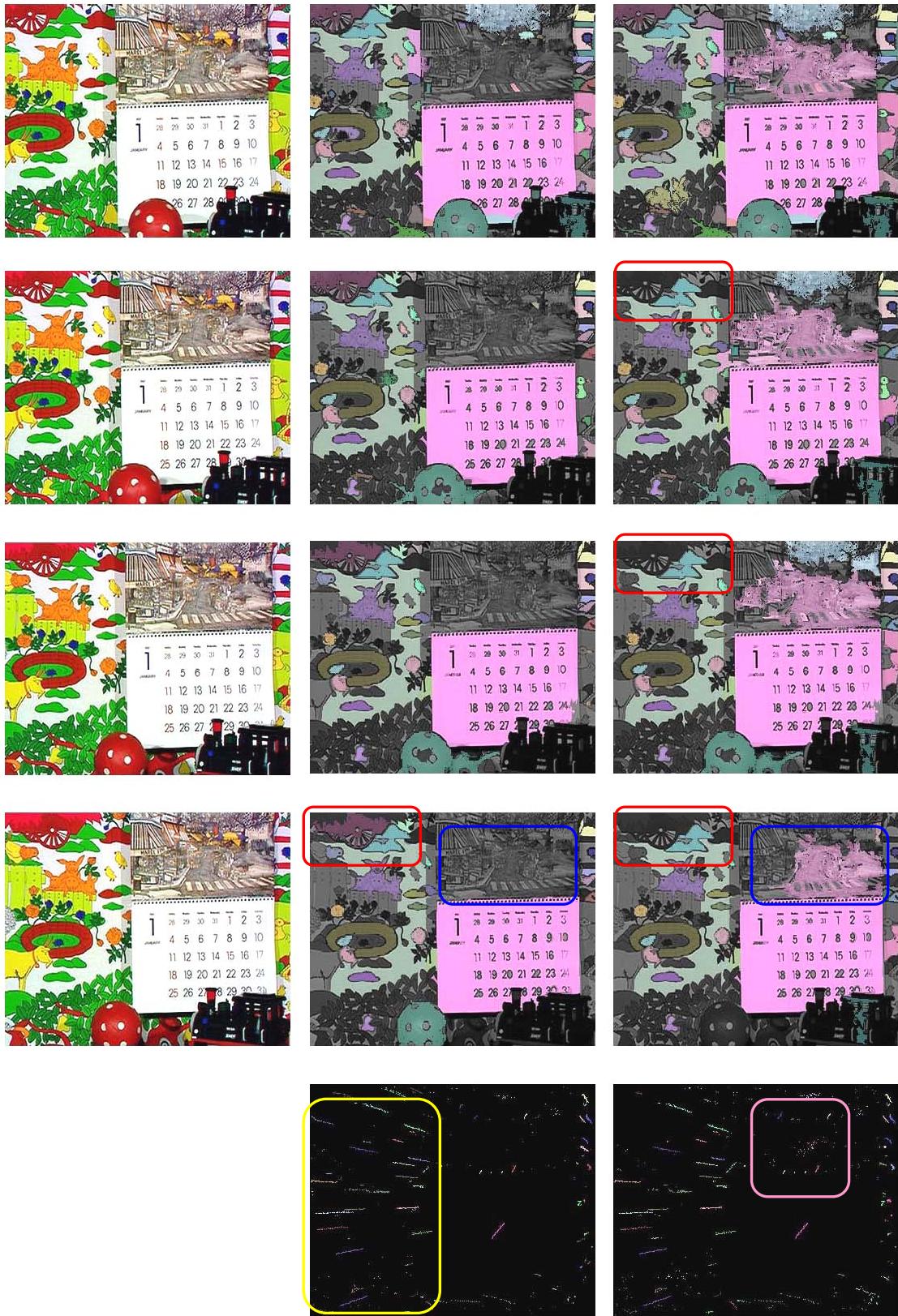


圖 4-4 原圖(左)、空間整合方法用交集追蹤方法(中)和材質與交集的追蹤方法(右)，所得之結果。

追蹤的錯誤，如圖 4-5(a)到(c)，由於(a)將球和旁邊相同顏色的區塊合在一起，而(b)的切割將球和旁邊的區塊分開來，經過相似性的比對，認為對應的是球旁邊的區塊，(c)中區塊切割的結果不佳，使得沒有任何一個區塊符合相似性追蹤的條件，因而追蹤失敗。

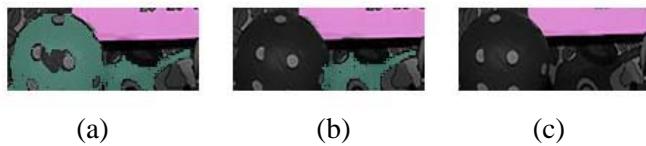


圖 4-5 相似度追蹤錯誤的例子。

圖 4-4(i)、(n)是將 40 張影像中有追蹤到之區塊的重心畫出來，同一區塊的點(同顏色的點)越多代表其追蹤到的次數越多，左半邊出現了許多條線，因為物體顏色鮮明、多數形狀很穩定，使得成功追蹤的機率高，反觀(n)中粉紅色圈起處，因為區塊形狀經常變動很大，因而看不出其運動的規律性。從這張重心的軌跡圖中，我們也可以稍微看出物體的移動情形，且其軌跡大致上符合了真實區塊的移動方向，說明了能夠以我們的方法計算出區塊遞移的大致情況。

最後是時間和空間整合方法的追蹤，圖 4-6是時間和空間整合方法在第 10、20、30 和 40 張影像中，一對一做交集及相似度追蹤對應的結果。雖然時間和空間整合方法在影像中球體合的切割因為受到光影影響，而不如在合併時只考慮交集的空間整合方法來的完整，使得採用交集追蹤時會追不到，但是在相似度追蹤卻能克服這個問題。

從圖 4-1成功追蹤的區塊改變曲線來看時間和空間整合方法，我們追到的區塊是逐漸在減少的，甚至一開始就比區塊擴增及合併演算法方法少，減少的也比空間整合方法稍快，但是在圖 4-2的追蹤面積改變曲線中，可以看到時間和空間整合方法比另兩種方法都來的穩定，這是由於時間和空間整合方法提供讓前一張影像好的切割能保留下來的機會，減少的區塊數目可能只是合併到其他的區塊中而不是消失。時間回饋演算法能保留過去切割好的部分，因此降低了小區塊出現的機率，也降低了用相似度追蹤錯誤的機率，故時間和空間整合方法在相似度追蹤上的表現能

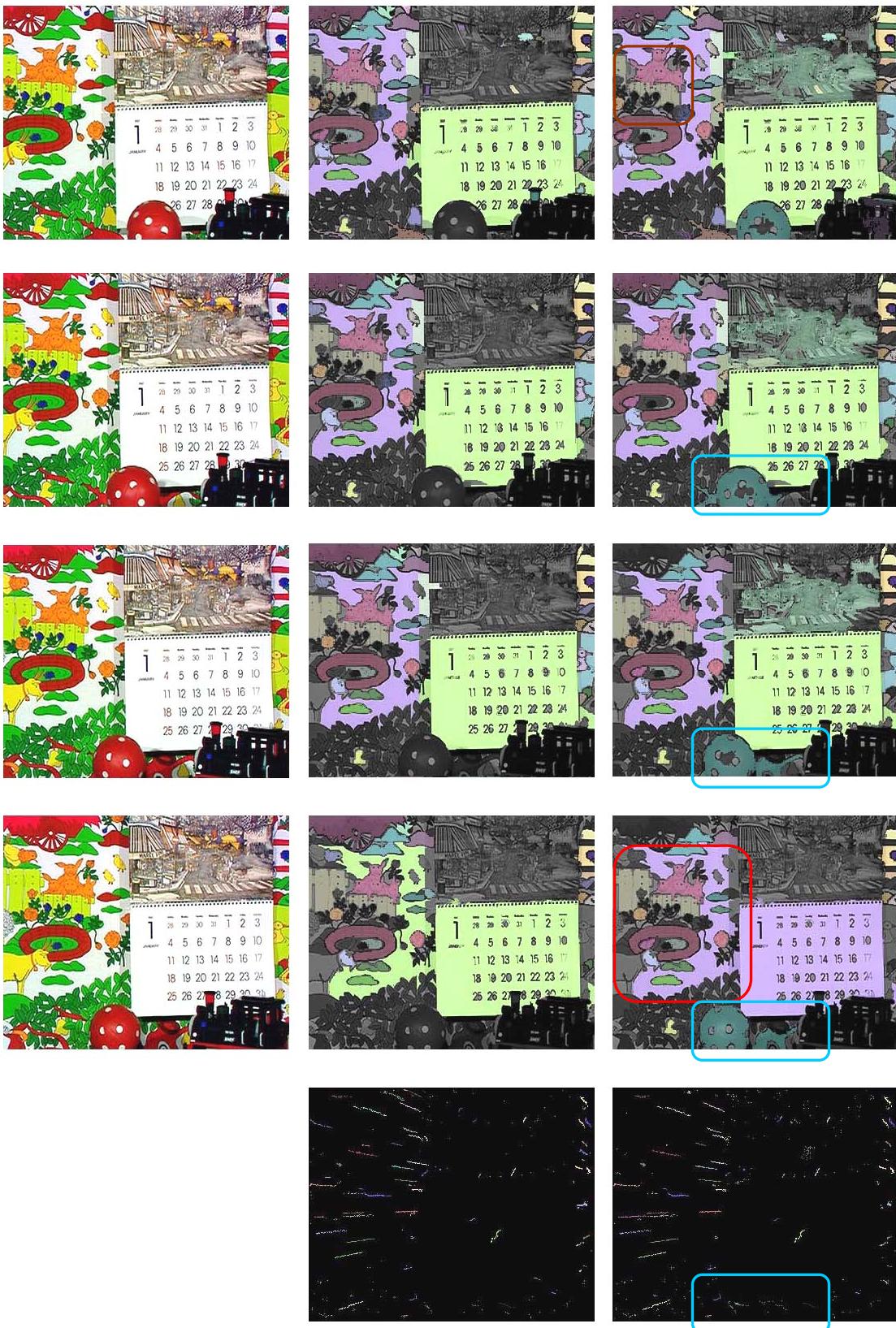


圖 4-6 原圖(左)、時間和空間整合方法用交集追蹤方法(中)和材質與交集的追蹤方法(右)，所得之結果。

比交集追蹤來的更好，因而我們可以發現一點，對於切割較穩定的區塊使用相似性追蹤能有很好的成效，而大小不穩定的區塊使用交集追蹤較能持續的將區塊保留。

4.2 區塊穩定性

在上一節中我們從成功追蹤的區塊數目及面積觀察演算法的特性及穩定性，在這些結果中有許多區塊一直持續的被追蹤，接下來我們想瞭解這些持續追蹤到的區塊面積是否穩定。

圖 4-7 是追蹤影像左上方四隻小豬的面積變化圖，因為限定一對一對應的追蹤方式，區塊面積大小與追蹤方式無關，因此只需比較三種演算法的面積消長。在連續影像上四隻小豬的區塊沒有形變，只有往右移動並稍微變小，因而我們理想中的區塊應是漸漸穩定的遞減。面積變化曲線圖中，我們看到區塊擴增及合併方法晃動極大，相較於另兩種方法面積也較小，如圖 4-8(a)，可知此種方法區塊穩定性不足，也經常有漏掉一些應屬於區塊中的點沒有被涵蓋到。空間整合方法及時間和空間整合方法在此區塊的表現上大同小異，只有在第 33 張影像時空間整合方法遺失了四隻小豬的上面那隻，如圖 4-8(b)，而使得區塊面積大幅度下降，雖然這樣的情況不多見，但我們可以認為時間和空間整合方法可以避免掉這樣的問題而顯得較穩定。

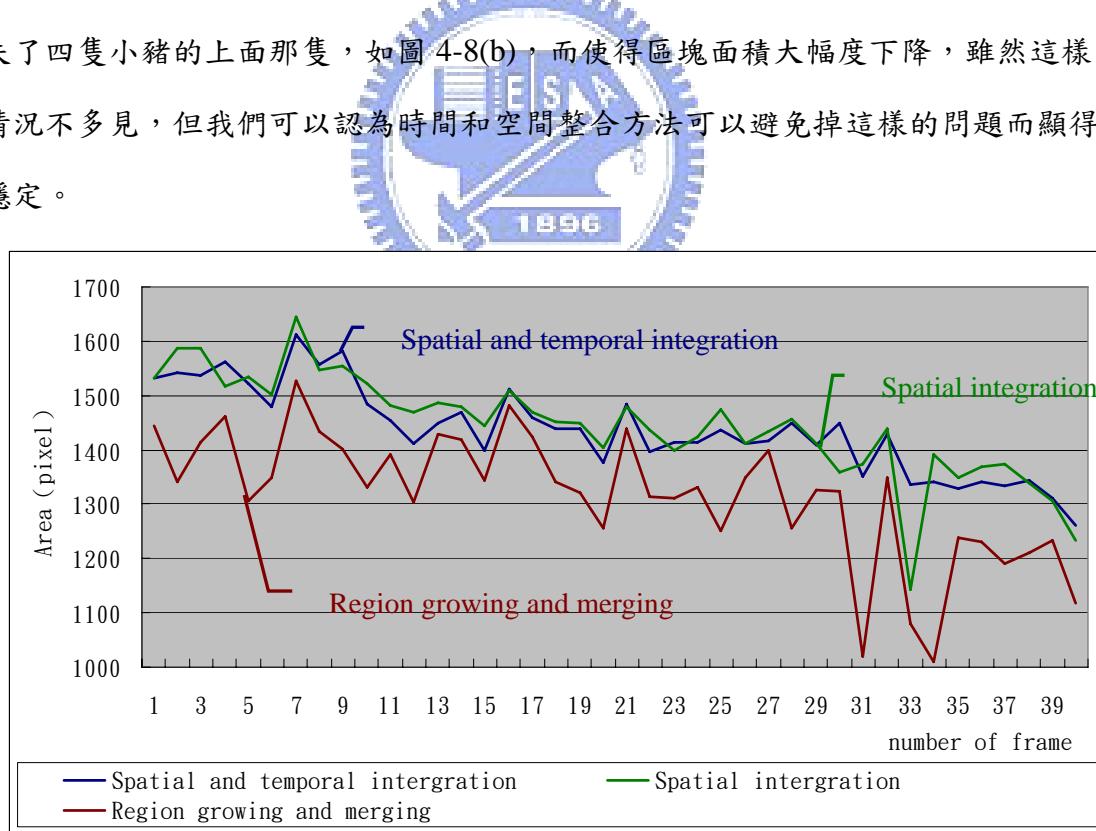


圖 4-7 單一區塊追蹤面積變化圖。

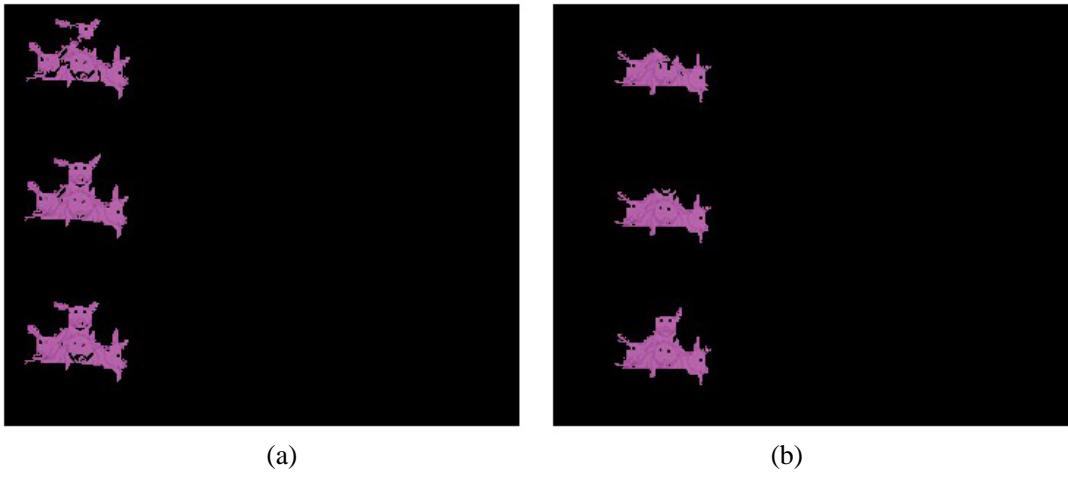


圖 4-8 單一區塊追蹤狀況，(a)第 1 張影像，(b)第 33 張影像。

4.3 分水嶺演算法

分水嶺演算法和區塊擴增演算法極為相似，在1.2.1.2曾經提到過分水嶺演算法是由區塊最小值的點開始向周圍延伸，當碰到來自於其他區塊的點時，即認為那一點為邊界。在本節中我們嘗試使用分水嶺演算法來取代之前的區塊擴增演算法，觀察其是否能得到類似區塊擴增演算法之結果。

4.3.1 分水嶺演算法的變形

我們採用F. Meyer and S. Beucher[2]提出的分水嶺演算法，不過我們如同區塊擴增演算法一般對影像放入格子狀分佈的種子，以這些種子作為區塊最小值向周圍擴增，避免找區塊最小值所造成的複雜度。圖 4-9是分水嶺演算法的切割結果，白色的部分代表區塊的邊界，我們可以觀察到其中有許多細碎的小區塊，想必是由於種子放置的位置可能是區塊中較高的色彩高度，因而無法向周圍擴增所造成。並且我們也發現有許多區塊跨在物體的邊界上，如紅色框中所示，這將會使我們合併之後的結果無法理想的逼近物體輪廓。



圖 4-9 分水嶺演算法初步切割結果。

4.3.2 分水嶺演算法的合併

我們希望得到的結果仍是一個穩定且盡可能靠近物體邊緣的結果，因此我們將分水嶺演算法的結果一樣做區塊合併，得到的結果如圖 4-10 所示，(a) 與 (b) 是連續的兩張影像。相較於區塊擴增演算法，分水嶺演算法有較多細碎的區塊，且有許多區塊是跨在物體邊界上的，也由於區塊有跨邊界的疑慮，使得合併時同屬一物體的區塊顏色平均值及變異量連帶受到影響，導致合併的過程並不順利。再觀察前後兩



圖 4-10 分水嶺演算法經過區塊合併的結果。

張影像間區塊的變動性，我們可以發現區塊的形狀、大小都改變很多，這樣是無法有效追蹤的，如果利用我們前面提到的整合方法，又因為無法確保一個區塊必定落於一個物體內，而空間整合方法只判斷交集的特性若應用至此，恐怕會造成區塊合

併上的錯誤，例如將不屬於同一物體的區塊合併在一起，因而分水嶺演算法觀察至此，可以確定這樣的分割方式不適用於我們的方法。

要提高分水嶺演算法的切割精準度，我們可以回歸原始的作法嘗試在做分水嶺演算法之前先對影像做前處理，算出其梯度影像，如此一來能較精準的切在物體的邊緣上，只不過梯度值的算法在彩色影像上並沒有定義，加上即使精準的切在邊界上，仍避免不了我們在2.1.1節中提到的因邊界色彩影像物體色彩的缺點。

4.4 區域擴增演算法的改進

在2.1.1節中我們在區域擴增時要求相似度限制，使得有些點會部屬於任何區域，如圖 4-11紅色框中黑色的部分，這是由於我們一開始給予的種子間距不夠密集，而使得沒有任何一個種子落於這些區域內，因此要是希望切到這些區域就得給予更小的種子間距。但是我們無法預期每一張影像需要多小的種子間距才夠，而在物體很大的部分（如月曆的部分）給予太小的種子間距只會造成區塊數增加，而使得區塊合併時的運算量增加。因此我們希望一開始給予較大的種子間距，再將未被標示區域的面積比例來動態調整種子的間距。



圖 4-11 有限制的區域擴增結果。

如圖 4-12(a)所示，一開始的種子間距較大，在進行區域擴增演算法之後，假設所獲得的區域如圖 4-12(b)，此時格子中有很多地方是不屬於任何區塊的，因此我們

再將格子分成四塊，然後對每一個格子去計算已被區塊涵蓋的面積比例，如圖 4-12(c)，我們對面積比例取一個門檻，如果未超過門檻值則在格子點的中央再放置新的種子，但是如果中央的那一點已屬於某個區塊則不放置種子。當 $1/4$ 格子大小的種子放置完畢之後，我們再次做區域擴增演算法，但是不重新判斷那些在上一輪已經加入別的區塊的點，以此加快執行的速度。之後我們重複將格子切得更小，如圖 4-12(d)，直到格子小於我們可以接受的程度。

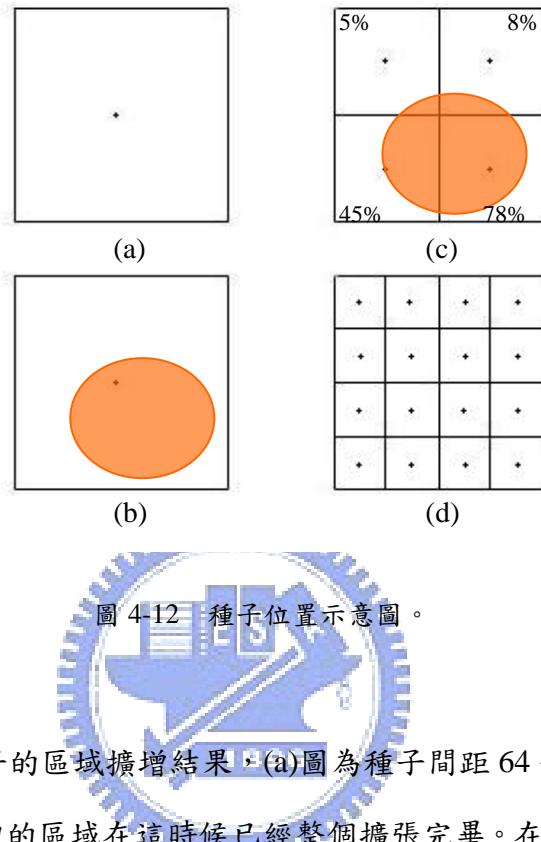


圖 4-12 種子位置示意圖。

圖 4-13 是動態種子的區域擴增結果，(a)圖為種子間距 64 個點 (pixel)，我們可以看到比較大塊而平坦的區域在這時候已經整個擴張完畢。在(b)中將種子間距降為 32 個點，由於我們並未更動(a)中區域擴增的結果，因此我們還可以在(b)中看到(a)的那些區塊，至於(c)和(d)分別是再將種子間距降為 16 和 8 個點。最小的種子間距和之前一樣，但是和之前的結果相比，如圖 4-13(e)，由於動態增加種子的機制使得區塊不再細碎，甚至和舊方法進行區塊合併之後的結果相比（圖 4-13(f)），也並未遜色太多，因而我們可以省略合併的步驟，直接利用時間上的整合方法達到差不多的追蹤結果，如圖 4-14。

雖然動態給予種子的區域擴增演算法能夠省略區塊合併的運算量，卻因為大面積的區域擴增時造成階層數很大，亦即離種子很遠的點需要走很多步才能加入，而使得區域擴增的運算量變大，損益相抵使得新的方法並未比舊的方法快上多少，未來也許能夠改進區域擴增演算法來提高運算的速度，也許時間上就能超越舊的方法。



圖 4-13 動態給予種子的區域擴增結果。



圖 4-14 原圖(左)、時間和空間整合方法用相似度追蹤方法(中)和動態給予種子的區域擴增與時間整合以相似度追蹤方法(右)，所得之結果。

4.5 總結

在本章中我們試著對三種區塊切割方法和兩種追蹤方式所做出來的結果進行量化分析，試圖從中找到我們認為時間和空間整合方法能夠穩定追蹤的事實。此外，我們嘗試了其他的切割方法，第一種是將區塊擴增演算法改為相似的分水嶺演算法，第二種則是改變區域擴增演算法給種子的方式，並分析其中的差異及優缺。期待能找到更適合我們應用的切割方法。

第五章 結論與未來工作

在本篇論文當中，我們提出了一個不需計算移動向量而能持續追蹤區塊遞移的方法，我們延伸基本的區塊擴增演算法，透過控制種子的位置及整合不同種子位置的切割結果來達到單張影像的最佳化，為了在時間方向上延續較好的切割結果，我們採用時間回饋（feedback）的方式，將前一張的區塊切割結果與目前的區塊擴增及合併結果做整合，使得區塊的合併不易受到區塊擴增演算法結果的優劣干擾。而在追蹤的部分，我們採取較為直觀的判斷方式，利用連續影像物體變動不大的特性，以區塊在位置上的涵蓋的比例，找出可能的同物體區塊。再透過區塊在色彩上的特徵判斷出相對應的區塊。

我們嘗試使用動態給予種子的方式來增進切割的效率，但卻因為區域擴增演算法的限制使得效率並未大幅增進，若能提出更好的區域擴增運算架構，就能將我們系統的複雜度大幅提昇。此外，在我們提出的方法中，是對整張影像的區塊進行追蹤，放置種子的位置也是單純的取格子點左上角或是中心，在接下來的工作中，可以嘗試在一開始定義出一些特定的區塊，使得之後種子的位置能隨之更動，降低遺失區塊的可能性。

我們也提出了一個判斷新出現區塊的方法，但是這樣的判斷並不是絕對正確，經常將一些因為區塊切割不穩定而為追蹤到的區塊認為是新的區塊，若是這些曾經遺失追蹤的區塊找回來之後還能持續追蹤一段時間也是不錯，只是這些變動很大的區塊經常只能出現在一兩張影像中。因此在之後的工作中，如何能正確的判斷出新出現的區塊也是很重要的課題之一。

參考文獻

- [1] R. Adams and L. Bischof, “Seeded region growing,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 6, pp. 641–647, 1994.
- [2] F. Meyer and S. Beucher, “Morphological segmentation,” *J. Vis. Commun. Image Rep.*, vol. 1, no. 1, pp. 21–46, 1990.
- [3] F. Meyer, “Color image segmentation,” in *Proc. IEE Int. Conf. Image Processing and its Applications*, The Netherlands, pp. 303–306, 1992.
- [4] J. M. Gauch, “Image segmentation and analysis via multiscale gradient watershed hierarchies,” *IEEE Trans. Image Processing*, vol. 8, pp. 69–79, 1999.
- [5] H. Gao, W.-C. Siu, and C.-H. Hou, “Improved techniques for automatic image segmentation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, 2001.
- [6] S. E. Hernandez, K. E. Barner, and Y. Yuan, “Region merging using homogeneity and edge integrity for watershed-based image segmentation,” *Optical Engineering*, vol. 44, no. 1, 2005.
- [7] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, “Hybrid image segmentation using watersheds and fast region merging,” *IEEE Trans. Image Processing*, vol. 7, no. 12, 1998.
- [8] D. Koller, K. Daniilidis, and H. Nagel, “Model-based object tracking in monocular image sequences of road traffic scenes,” *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, 1993.
- [9] D. Wang, “Unsupervised video segmentation based on watersheds and temporal tracking,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 525–538, 1998.
- [10] N. Peterfreund, “Robust tracking of position and velocity with Kalman snakes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 6, pp. 564–569, 1998.
- [11] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, “A real-time computer vision system for measuring traffic parameters,” in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 495–501, 1997.
- [12] A. Cavallaro, O. Steiger and T. Ebrahimi, “Tracking video objects in cluttered background” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 4, 2005.