

# 國立交通大學

資訊科學系

碩士論文

以建構背景擷取運動物件之方法  
Moving Object Segmented by Background Construction



研究生：黃偉城

指導教授：陳玲慧 教授

中華民國九十三年十一月

# 以建構背景擷取運動物件之方法

研究生:黃偉城

指導教授:陳玲慧

國立交通大學資訊科學研究所

## 摘要

以物件為基礎的影片壓縮方法當中，壓縮的效率跟運動物件的擷取是成正比的，若是運動物件的擷取越準確，則其壓縮的效率越好。在此篇論文當中，我們分別對時間域跟空間域做分割，再將兩種結果結合。在空間上我們利用分水嶺演算法，將空間上的特徵資訊找出來。在時間方面，我們將連續畫面相減，然後以自動產生的門檻值判別影像是否有改變，利用此項資訊，建構一個可靠的背景，當一個像素連續多個畫面被判定為沒有改變的像素，則此像素是背景像素，我們便將它建構成背景。最後，將目前的畫面跟背景影像比較而定義出移動物件遮罩。此遮罩再與分水嶺演算法的結果整合即可準確的將物件切割出來。

# Moving Object Segmented by Background Construction

Student: Wei-Cheng Huang

Advisor: Dr.Ling-Hwei Cheng

Institute of Computer and Information Science

Nation Chiao Tung University



In the method of the object-base compression video, the compression performance and the motion object segmented are in the direct proportion. If the motion object segmented is more accurate, the compression performance is better. In this paper, we segment the object on the temporal domain and the spatial domain, and then we combine the results of them. In spatial domain, we use the watershed algorithm to find out the characteristics of the spatial domain. In the temporal domain, we subtract the constant frames, and then we determine the frames by the threshold of auto generated if change or not. We make use of this information to construct a reliable background. If the constant frames of pixel are categorized to an unchangeable pixel, the pixel is background pixel. We will construct it to the background. Finally, we compare the current frame with the background to define the motion object mask. The combination of the object mask and the result of Watershed Algorithms is able to segment the objects exactly.

## 誌謝

首先我要感謝指導教授陳玲慧老師，她耐心以及細心的指導讓我深深感動，而老師並非只有在學業方面給予指導，她還提醒我，教導我許多做人處事的態度，能得到老師如此般的照顧以及關心，我真的很感激，能有緣遇到陳老師真的是很幸運的事，所以我真的非常感激她。

接著我要感謝的是實驗室裡面的所有成員，尤其是博士班的學長也給了我很多的幫助，學業上的疑惑或是一些生活上的問題，若是請教他們，他們都很熱心的協助我。而跟我一起的同學以及學弟們，大家也會互相的幫助，互相的勉勵，是我更有衝勁。

再來我要感謝我的女友，她給予我的鼓勵是不間斷的，她對我的肯定以及對我的支持也是我很重要的支柱，再我遇到挫折時總會因為她的鼓勵而在度爬起來，所以我真的很感激她對我所做的一切。

最後我要感謝我的父母以及親人，他們對我的關心與愛護也是無條件的付出，不管多辛苦，他們只希望我能有更好的前途，這也是我努力的動力來源之一，所以我真的很感謝你們，僅將這篇論文獻給我親愛的家人。

# 目錄

摘要 .....	i
Abstract .....	ii
誌謝 .....	iii
目錄 .....	iv
LIST OF TABLE .....	vi
LIST OF FIGURES .....	vii
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 相關研究討論.....	2
1.3 論文架構.....	4
第二章 空間域的切割.....	6
2.1 Morphological Gradient.....	6
2.2 分水嶺切割(Watershed segmentation).....	11
2.2.1 用氾濫(immersion)模擬的分水嶺演算.....	11
2.2.2 分水嶺演算法之實現.....	12
第三章 時間域的切割.....	15
3.1 概要.....	15
3.2 演算法的整個流程.....	16

3.2.1 畫面差(Frame Difference).....	17
3.2.2 背景的建構(Background Construction).....	19
3.2.3 背景-目前畫面差(Background-Current Frame Difference).....	21
3.2.4 偵測物件(object detection).....	23
3.2.5 後續的處理(post process).....	27
3.3 門檻值的自動產生.....	29
第四章 空間-時間域切割及實驗結果.....	37
4.1 結合時間域跟空間域的切割.....	37
4.1.1 移動區域的偵測.....	37
4.1.2 移動邊界的偵測.....	40
4.2 實驗結果.....	41
第五章 結論.....	48
REFERENCES.....	49

## LIST OF TABLES

表 3.1 物件偵測判斷表.....	23
--------------------	----



## LIST OF FIGURES

圖 2.1 Dilation 和 Erosion 之示意圖.....	8
圖 2.2 梯度運算的結果.....	10
圖 2.3 分水嶺示意圖.....	11
圖 2.4 分水嶺演算法步驟示意圖.....	13
圖 2.4 分水嶺之結果.....	14
(a)左邊為 akiyo 分水嶺之結果右邊為原圖	
(b)左邊為 hall 分水嶺之結果右邊為原圖	
圖 3.1 時間域切割之流程圖.....	17
圖 3.2 FDM.....	18
(a)Akiyo 的 FDM	
(b)hall 的 FDM	
圖 3.3 建構出來的背景資訊.....	21
圖 3.4 hall 的 BCDM.....	22
圖 3.5 物件些微改變在畫面差中的影響.....	25
圖 3.6 示意圖.....	26
圖 3.7 經過消除雜訊後的物件遮罩.....	28
圖 3.8 自動產生門檻值流程圖.....	30
圖 3.9 期望值為 0 之高斯分佈.....	31
圖 3.10 Akiyo 之直方圖以及 FDM.....	33



圖 3.11 hall 之影像差直方圖.....	35
圖 3.12 hall 影片不同門檻值之比較.....	36
圖 4.1 合併時間空間域切割之流程圖.....	38
圖 4.2 移動區域偵測之結果.....	39
圖 4.3 偵測移動邊界的結果.....	41
圖 4.4 Akiyo 的分割結果.....	42
圖 4.5 news 的分割結果.....	43
圖 4.6 claire 的分割結果.....	44
圖 4.7 hall 的分割結果.....	46
圖 4.9 mother 的分割結果.....	47



# 第一章

## 緒論

### 1.1 研究動機

隨著影音多媒體快速的發展，多媒體在網路上的傳輸越來越頻繁，再加上未來可能普及化的數位電視，甚至在手機上影音多媒體的收發，使得視訊的壓縮佔著相當重要的地位，一個好的壓縮標準是不可或缺的。自 1990 年開始制定的 MPGE-1 標準，到後來的 MPEG-2 標準，都提供了不錯的壓縮效率，目前也被廣泛的運用。在這些標準當中，都是使用以區塊為主的編碼(block-base coding)方式，也就是把每張影像切成一個一個的方塊(block)，每個方塊各自根據本身的內容來壓縮，但是這樣的壓縮效益已經不能滿足某些需求了。這時，以物件為基礎的壓縮(object-base coding)逐漸受到重視，而在 1998 年 10 月制定 MPEG-4 標準[1、2]就是一個以物件為基礎的壓縮標準，它是針對數位電視、互動式繪圖應用(影音合成內容)、互動式多媒體之整合技術訂定的國際標準。在 MPEG-4 中，視訊資料的內容被分割成數個 VOP(video object planes)，每個 VOP 代表了畫面當中的一個物件的內容。這些 VOP 可以被分開編碼、儲存及傳送，而接收端可依照需求及應用將這些 VOP 加以重組、刪減或是替換。

以物件為基礎的壓縮方式是將影像序列當中的動態物件(motion object)從背景中分離出來，並分別壓縮，這些物件也就是 MPEG-4 中的 VOP。因為前景

即為物件，也就是會改變的部份，所以將物件分離壓縮相當於只須對畫面中產生改變的地方做壓縮，而不動背景的部份不需要和傳統的壓縮標準一樣，在壓縮每一個畫面(frame)時都將該畫面的背景一起壓縮。物件為基礎的壓縮則會將背景建構出來，然後對整個背景做壓縮，之後只需要用較少的資訊來維護背景的資訊。這是提升壓縮效益的一個重要因素。所以在物件為基礎的壓縮當中，物件的擷取變成一個重要的課題，在標準當中並沒有提供物件切割的方法，而物件擷取的好壞也會直接影響到壓縮的效率，越是能夠精確的擷取出動態的物件，則能夠越清楚的將畫面中改變的像素和沒有改變的像素區分出來，然後對改變的像素做壓縮，壓縮的效益也就越高，但是若將不會變動的背景也當做事務見擷取出來，則對這些背景部份所做的壓縮是多餘的。因此本論文當中，我們將討論如何使用影像序列當中影像的內容，來切割出動態的物件。

## 1.2 相關研究探討

動態影像切割技術的發展已經有相當的時間，經過這麼久的發展已經存在了相當多的方法。在切割的時候，我們擁有兩種資訊可以運用，一個是利用空間域上的資訊，另一個就是同一物件在時間  $t$  跟  $t+1$  因為運動而在這兩張影像上面產生了某種關聯性，這是所謂時間軸的資訊。所以根據這兩種資訊，可以幫助我們將動態的物件擷取出來。

空間域影像切割是對於靜態的影像做切割，也就是影片中的一個畫面

(frame)。它切割出來的結果，必須盡量跟人類視覺上的看法相同。而大部份的方法是將影像中空間的特徵切割出來，例如：邊、顏色相似區域或是紋理相似的區域等等。

由 Comaniciu 所提出來的，對空間以 Mean shift 來做分割的方法 [3、4]。它會將整個影像分割成多個區域，每個區域內的像素具有相似色彩或是相似灰階強度。Mean shift 演算法是在一個空間域跟色彩域結合的象限中做處理，我們稱之為空間-強度域，空間域是影像中的二維座標(x,y)，色彩域則是像素的色彩資訊，在灰階影像中為像素的灰階值強度則維度 I，彩色影像則為 R，G，B 共三維。所以每個像素可以看成三維(x,y,I)或五維(x,y,R,G,B)的向量。

當我們最某個像素 p 做 mean shift 計算時，我們以 p 為中心點，以維度中限定一個範圍 S，此範圍 S 是必須事先給定。例如，在三維的灰階影像，我們給定一個 S 其範圍如下：“和 p 的距離為 10 個像素且和 p 的灰階值強度差值在 5 以內”，所以若某像素和中心像素的距離小於 10 且強度差值在  $\pm 5$  以內，此像素就在 S 以內。

接著算出範圍 S 內所包含的的像素之平均向量(mean vector)，將中心點移至算出來的平均向量。一直重複這個計算動作，直到平均向量開始出現重複為止。此時我們將起始的像素 p 之色彩資訊以最後停止的平均向量之色彩資訊來取代。最後使用分群(clustering)的方法來將區域區分出來。

接著要介紹一些空間域跟時間域合併的動態影像切割方法。在 Caplier，

Bonnaud 和 Chassery[6]提供了一個利用畫面差(frame difference)與合適的參考影像(reference image)來擷取影片中物件的方法，其中參考影像就是影片中不變的背景部份。這個方法提供了一個演算法來建立以及更新所要用到的參考影像。然後整個物件可以由比較目前畫面跟參考影像取得，即使物件停止移動或是只有些微移動，也可以擷取出來。

在處理時間域的方法中，估測運動向量是一個常常被使用到的方法，在 John Bea Kim 和 Hang Joon Kim 提出來的的方法當中[7]，也有使用到估測運動向量。而這個方法是一個以區域為基礎的動態物件切割，它主要是應用於錄影監視系統(video monitoring system, VMS)，整個過程是由兩個步驟所組成的，第一個階段是偵測畫面中改變的區域，利用一個適合的門檻值在畫面差中決定改變的區域，其中門檻值是自動產生的。第二階段以估測運動向量所得到的運動資訊來進行移動物件的切割。把第一個步驟找到的區域當中具有相似強度以及相似運動資訊的像素利用有權重的 k-mean 分群演算法產生我們所要的移動物件的遮罩。爲了提高運算速度，估測運動向量並不是對整張影像執行，只對偵測出有改變的區域來執行。

### 1.3 論文架構

在我們所提出來的的方法當中，並沒有利用到運動向量的估測，因爲估測運動向量花費的時間比較長，所以在時間域上，是利用畫面間像素強度改變的資

訊以及建構和維護一個可以信任的背景影像。這方法是針對於用在場景不會改變的影片上。

第二章將會提到我們空間域切割所用的演算法，分水嶺演算法，此演算法能將空間域的資訊找到。接下來第三章將會介紹我們再時間域的演算法，此演算法主要是利用畫面間，各像素強度改變的資訊，來建構背景以及切割物件。第四章是將空間域資訊跟時間域資訊結合，以產生一個完整的物件遮罩。最後將呈現一些實驗的結果。



# 第二章

## 空間域的切割

空間域影像切割是對於靜態的影像做切割，也就是影片中的一個畫面 (frame)。它切割出來的結果，必須盡量跟人類視覺上的看法相同。而大部分的方法是將影像中空間的特徵切割出來，例如：邊、顏色相似區域或是紋理相似的區域等等，也就是說，將影像分成許多強度相似的區域，並且以準確的邊界來將這些區域隔開。我們所提出的方法中利用到 morphological gradient 和分水嶺(watershed)兩個演算法。下面將對這兩個演算法做進一步的說明。



### 2.1 Morphological Gradient

首先，在 Morphological 處理中的基本運算有下面幾個：

反射(Reflection)：

$B$  為一個在  $Z^2$  上的集合，而  $B$  的 reflection 我們記作  $\hat{B}$ ，定義成：

$$\hat{B} = \{\omega \mid \omega = -b, \forall b \in B\}$$

遞移(Translation)：

$B$  為一個在  $Z^2$  上的集合，而  $B$  在一點  $z = (z_1, z_2)$  的 Translation 我們記作  $(B)_z$ ，定義成

$$(B)_z = \{c \mid c = b + z, \forall b \in B\}$$



膨脹(Dilation)：

A 跟 B 皆為在  $Z^2$  上的集合,用 B 對 A 作 Dilation 我們記作  $A \oplus B$ ,定義成

$$D(A,B) = A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

侵蝕(Erosion)：

A 跟 B 皆為在  $Z^2$  上的集合,用 B 對 A 作 Erosion 我們記作  $A \ominus B$ ,定義成

$$E(A,B) = A \ominus B = \{z \mid (B)_z \subseteq A\}$$

在膨脹(Dilation)和侵蝕(Erosion)中的集合 B，我們稱之為結構元素 (Structuring element)。圖 2.1 為膨脹和侵蝕的示意圖。圖 2.1(a)中所指的 A、B 也就是定義中的集合 A 跟 B，圖 2.1(b)中深色的區域則是 A 跟 B 作膨脹運算後的結果，圖 2.1(c)中有顏色的區域則是 A 跟 B 作侵蝕運算後的結果。



(a) 集合 A 跟集合 B



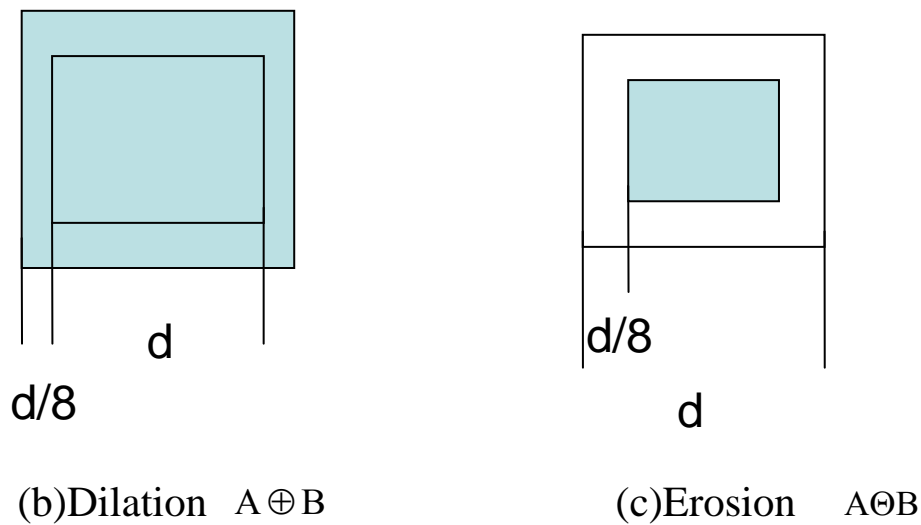


圖 2.1 Dilation 和 Erosion 之示意圖

前面所提到的膨脹跟侵蝕是用在二元影像(binary image)上，若要在灰階值的影像上應用膨脹和侵蝕，方法需做下述之修正。若欲處理的影像中座標為(x,y)的灰階值表示為  $f(x,y)$ ，所使用的結構元素(structuring element) 中座標為(x,y)的灰階值表示為  $b(x,y)$ 。

#### 灰階值的膨脹(Gray-scale dilation)

以  $b$  對  $f$  作灰階值膨脹(Gray-scale dilation) 我們可以記作  $f \oplus b$ ，定義如下：

$$(f \oplus b)(s,t) = \max\{f(s-x,t-y) + b(x,y) \mid (s-x,t-y) \in D_f; (x,y) \in D_b\}$$

其中，(s,t)是欲處理的影像當中，正在執行運算的像素座標。 $f$ 和  $b$ 皆為函數，而不是集合， $D_f$ 跟  $D_b$ 分別是  $f$ 和  $b$ 的定義域。上式是說，我們將結構元素中每個位置的值分別跟影像中相對應的值做相加，其中結構元素的中心位置(也

就是結構元素中座標是(0,0)的位置) 是對應到影像當中的座標(s,t)，相加結果中的最大值即是此像素膨脹的結果。

灰階值侵蝕(Gray-scale erosion)：

以  $b$  對  $f$  作灰階值侵蝕(Gray-scale erosion) 我們可以記作  $f \ominus b$ ，定義如下：

$$(f \ominus b)(s, t) = \min\{f(s+x, t+y) - b(x, y) \mid (s+x), (t+y) \in D_f; (x, y) \in D_b\}$$

在此處，我們將結構元素中每個位置的值分別跟影像中相對應的值做相減，然後取相減結果中的最小值。

最後，結合灰階影像膨脹和侵蝕的結果，我們可以得到此影像的梯度 (gradient)



$$g = (f \oplus b) - (f \ominus b)$$

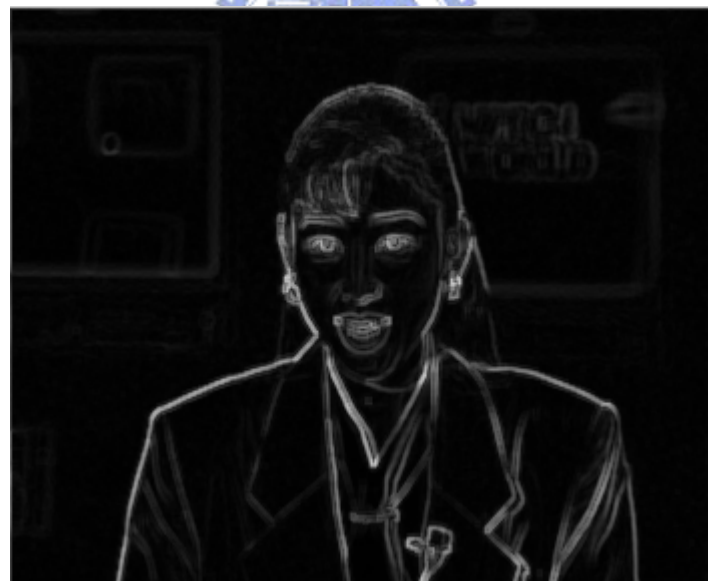
上面的式子是說，將要取梯度的影像分別做膨脹跟侵蝕，然後將膨脹後的影像減去侵蝕後的影像所得到的差值影像就是此張影像的梯度。

當一張影像，經過了梯度(gradient)運算的處理之後，所產生的結果有一些特性，影像中邊緣的部分，會產生比較大的梯度(gradient)值(亮度較強)，而影像中灰階值變化比較小的部分，產生的梯度(gradient)值會比較小，因為有這樣的特性，所以梯度影像有助於接下來分水嶺演算法的實現。另外，在影像序列中，常會有光線的改變，在我們利用梯度運算的處理以後，可以消除因光線改變所帶來的影響。圖 2.2 是影像經過處理後的結果，2.2(a)是原圖， 2.2(b)則是

經過梯度運算的結果。



(a)



(b)

圖 2.2 梯度運算的結果

## 2.2 分水嶺切割(Watershed segmentation)

分水嶺切割是將影像分成相同性質的封閉區域，而且區域之間並不會有重疊發生。所以我們將分水嶺切割作為空間域分割所用的演算法。

### 2.2.1 用氾濫(immersion)模擬的分水嶺演算法

當我們把一張影像的灰階值當成是地理上表面高度的高低起伏時，則此張影像在地理的觀點上就猶如有高山、平原、盆地等等的地形分佈。圖 2.3 為分水嶺演算法的示意圖，圖中地勢越高的地方則是影像中強度越強的地方。圖 2.3 中有二個集水盆地(catchment basins)，每一個盆地中都有高度最低的區域，所以在整張影像當中，每塊局部的最小值都會形成一個集水盆地。我們以氾濫的模式，去模擬水從各個盆地的最低處開始慢慢的向上漲，當有兩個不同盆地的水上漲到要互相接觸到時，就用一個水壩將兩邊隔開，而此時的水壩即是我們所要找的分水嶺線(watershed line)，當水漲到最高點的時候，所有的分水嶺皆會被找出來。

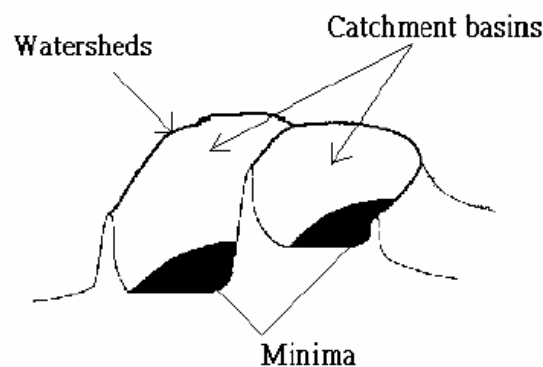



圖 2.3 分水嶺示意圖

## 2.2.2 分水嶺演算法之實現

Vincent 和 Soille's algorithm[5]是最常被使用的到的一個找出分水嶺的演算法。接下來將簡單的敘述此演算法的處理流程：

- 步驟 1) 先對要處理的影像  $F$  的每個點作梯度(gradient) 的運算，找出每個點的梯度值。而經過此運算後的影像稱為  $G$ 。
- 步驟 2) 將  $G$  量化成 32 個梯度等級(gradient level)，並對  $G$  中的所有像素依梯度值由小到大來做排序。
- 步驟 3) 因為每塊局部最小的區域都會形成一個集水區域，所以將那些擁有最小梯度等級的像素找出來，這些像素相當於某些集水盆地的最低處，並且將在同一個盆地中的最小梯度等級像素給予一樣的編號。且每個盆地有自己所屬的編號。
- 步驟 4) 接著將對影像中的像素由低到高的梯度等級來做處理。假設我們目前正要對梯度等級為  $N$  的像素來做處理，圖 2.4 中顏色較深的部份是梯度等級小於  $N$  的像素所構成的集水盆地，顏色較淺的是梯度等級為  $N$  的像素，而白色的部份則是梯度等級大於  $N$  的像素。將圖中已找出來的集水盆地當作是中心(也就是顏色較深的區域)，在這些區域周圍一圈的像素，若是有梯度等級為  $N$  且還沒有被編號過的像素，我們將之給予和集水盆地相同的編號。不斷重複這個動作，每做一次各個集水區便會向外膨脹一圈擁有梯度等級為  $N$  的像素，直到各集

水區周圍已經沒有符合條件的像素可以來做膨脹為止。此時，若還有梯度等級為  $N$  的像素沒有被給予編號，這些像素構成的區域相當於圖 2.4 中的”新發現之集水盆地”，也就是說這些像素構成的區域將會是另一些新的集水盆地之最低處，我們就利用步驟 3 之方式給予這些像素編號。

- 步驟 5) 當所有梯度等級為  $N$  的像素都經過編號後，檢查這些被編號的點，當像素周圍有跟本身編號不同編號的像素存在，則此像素的位置就是分水嶺的位置，我們就將此像素標記成分水嶺，以此條件，找出所有再分水嶺位置的點。而被分水嶺隔開的集水區，將合併成一個大的集水區，再繼續執行步驟 4，對梯度等級  $N+1$  的像素來處理。
- 步驟 6) 當做完最高梯度等級的處理之後，所有的分水嶺的點也都會被找出來。

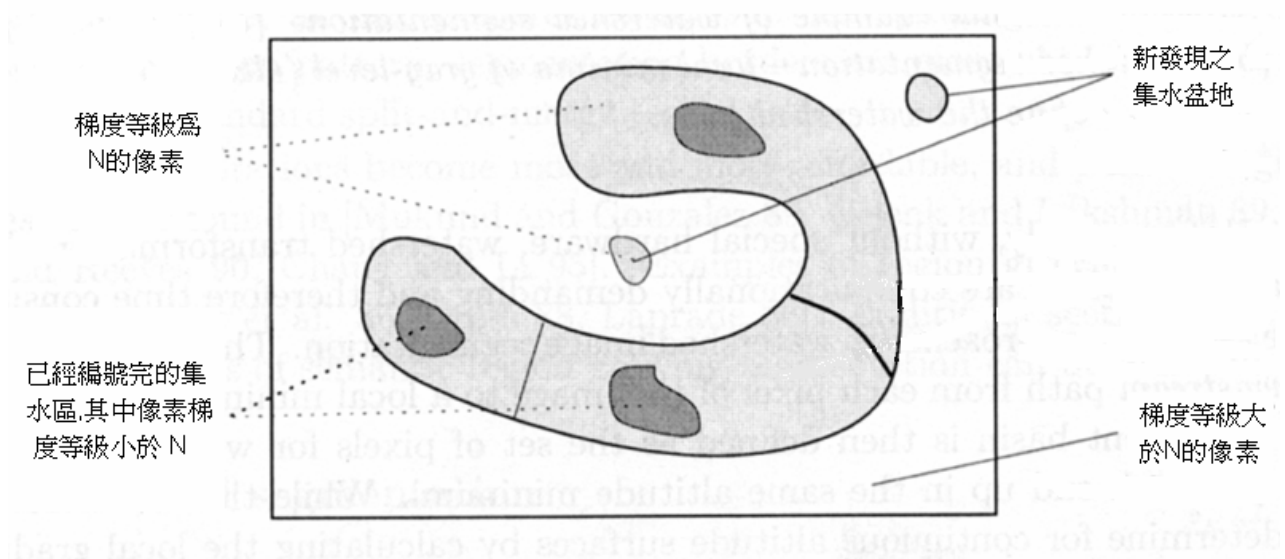


圖 2.4 分水嶺演算法步驟示意圖



當執行完分水嶺的演算法的之後，我們可以發現，所找出來的分水嶺都是位於影像中梯度值比較強的位置，也就是說是在影像中邊緣部份。圖 2.5 中是經過分水嶺演算法的結果以及他們的原圖。

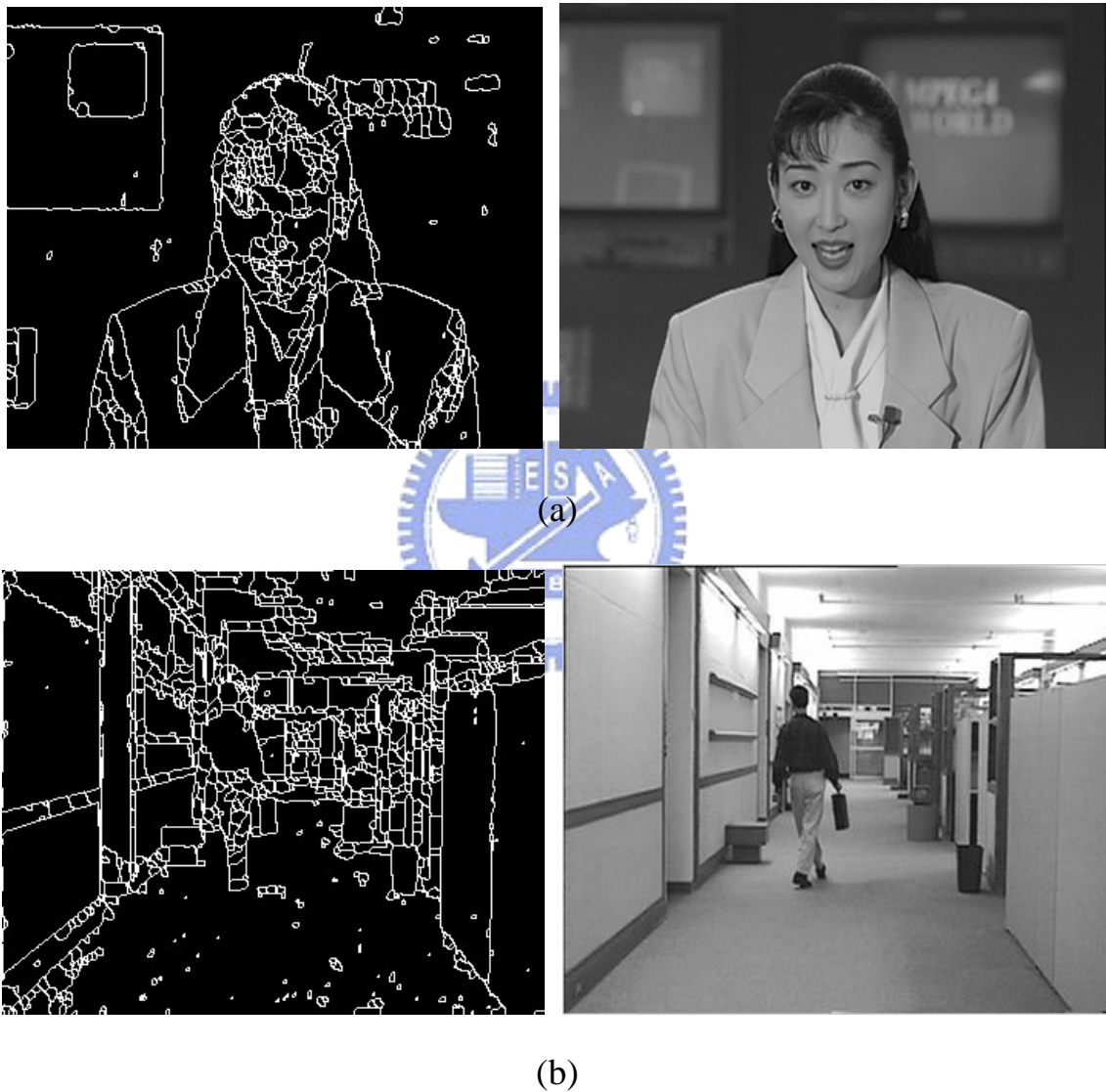


圖 2.4 分水嶺之結果 (a)左邊為 akiyo 分水嶺之結果右邊為原圖

(b)左邊為 hall 分水嶺之結果右邊為原圖

# 第三章

## 時間域的切割

在上一章我們介紹的空間域切割所用到的分水嶺演算法，接下來我們在這章要進入時間域的切割。而時間域的切割最主要的目的在於，切割出影片中在運動的前景(運動的物件)，也就是將會移動的前景跟靜止的背景分開。時間域切割的基本觀念為，利用影片中影像序列之間的相關性來區分出移動的前景跟靜止的背景。

所以通常在做物件的切割之前，會先做場景切換的偵測，因為當場景切換時，兩張畫面之間幾乎沒有關聯性，所以就不會對正在切換的畫面作切割，我們利用了 Munchurl Kim 的方法[8]取畫面間差值的平均絕對值(mean absolute difference, MAD)來判定是否發生場景切換。

### 3.1 概要

時間域切割中有兩項資訊是最常被利用參考到的，分別是影像中的運動資訊和畫面間強度的改變。

首先，影片中物件移動的資訊是以運動向量(motion vector)來表示，而運動向量可透過運動估測(motion estimation)的技術來得到。許多方法中使用了物件移動的資訊來做切割[9]，通常在影像序列中具有相似運動方向以及速度的區塊



被視為同一個物件。

畫面間強度的改變所指的是每個像素再不同畫面其強度的差異，之所以在影像序列中像素強度(intensity)的改變為一項重要資訊，是因為移動物件在不同畫面間的位置、形狀或是內部紋理會有所不同，造成在這些位置的像素其強度在不同畫面間亦有所不同。我們提出的時間域切割演算法就是利用此項資訊來做動態物件的切割。利用像素強度改變的資訊來建構和維護一個背景資訊(靜止不動的部分)，而將要做分割的畫面跟已建構好的背景做比較，若是跟背景不同的部份便被判定為移動的物件的部份。

### 3.2 演算法的整個流程



演算法可以分成五個主要步驟，圖 3.1 便是這五個主要步驟的流程圖。畫面差(frame difference)主要的目的就是在取得兩張連續畫面間所產生改變的部份，所以畫面差遮罩中被標記的部份則是判定為有所改變的部份。背景建構(background construction)的動作將會建構出一個可靠的背景資訊，而背景-目前畫面差(background-current frame difference)會將目前要做分割的畫面跟背景做比較，找出和背景不同的部份。將上述三個動作所取得的資訊整合起來，即可得到一個初始的物件遮罩，這個遮罩當中會有雜訊，所以我們還需做最後的處理，將雜訊去除得到一個精確的物件遮罩。

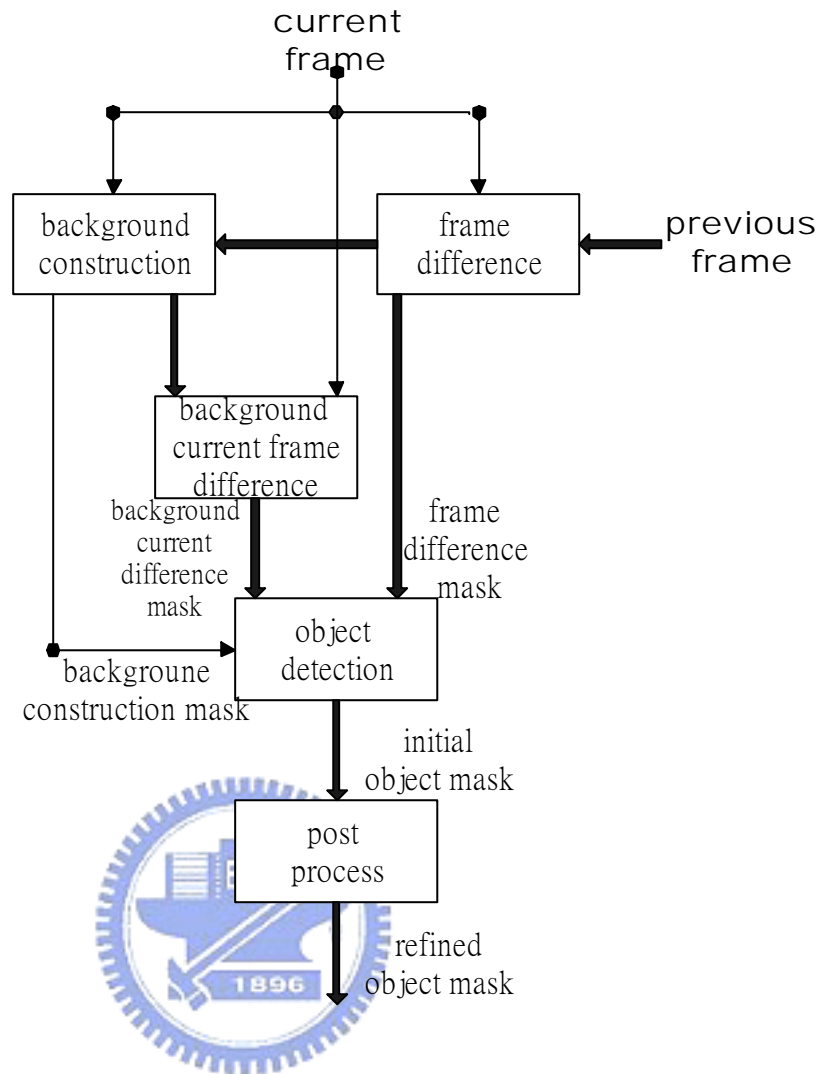


圖 3.1 時間域切割之流程圖

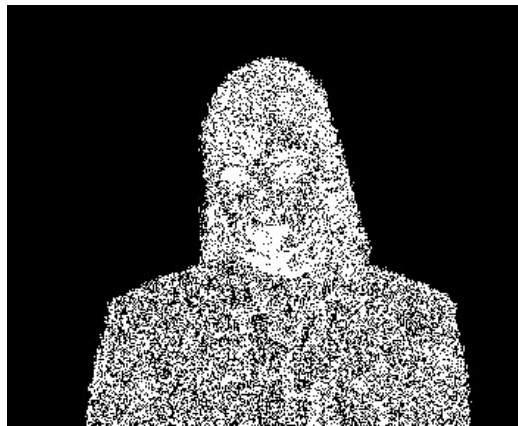
### 3.2.1 畫面差(Frame Difference)

畫面差的做法是將目前正要處理的畫面跟前一張畫面的每一個相對座標的像素相減，若相減出來的絕對值大於門檻值，便認為此像素有所改變。我們使用一個畫面差遮罩(frame difference mask, FDK)來紀錄哪些像素被認為是移動的像素，且這個遮罩也會被傳給背景重健的步驟當作輸入。下面的式子將實現此動作。

$$FDM(x, y) = \begin{cases} 1 & \text{if } |I_t(x, y) - I_{t-1}(x, y)| \geq \text{門檻值} \\ 0 & \text{otherwise} \end{cases}$$

其中  $I_t(x, y)$  為時間  $t$  的畫面中，座標為  $(x, y)$  的像素其強度值

若是我們只用畫面差當作分割的資訊，如此一來其分割結果的品質就跟許多因素有很大的關係，像是背景的雜訊、物件的運動速度或是前景跟背景的對比，這些都會對結果產生影響。所以光是以畫面差來做切割比較困難，且其結果也未必滿意。圖 3.2 分別是 *Akyio* 和 *hall* 兩段影片做完畫面差所產生的畫面差遮罩 (FDM)。



(a)



(b)

圖 3.2 FDM (a)akiyo 的 FDM (b)hall 的 FDM

### 3.2.2 背景的建構(Background Construction)

由於只利用畫面差會因許多外在因素而使得結果不令人滿意，但是背景資訊和目前畫面的差異會比兩張連續畫面來的大，所以我們要建構一個可靠的背景。背景建構就是爲了要由影像序列中建構出一個可靠的背景資訊來提供給整個切割的演算法使用。

我們利用兩項資訊來做背景的建構，一個是目前正在處理的畫面之影像，另一項資訊就是上一個步驟產生的 FDM，建構出來的背景必須存放於一個背景緩衝器(background buffer)B 內，這個背景緩衝器 B 在一開始的時候是空的。我們建構或是更新背景緩衝器內像素的主要原則就是，當一像素在連續 N 張畫面沒有改變，我們才認爲它是背景像素，將之加入到背景緩衝器內。所以建構以及更新主要是靠 FDM 所提供的資訊，因爲在畫面差步驟當中會去判段此像素是否有改變。FDM 若是 0 表示此像素沒有改變，此像素則有可能是背景像素，但是我們現在要建構的是一個可靠的背景，則不能很輕易的就將一個可能是背景的資訊拿來建構背景，因爲有些屬於移動緩慢物件的像素或是物件中的雜訊也可能在短短幾張畫面中被認爲是沒有改變的，這樣會將錯誤的資訊加入至背景緩衝器內，而使得之後的切割產生不正確的結果。因此，我們在建構背景或是更新時，是當某個座標的像素在連續一定數量的畫面被認爲是背景時，我們才建構或是更新該像素在背景緩衝器內的資訊。爲了達到這個目標，我們使用到一個稱爲 stationary index(SI)的陣列，它的大小跟畫面的大小是相同

的，是用來紀錄各個像素連續被認為沒有改變的次數，換句話說，當某個像素在畫面差步驟被認為是沒有改變的像素時，就在 **SI** 對應的位置上加一，但是當某個像素一但再畫面差中被認為是有所改變時，該像素在 **SI** 對應的位置重設為 0。我們給定一個門檻值 **N**，當某個像素連續被認為是背景的次數等於 **N** 時(即該像素對應的 **SI** 位置紀錄是 **N**)，我們便將該像素位在目前畫面的資訊加入到背景緩衝器當中。下面式子將實現上述動作

$$SI_t(x, y) = \begin{cases} SI_{t-1}(x, y) + 1 & \text{if } FDM(x, y) = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$B(x, y) = I_t(x, y) \quad \text{if } SI_t(x, y) = N$$

其中 **B(x,y)**是背景緩衝器內座標(x,y)之像素其背景資訊，**SI<sub>t</sub>(x,y)**是在時間 **t** 座標(x,y)之像素其強度連續被認為是沒有改變的次數。

另外，在此步驟我們也會去維護一個背景建構遮罩(background construction mask，**BCM**)以紀錄各個像素的背景資訊是否已經建構，它一開始皆設為否定(false)，而當一個像素的背景資訊第一次加入至背景緩衝器時，就將**BCM**設定為真(true)。圖 3.3 是 *Akiyo* 和 *hall* 兩段影片所建構出來的背景。



(a) Akiyo 影片所建出來的背景



(b)hall 所建構出來的背景

圖 3.3 建構出來的背景資訊

### 3.2.3 背景-目前畫面差(Background-Current Frame Difference)

建構完背景後，我們將對正要處理的畫面跟建構的背景做比較，產生一

個背景-目前差遮罩(background-current difference mask, BCDM), 下列的式子實現此步驟

$$BCDM(x, y) = \begin{cases} 1 & \text{if } |I_t(x, y) - \text{Background buffer}(x, y)| \geq \text{門檻值} \\ 0 & \text{otherwise} \end{cases}$$

此步驟和畫面差唯一的差別就在於在此步驟是將目前的畫面跟背景緩衝器(background buffer)內的值相減, 而不是跟前一個畫面相減。圖 3.4 為 hall 經過此步驟產生的結果, 很明顯的看出來, 它所產生的分割資訊 BCDM, 比圖 3.2(b) 的 FDM 準確, 尤其是在影像中人的內部, 在 FDM 中無法完整的呈現, 但是在 BCDM 中卻能較完整的呈現。這就是建構背景的最大好處。

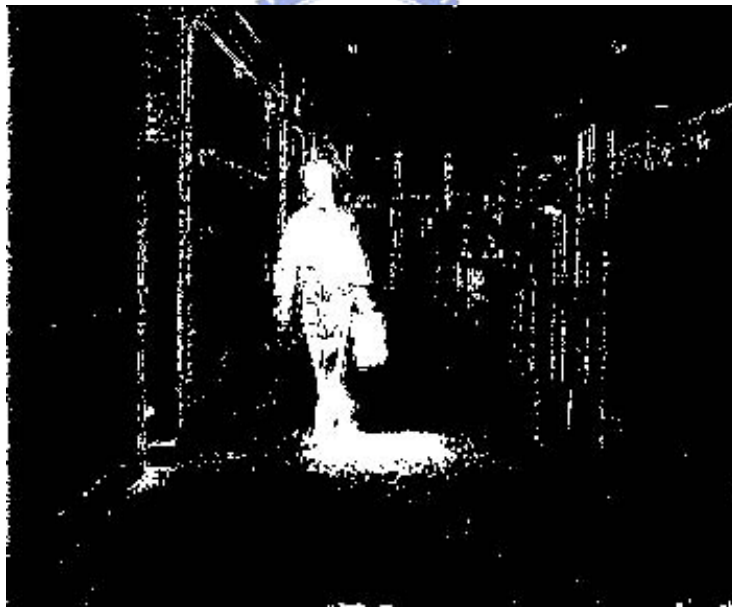


圖 3.4 hall 的 BCDM

### 3.2.4 偵測物件(object detection)

經過前面的處理取得一些物件相關的資訊後，則利用這些資訊將物件偵測出來，以一個初始物件遮罩(initial object mask, IOM)來表示偵測出的物件。IOM 是判斷 BCDM(background current difference mask)、FDM(frame difference mask)和 BCM(background construction mask)所提供的資訊而產生的。表 3.1 描述了偵測物件的規則，影像中的每個像素皆是以此表來做判斷。表中物件像素(Object pixel)的欄位是表示該像素是否為物件像素。

BCM	BCDM	FDM	Object pixel
False	unused	1	Yes
False	unused	0	No
True	1	unused	Yes
True	0	unused	No

表 3.1 物件偵測判斷表

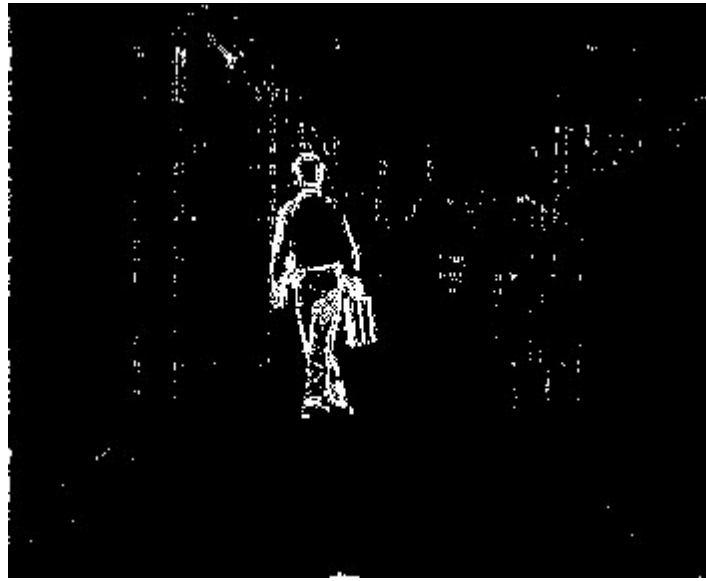
對於畫面中的每一個像素，首先，我們由 BCM 判斷它的背景資訊是否已經建構，若尚未建構，則它的 BCM 為否(false)，表示背景緩衝器內目前是空的。我們在偵測物件的時候就只能利用 FDM(畫面差遮罩)的資訊而不能參考 BCDM(背景-目前差遮罩)。所以若是該像素在 FDM 中是 1 的話，則判斷它是屬於物件，若是為 0，表示在 FDM 中被認為是靜止的像素，則判定它是屬於背



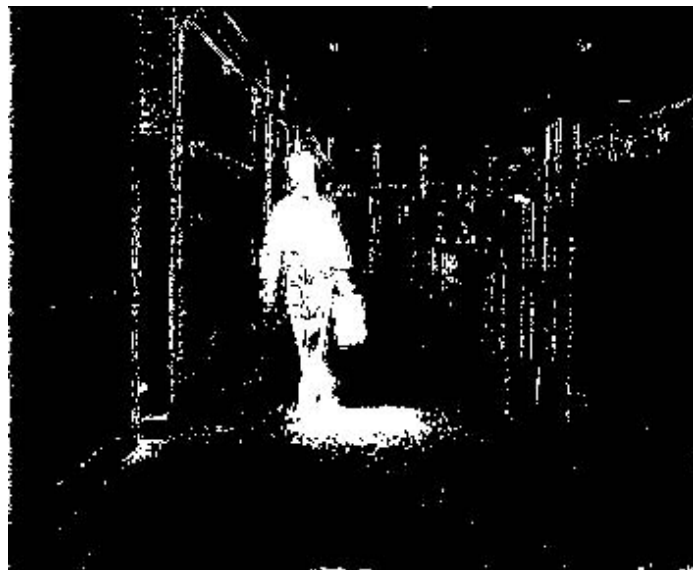
景。

若像素的背景資訊已建構，即 BCM 為真(true)，則表示此時我們可以利用背景緩衝器內的資訊，所以直接以 BCDM 來做判斷即可，BCDM 為 1 是物件像素，為 0 則是背景像素。而不採用 FDM 的資訊，因為在下面兩種情況當中，FDM 會提供錯誤的訊息。

第一種就是當 FDM 是 0，但是此像素並不是背景。當一個物件移動很緩慢，或是只有些微移動甚至短暫停止，且內部的紋理(texture)很單調，則它內部的像素就有可能在畫面差當中低於門檻值，而被認為是背景像素，如此一來便造成錯誤。但是在 BCDM 中，這些位置的像素其值為 1，因為這些像素是屬於物件，跟背景有明顯差異，所以不會因為上述的原因而不能將這些物件像素找到。圖 3.5(a)是 hall 影片中某一個畫面的 FDM，在這張影像的 FDM 當中，我們可以看出來有一個人，但是這個人身體中間的區域在畫面差中判斷為背景，主要是因為它一直向前走，在影像的序列中他的位置變動的不多，且他穿單色的衣服，所以畫面差中找到的只有此人邊緣的部份，而其身體中央反而被認為是背景，但是圖 3.5(b)是拿目前畫面跟背景相減的，很明顯的分辨出這個人的整體，因為在原本的背景當中並沒有這個人存在，一相減便很容易得到物件像素。



(a)



(b)

圖 3.5 物件些微改變在畫面差中的影響

第二種情形是 FDM 為 1，但是此像素卻是背景。以圖 3.6 來說明。圖 3.6(a) 是上一個畫面，圖 3.6(b) 為目前正在處理的畫面，圖 3.6(c) 則是背景，圖 3.6(d) 是兩張畫面的 FDM(畫面差遮罩)，黑色的部份是 FDM 為 0，紅色及白色處皆為

FDM 為 1 的部份，圖 3.6(e)是這兩個畫面的 BCDM(背景-目前差遮罩)，黑色處的 BCDM 為 0，白色處 BCDM 為 1。比較(d)跟(e)，發現在(d)中比(e)多出來的紅色部分就是 FDM 為 1，但卻是背景的部份。紅色部分是因為原本被遮蓋的部份，後來因為物件移開而露出來，所以他是前一張畫面的物件像素減去目前畫面的背景像素，而造成紅色的部份。但是在(e)中並不會有這種困擾是因為物件移開後露出的背景是跟背景相減，所以當然不會認為它是物件。

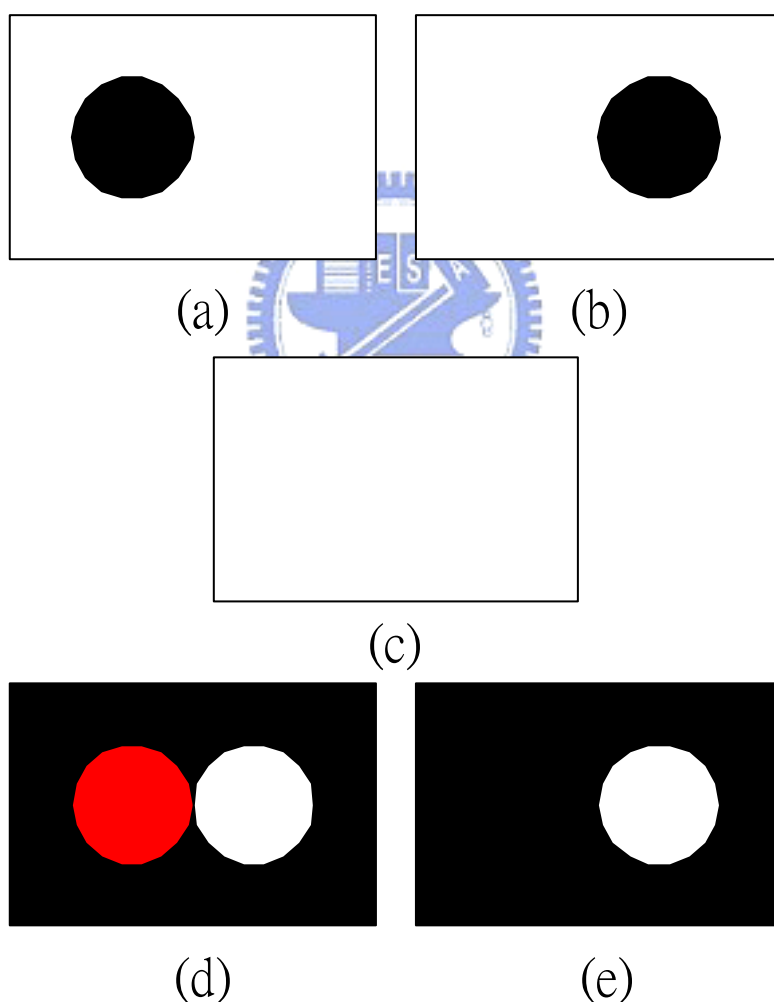
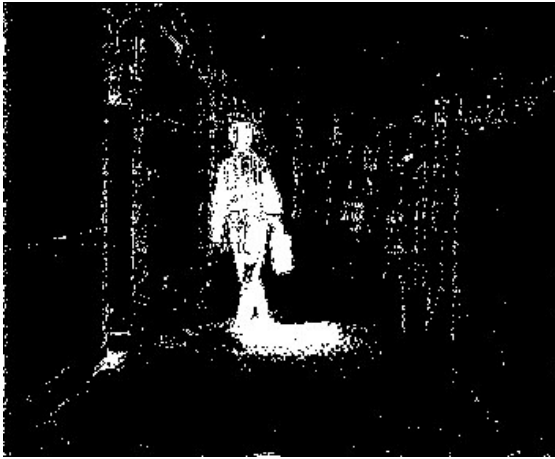


圖 3.6 示意圖

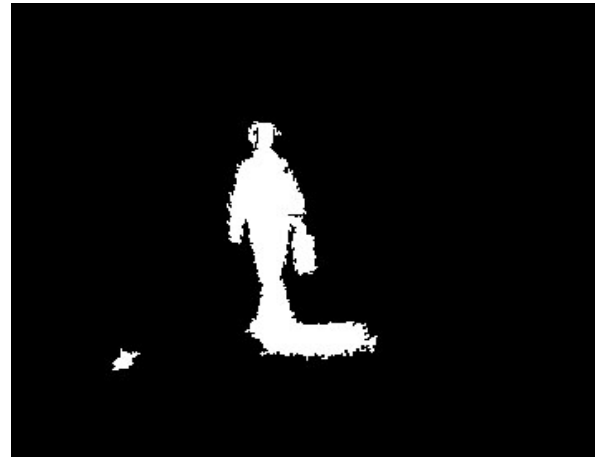
所以判段一像素是否為物件像素很簡單，若是已經建構背景資訊的像素就以 BCDM 來做判斷，否則就只有 FDM 可以利用來判斷。

### 3.2.5 後續的處理(post process)

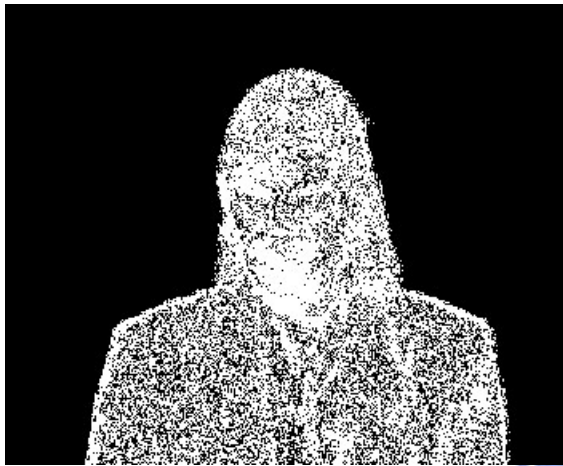
此時，得到一個初始的物件遮罩(initial object mask)，但是這個遮罩還不是理想的結果，因為遮罩中有許多的雜訊，這些雜訊有可能是因為光線造成的或著是攝影時的雜訊。所以要將這些雜訊的區域消除掉。圖 3.7(a)中，背景的部分有判斷成物件的雜訊(白色小區域)，圖 3.7(c)中在物件的部分也有誤判成背景的雜訊(黑色的小區域)，而這些都是我們要消除的區域。我們所使用的方法是将小的白色區域消除掉以及填補物件中的小破洞，所以必須先確定每個區域所包含的像素，利用區域成長(region growing)的方式來取得此項資訊。接下來給定一個門檻值，若是區域所包含的像素小於門檻值，我們就將它移除。因為在初始物件遮罩中的雜訊有兩種，一種就是在背景中的雜訊(小的白色區域)和另一種在物件裡的雜訊(小的黑色區域)，因此要分別消除這兩種雜訊。首先利用剛剛提到的方法來消除小的黑色區域(誤判成背景的雜訊)，這些雜訊通常是物件中的小洞。接著再移除小的白色區域(誤判成物件的雜訊)，這些雜訊是位於背景之中的。當經過這些消除雜訊的處理之後，得到一個精確的物件遮罩，我們稱之為精確的物件遮罩。圖 3.7(b)和(d)為經過消除雜訊的精確物件遮罩(refined object mask)。



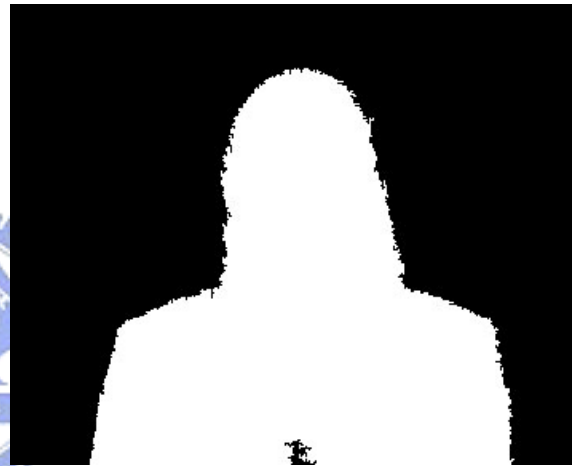
(a)



(b)



(c)



(d)

圖 3.7 經過消除雜訊後的物件遮罩

經過上述的五個主要的步驟，所產生的遮罩是時間域切割的最後結果。不過還有一個重點沒有提到，就是第一個步驟跟第三個步驟所使用到的門檻值，下面一節將會說明如何自動產生這個門檻值。

### 3.3 門檻值的自動產生

由於在產生畫面差遮罩和背景-目前差遮罩的過程都有用到門檻值來判定該像素是否為物件像素，因此門檻值扮演著重要的角色，若是門檻值不合適，使得不能夠將背景跟物件清楚的分辨，不是將許多背景誤判成物件就是切割出來的物件支離破碎的，所以門檻值的準確是一個重要的環節。我們將利用機率上的分佈以及空間上的特性來找出合適的門檻值。

圖 3.8，它是整個產生門檻值的流程圖，它有二處輸出的位置，在(A)位置的輸出，通常是品質較好得影像，也就是說影片拍攝時，攝影器材所造成的雜訊很微弱。另外(B)位置的輸出是經過更進一步空間的分析而產生的，因為此時攝影器材造成的雜訊比較強。接下來我們將對整個流程做說明。



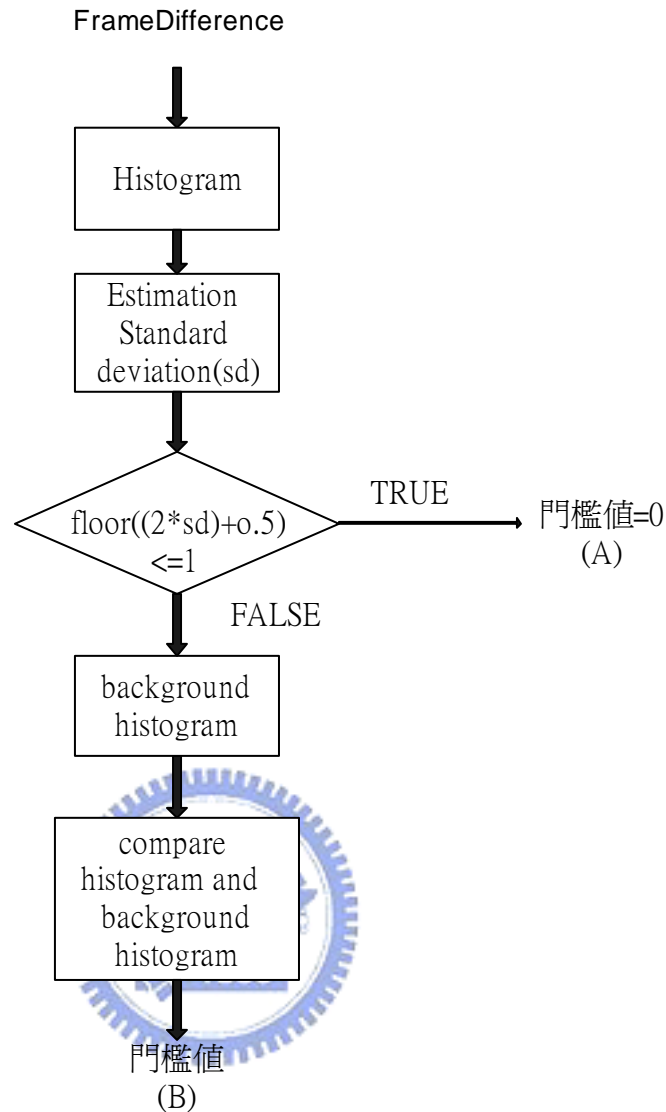


圖 3.8 自動產生門檻值流程圖

在影像串列中，我們利用像素強度之改變來分辨背景以及運動物件，其中背景是靜止的像素，所以強度沒有改變，也就是改變量為 0 的像素，但是門檻值並不一定設為 0，因為攝影器材有高斯分佈(Gaussian-distributed)的雜訊，使得背景部份的差值並不是皆為 0，所以背景像素在差值影像中是呈現以 0 為期望值的高斯分佈，圖 3.9 則是一個以 0 為期望值的高斯分佈，下式為此種分佈的機率密度函數，其中  $d$  表示兩張影像中，像素差值為  $d$ ， $\sigma$  為其標準差，

$p(d)$ 是差值為  $d$  的像素出現的機率

$$p(d) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-d^2}{2\sigma^2}}, \quad -255 < d < 255$$

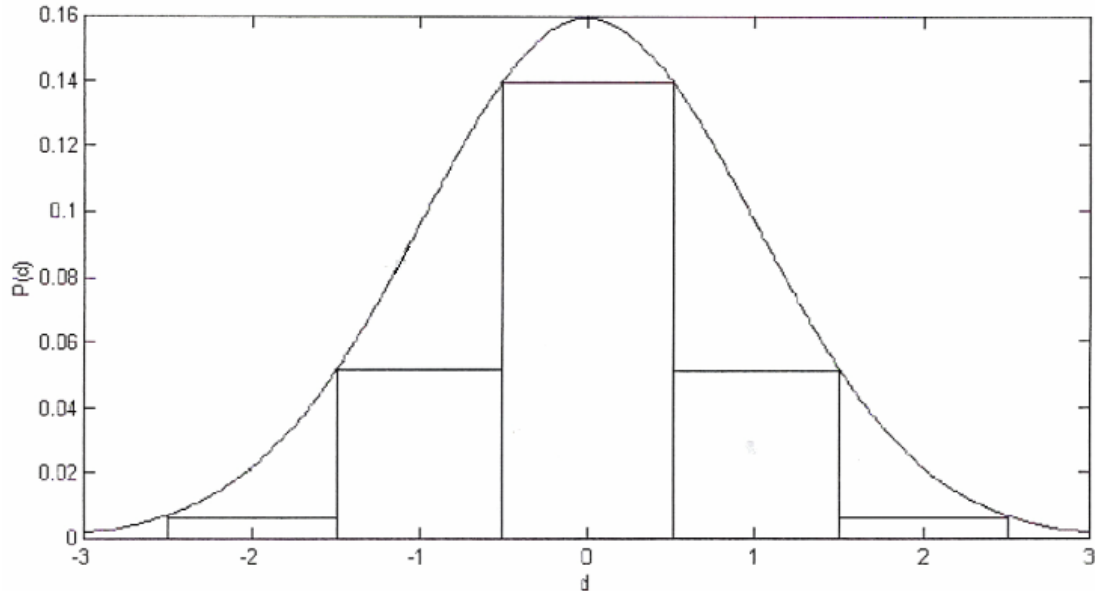


圖 3.9 期望值為 0 之高斯分佈

首先，產生欲取門檻值的差值影像其直方圖(histogram)，此直方圖所呈現的機率密度函數是由背景像素和物件像素混何在一起形成的，其中背景像素的機率分佈就是以 0 為期望值的高斯分佈，所以 95%的樣本會落在 $\pm 2$ 標準差之間，因此利用下面的推導來估計接近的背景像素機率分佈之標準差。因為背景像素是高斯分佈，所以其機率密度函數也就是

$$p(d) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-d^2}{2\sigma^2}}, \quad -255 < d < 255$$

當  $d=0$  時，上是可以簡化成

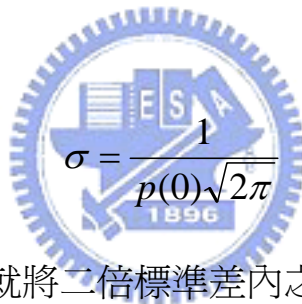
$$p(0) = \frac{1}{\sigma\sqrt{2\pi}}$$



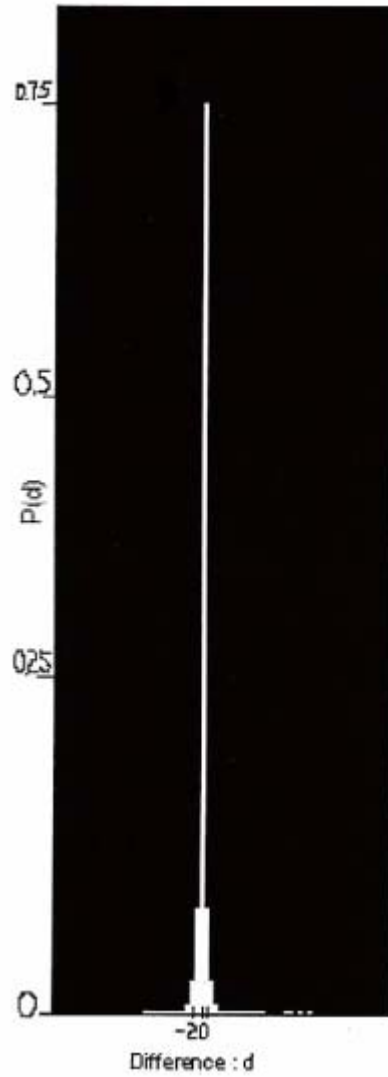
$p(0)$ 也就是背景像素中差值為 0 的像素出現的機率。若我們可以將  $p(0)$ 估計出來，便可以估計  $\sigma$ 。所以利用下式來估計  $p(0)$

$$p(0)=\text{number}(\text{pixel difference}=0)/\text{previous background pixel number}$$

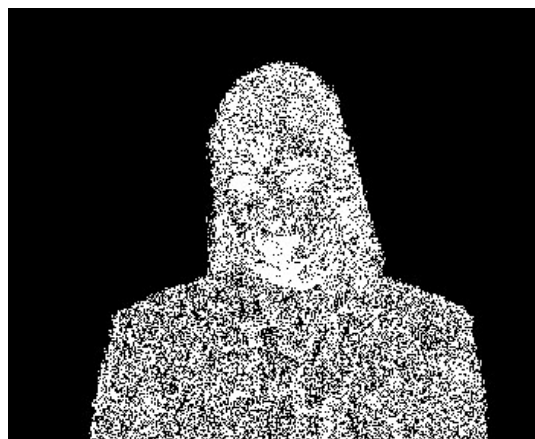
上式的  $\text{number}(\text{pixel difference}=0)$ 為此畫面中差值為 0 的像素之個數， $\text{previous background pixel number}$  為上個畫面切割的結果中背景像素的個數，其初始值為整張畫面像素的總數。會如此估計  $p(0)$ 是因為我們假設上個畫面中被認為是背景的像素個數會跟此畫面中背景的像素個數接近，所以就利用上個畫面被認為是背景像素的個數當做此畫面背景像素分佈的樣本空間，而估計出  $p(0)$ 。則標準差  $\sigma$  明顯的可以估計出來


$$\sigma = \frac{1}{p(0)\sqrt{2\pi}}$$

得到標準差之後，我們就將二倍標準差內之差值的像素認為是背景像素，以此作為空間上分析的基礎。但要注意的是，若二倍標準差四捨五入後不大於 1，我們便不做進一步分析，因為拍攝此影像串列的攝影器材所造成的雜訊非常小甚至沒有，所以我們直接以 0 當作門檻值，圖 3.10(a)是 Akiyo 影片的差值影像直方圖，它的標準差不到 0.5，因此直接以 0 當作其門檻值。圖 3.10(b)就是以 0 做門檻值所產生的 FDM。



(a) akiyo 之直方圖



(b)以 0 為門檻值產生之 FDM

圖 3.10 Akiyo 之直方圖以及 FDM

現在要對空間上做進一步的分析找出合適的門檻值，要產生另一個新的直方圖，稱做背景直方圖(background histogram)B，此直方圖統計的像素必須滿足下列條件

$$\begin{cases} |d| > 2|\sigma| \\ \text{周圍有超過4個背景像素} \end{cases}$$

換句話說，我們統計的像素是差值絕對值  $d$  大於二倍標準差，且其八連通的周圍像素，有四個以上為背景，此處所謂的背景像素是差值絕對值不大於二倍標準差的像素。滿足上述條件的像素，幾乎都是背景的背景的像素，所以統計這些像素的直方圖稱做是背景直方圖。最後我們將差值影像直方圖和背景直方圖做比較來找出合適的門檻值。在差值影像直方圖中呈現的機率密度函數是由背景像素跟物件像素混合在一起形成的密度函數，下式表示像素差值為 $|d|$ 之機率，它是由差值為 $|d|$ 之背景像素和差值為 $|d|$ 之物件像素一起統計而得到的。

$$p(|d|) = p(\text{background with } |d| \text{ and object with } |d|)$$

因此欲在二倍標準差外找到一個合適的門檻值將背景以及物件區分，此門檻值  $K$  必須是一個滿足下列條件的值

$$\frac{p(\text{background with } |k|)}{p(|k|)} < 0.5 \quad , k > 2\sigma$$

$$p(\text{background with } |k|) \approx B(|k|) / (h * w)$$

其中  $B(|k|)$  則是背景直方圖中，被統計到差值為 $|k|$ 的像素之個數，在將之除以整個畫面像素的總數即為  $p(\text{background with } |k|)$  的近似值。而此判定的不等式是表

示，在二倍標準差外，找到第一個  $K$ ，會使得影像差值絕對值為  $K$  的像素中，屬於背景的像素比屬於物件的像素少，則此  $K$  即為門檻值。圖 3.11 為 hall 影片其中一張畫面的差值影像值方圖，此畫面估計出來的標準差趨近於 2，所以需要空間上的分析，最後將適合的門檻值找到，而所找到的門檻值為 9。

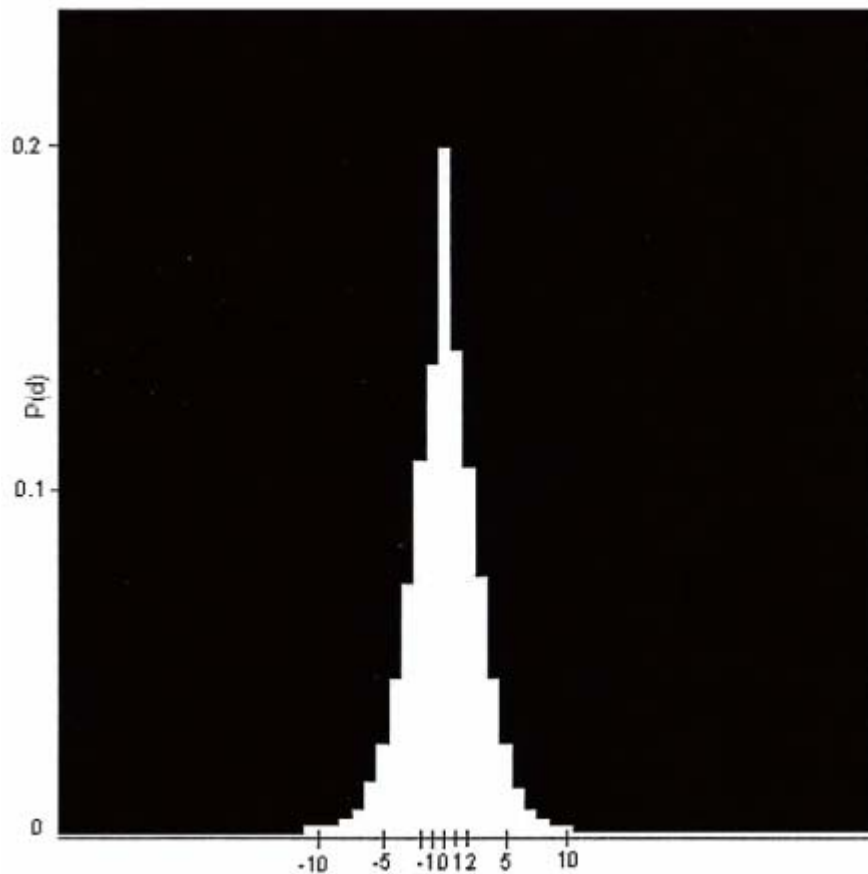
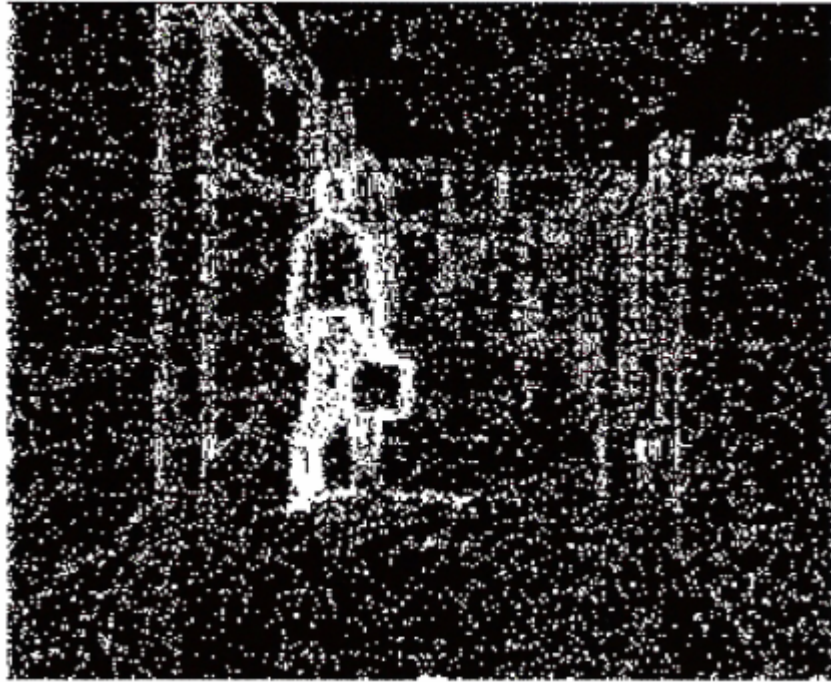


圖 3.11 hall 之影像差直方圖

圖 3.12(a)是以圖 3.11 直方圖來取二倍標準差 4 當作門檻值所得到的 FDM，其結果相當的不理想，雜訊非常的多，圖 3.12(b)則是利用演算法找到的 9 當作門檻值所得到的 FDM，遮罩中的雜訊數量明顯的比圖 3.12(a)少。



(a) 以 4 為門檻值產生之 FDM



(b) 自動產生之門檻值 9 的 FDM

圖 3.12 hall 影片不同門檻值之比較

# 第四章

## 空間-時間域切割及實驗結果

爲了使影像切割的結果更爲精確，我們將空間域切割的資訊跟時間域切割的資訊加以合併，這章將說明合併的方法以及實驗的結果。

### 4.1 結合時間域跟空間域的切割

將時間跟空間資訊合併有二個步驟，分別是以區域爲觀點和以邊界爲觀點，第一個步驟是先對區域來處理，再對邊界來加以修正補強。圖 4.1 是將空間域以及時間域的流程圖合併爲整個演算法的流程圖，從圖中可以看出，是將欲切割畫面的分水嶺結果跟時間域上最後產生的物件遮罩做結合。



#### 4.1.1 移動區域的偵測

在這步驟，利用目前要做分割的影像畫面  $F_k$  跟前一張影像畫面  $F_{k-1}$  在時間軸切割的結果，然後跟空間域的分水嶺結合，偵測出  $F_k$  中有所移動的區域。使用的方法爲，檢查在分水嶺結果的每一個區域(集水區)，若某區域  $R_i$  被偵測爲移動的區域，它會滿足下列的不等式

$$p = \frac{N}{M} > T_r$$

其中， $M$  是代表  $R_i$  中所包含像素的數目， $N$  代表  $R_i$  中，在時間域切割中被認爲是運動物像素的數目。就是當  $R_i$  內經過時間域切割後，判斷爲物件像素佔



$R_i$ 所有像素的比例  $p$  超過一個門檻值  $T_r$  時，我們就將整個  $R_i$  判斷為移動區域，反之則為背景的区域。如此一來可以將時間上切割不完全的地方加以修正。使結果更為精確。圖 4.2 則是做完此步驟的結果，4.2(a)是經過時間域切割後的物件遮罩，4.2(b)是做完此步驟後的結果，可以發現，原本遮罩中人影的部份被誤判成一個面積不小的物件區域，但是經過空間域的資訊修正後，將它移除為背景的部分，而且也將物件的邊緣修飾的比較平滑。

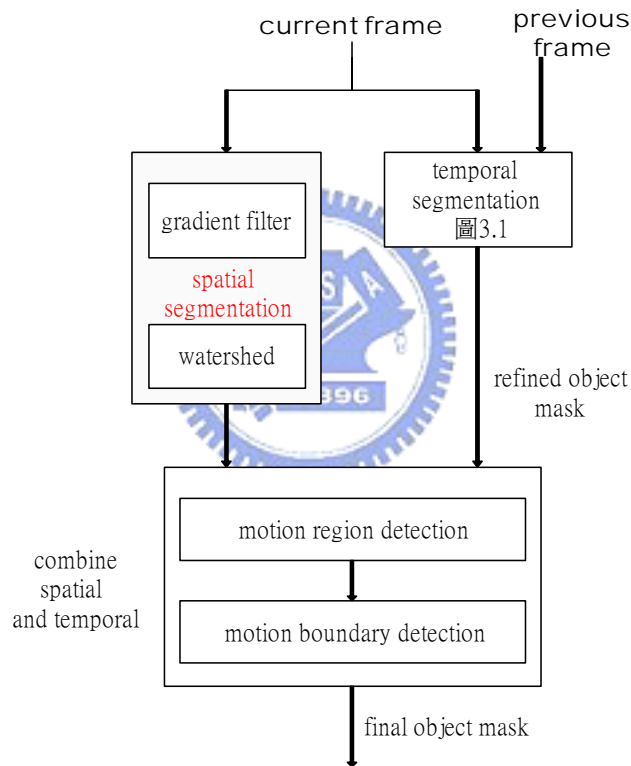
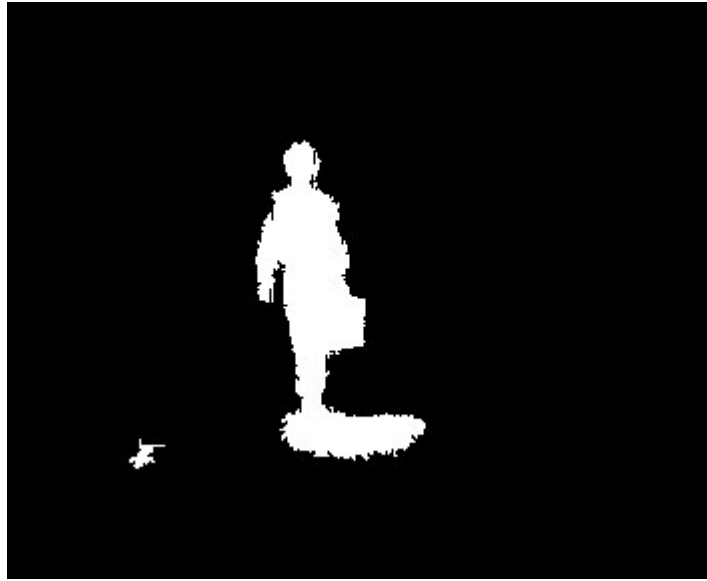


圖 4.1 合併時間空間域切割之流程圖



(a)



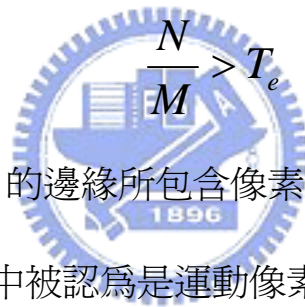
(b)

圖 4.2 移動區域偵測之結果



## 4.1.2 移動邊界的偵測

當兩張畫面之間物件只有些微的移動或是移動很緩慢，且其背景資訊還尚未建構完全時，若只利用區域偵測來判斷往往會漏掉這些移動些微的區域，將之認為是背景。而這些區域在時間域的移動偵測時，通常只有區域的邊界部份會被偵測到，切割出來，所以我們就利用移動邊界的偵測來將這些區域找出來。此處用到的方法跟上一個步驟的觀念類似，它們不同的地方是在區域偵測時是以區域內的點來判斷在此處是以區域邊界的點來做判斷，若判斷為移動的區域邊界，它會滿足下列不等式


$$\frac{N}{M} > T_e$$

上式， $M$  是代表包圍區域  $R_i$  的邊緣所包含像素之數目， $N$  代表包圍區域  $R_i$  的邊緣像素中，在時間域切割中被認為是運動像素的數目。它的理由跟區域偵測一樣，所以不多做解釋了。但是有一點必須要注意的，就是並不是所有的區域邊界都要做偵測，被我們偵測的區域邊界當中，它區域內的點還是必須有一定的比例要屬於時間軸所判斷為移動的點，但是這比例卻不足夠讓這個區域在區域偵測中判斷為移動區域，這些才是我們要拿來判斷的對象。圖 4.3 就是說明此步驟所修正的效果。圖 4.3(a) 是分水嶺的結果，圖(b) 是時間域切割的結果，圖 4.3(c) 是經過區域偵測的結果，此時發現臉上缺了一大塊區域，但是經過此步驟後便將此錯誤修正，結果為 4.3(d)。

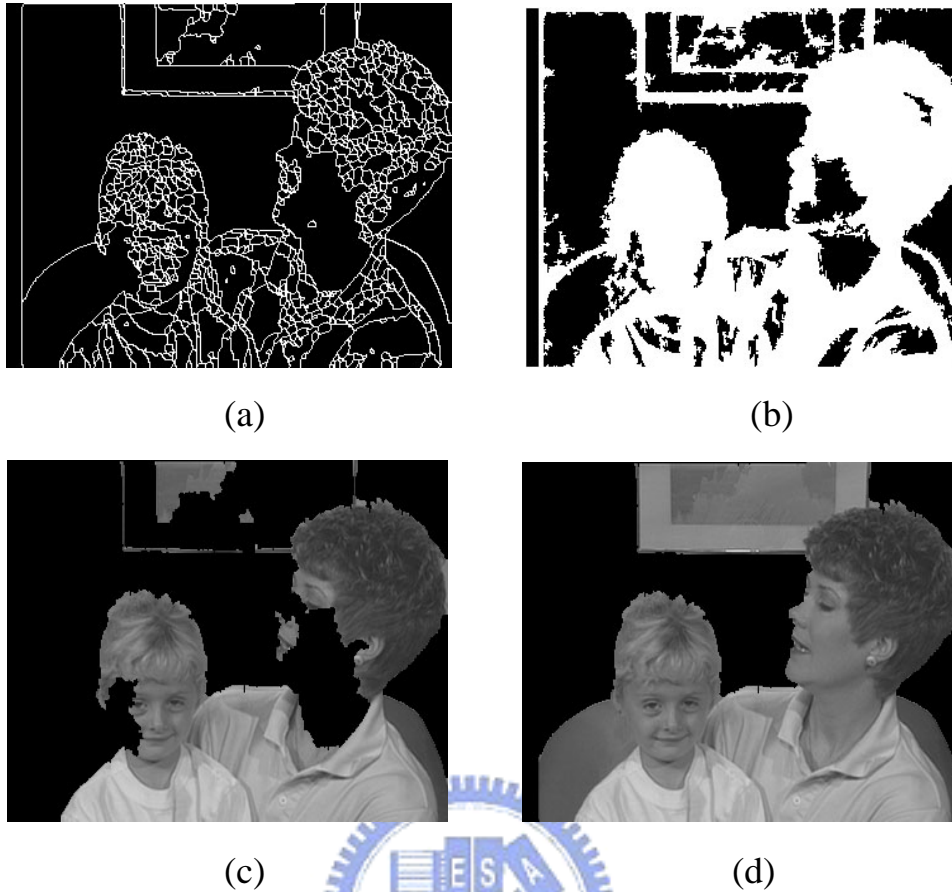


圖 4.3 偵測移動邊界的結果

而經過時間以及空間域的合併之後，不僅僅可以將許多錯次的判斷移除，也可以利用空間的資訊，使得切割結果中的物件邊界更為準確。

## 4.2 實驗結果

我們接下來對一些影片用之前所提到的演算法來加以分割，我們利用到的影片有兩種規格，其中比較大的是 CIF(影像大小是 288x352)，而比較小的是 QCIF(影像大小是 144x176)。圖 4.4 是 *Akiyo* 影片分割的結果，它是 CIF 的格式。共有 300 個畫面。下面的結果從左到右，從上到下，分別是畫面 1，30，60，.....270。而整段影片臉部改變的最頻繁。



圖 4.4 Akiyo 的分割結果

圖 4.5 是 News 影片分割的結果，它是 CIF 的格式。共有 300 個畫面。下面的結果從左到右，從上到下，分別是畫面 1，30，60，.....270。而整段影片中，除了播報員為運動物件外，後面的電視牆也是動態的。

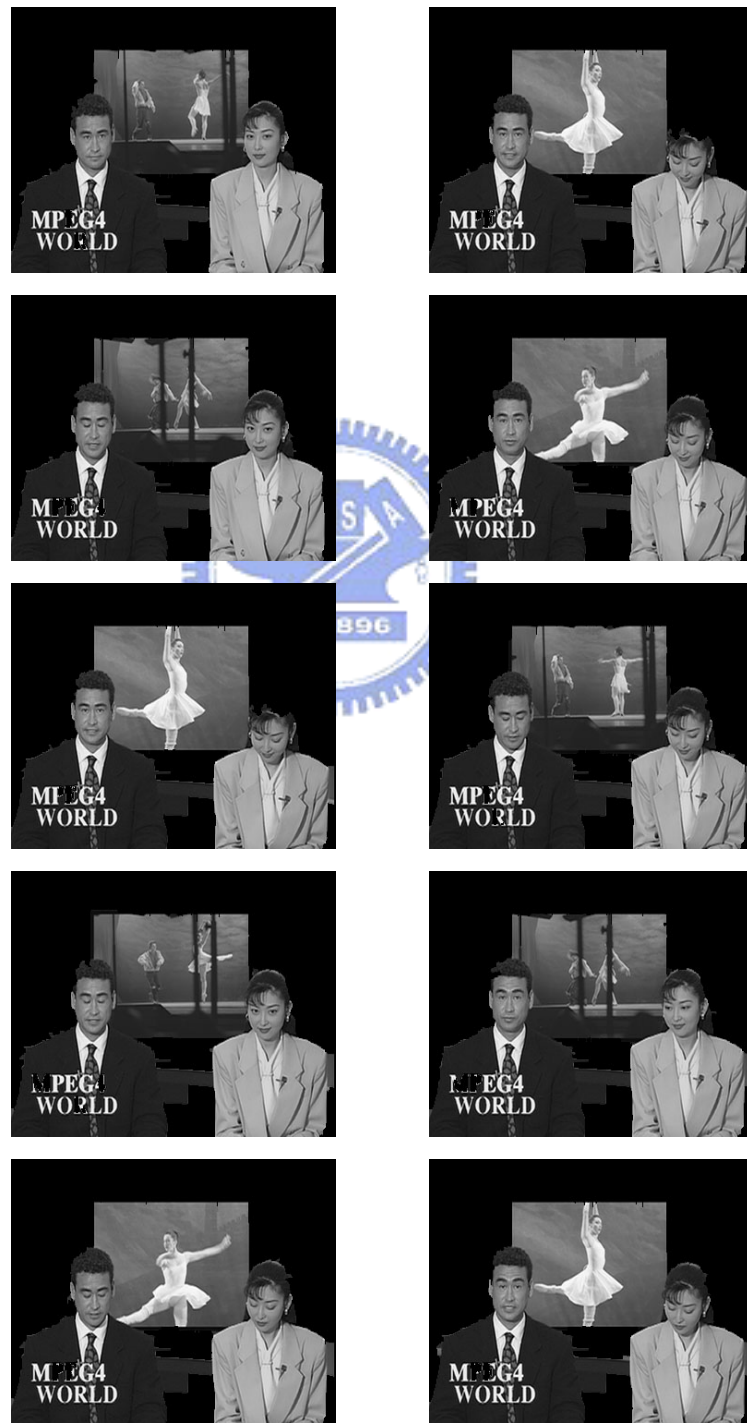


圖 4.5 news 的分割結果

圖 4.6 是 *claire* 影片分割的結果，它是 QCIF 的格式，共有 494 個畫面。下面的結果從左到右，從上到下，分別是畫面 1，30，60，.....300....450。



圖 4.6 *claire* 的分割結果

圖 4.7 是 hall 影片分割的結果，它是 CIF 的格式，共有 300 個畫面。下面的結果從左到右，從上到下，分別是畫面 1，20，40，60，.....300。在下面的結果當中，第一個畫面沒有東西是因為此時並沒有任何動態物件。在第 20 個畫面時，人一開始進入的畫面當中，其分割的結果並不太理想，主要是因為一開始的時候背景還沒有完全建構完成，再加上此段影片有相當多的雜訊，所以造成如此的結果。





圖 4.7 hall 的分割結果



圖 4.8 是 mother and daughter 影片分割的結果，它是 CIF 的格式，共有 300 個畫面。下面是其中四個畫面的分割結果。



圖 4.9 mother 的分割結果



# 第五章

## 結論

在影像序列的壓縮當中，物件的分割佔了相當大的重要性，它的結果會直接影響到壓縮的效益。若是能夠有效的將物件跟背景分開，這樣一來我們只需要對物件做壓縮，所以可以大大的提升壓縮的效益。但是在 MPEG4 以及一些壓縮的標準當中，並未提到一套有效的物件切割方法，因為對於不同型態的影片常常需要用到不同的方法。

在提出的方法當中，是針對於場景固定的影片而應用的，它主要的想法就是建構一個可靠的背景，需要被切割畫面，將會拿來跟背景的資訊做比較，若是強度跟背景相異，我們就將它歸類於是運動的物件，否則就認為它是背景，這些都是時間軸上的處理。爲了要使物件能夠更完整的被切割出來，我們就必須利用一些空間上的資訊，所以我們利用到分水嶺的演算法，在做分水嶺之前還必須將畫面先經過梯度(gradient)運算的處理，若是沒有這麼做，會造成影像過度分割，如此一來便失去了分水嶺的意義。經過梯度運算的影像完成分水嶺演算法之後會將相似的區域分割出來，且是封閉且不會重疊的。最後再將時間跟空間的結果加以合併，使得最後物件能夠完整的呈現以及物件的邊緣能夠比較準確的呈現。而這個方法可以用在一些新聞影片中播報員片段的壓縮或是一些監視系統的壓縮等等，因為這類的影片鏡頭都是固定的。

## REFERENCES

- [1] “MPRG-4 Video Verification Model version 18.0” ISO/IEC JTC1/SC29/WG11 N3908, January 2001.
- [2] “Information Technology – Coding of audio visual objects Part 2: Visual” ISO/IEC 14496-2:2003.
- [3] D. Comaniciu and P. Meer, “Mean Shift Analysis and Applications”, Proc. IEEE International Conference on Computer Vision, VOL.2, pp.1179-1203, 1999.
- [4] D. Comaniciu and P. Meer, “Mean Shift : A Robust Approach Toward Feature Space Analysis”, Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL.24, NO.5, pp.603-619, May, 2002.
- [5] L. Vincent and P. Soille “Watersheds in Digital Space: An Efficient Algorithm Based on Immersion Simulations”, Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL.13 , NO.6 , pp.583-598, June, 1991.
- [6] A. Caplier , L. Bonnaud and J.-M. Chassery, “Robust fast extraction of video objects combining frame differences and adaptive reference image”, Proc. IEEE International Conference on Image Processing , VOL.2, pp.785-788, 2001.
- [7] J.-B. Kim and H.-J. Kim, “Efficient region-based segmentation for a video monitoring system”, Proc. Pattern Recognition Letters 24, pp.113-128, 2003.
- [8] M. Kim , J.-G. Choi , D. Kim , H. Lee , M.-H Lee , C. Ahn , Y.-S. Ho “, A VOP

generation tool: automatic segmentation of moving objects in image sequences

based on spatio-temporal information” ,Proc. IEEE Trans. Circuit and System for Video Technology. VOL.9,No8 ,pp. 1216-1226,Dec 1999.

[9]C. Stiller and J. Konrad , “Estimation motion in image sequences”, IEEE Signal Processing Magazine . VOL.16,Iss. 4,pp 70-91,July 1999.

[10]S.-Y. Chien ,S.-Y. Ma and L.-G. Chen, “Efficient Moving Object Segmentation Algorithm Using Background Registration Technique”, IEEE Trans. Circuit and System for Video Technology. VOL.12 ,No.7,pp.577-586,July,2002.

[11]Wei-Jung Chien and Sheng-Jhy Wang , “The Study of Spatio-Temporal segmentation for image sequences”, Master Thesis, National Chiao Tung University, Hsinchu ,Taiwan,ROC,2001.



[12]R.C. Gonzalez and R.E. Woods, “Digital image Processing”, Addison Wesley Publishing Company,USA,1992.

[13]Linda G. Shapiro and George C. Stockman, “Computer Vision”, Prentice Hall,2001.