

國立交通大學

資訊科學系

碩士論文

利用前景和背景分析提升 H.264 之壓縮效率



On the Performance Improvement of H.264 through Foreground
and Background Analyses

研究生：林哲寬

指導教授：莊仁輝 教授

中華民國九十三年六月

利用前景和背景分析提升 H.264 之壓縮效率

On the Performance Improvement of H.264 through Foreground and Background Analyses

研 究 生：林哲寬

Student : Zhe-Kuan Lin

指 導 教 授：莊仁輝

Advisor : Jen-Hui Chuang

國 立 交 通 大 學

資 訊 科 學 研 究 所



A Thesis

Submitted to Institute of Computer and Information Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer and Information Science

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

利用前景和背景分析提升 H.264 之壓縮效率

研究生：林哲寬

指導教授：莊仁輝 博士

國立交通大學

資訊科學研究所



本論文針對現行的視訊壓縮標準 H.264，探討前景和背景分析的應用，目的在透過前景的影像品質和主觀視覺接受度提升、背景的资料量之降低，以及壓縮速度之提升等三方面，來改善整體標準的壓縮效率。由於 H.264 提供的編碼工具使用彈性很大，所以本論文在前景和背景區域已知的前提下，對於標準中可變區塊大小和失真模糊率的兩種編碼工具進行調整與分析，並提出兩個增加編碼效能的方法。同時，這些以前景和背景分析的編碼方法亦均能符合 H.264 標準內所規範的語法和語意。而實驗的結果也顯示，我們所提出的方法能相當有效的改善 H.264 的編碼效率。

On the Performance Improvement of H.264 through Foreground and Background Analyses

Student : Zhe-Kuan Lin

Advisor:Dr. Jen-Hui Chaung

Institute of Computer and Information Science
Nation Chiao Tung University



The objective of this thesis research is to study possible improvements of the encoding performance of the H.264 video standard given the foreground and background models of the source video sequence. The encoding performance is measured in three aspects: the PSNR, bit-rates and coding speed. By analyzing the variable block-size and rate-distortion coding tools of H.264, we propose two different encoding methods for the video foregrounds and backgrounds to enhance the video quality of the foreground objects and to reduce the compression bit-rates of the background area while ensure that the syntax and semantics of the encoded bit-streams agree with H.264. Experimental results show the proposed methods are quite effective in the performance improvement of the H.264 encoding.

致謝

這篇論文的誕生，要感謝曾經幫助過我、教導過我的人。首先感謝我的指導教授莊仁輝博士，在兩年來的研究及學習中給予熱心的指導與教誨，老師悉心的指導讓我在資訊領域有更廣的視野並且學習到更深入的知識，在此向老師致上最高的謝意。同時，也感謝口試委員陳玲慧教授、雷欽隆教授和顏嗣鈞教授的建議和指教，還有蔡淳仁教授指引我研究的重要資訊，使得本論文的内容更加充實。

另外，也感謝實驗室林建州、高肇宏和林泓宏三位博士班學長，在這兩年的研究生活中，提供給我研究的方向和知識上的教導。耀輝、存厚、兆良三位同學，在求學的生活中，因為有他們的陪伴與互相扶持，讓我度過了寫論文的艱苦日子。還要謝謝盈儀的陪伴，在我遇到困難和心情低潮時，給我鼓勵與幫助。

最後，我要謝謝我的家人，給我舒適無憂的求學環境，以及生活上的支持與照顧，才能順利取得碩士學位。僅以本篇論文，獻給我的家人，以幫助過我所有的人，謝謝你們。



目錄

摘要.....	I
ABSTRACT	II
致謝.....	III
目錄.....	IV
表目錄.....	VI
圖目錄.....	VII
第一章 簡介.....	1
第二章 H.264/AVC視訊壓縮標準.....	4
2.1 簡介.....	4
2.2 H.264/AVC視訊編碼工具.....	5
2.3 視訊壓縮位元串列架構.....	9
第三章 利用前景與背景來提升壓縮效能的探討與相關研究.....	14
第四章 可變區塊大小的方法.....	16
4.1 可變區塊大小的種類.....	17
4.1.1 畫面內預測(Intra Prediction).....	17
4.1.2 運動估測(Motion Estimation).....	18
4.2 固定區塊大小的分析.....	20
4.3 前景(Foreground) 和背景(Background)的使用.....	23
4.3.1 前景和背景的使用觀念和流程.....	23
4.3.2 前景和背景區塊種類的比較.....	26
4.3.3 利用Rate-Distortion分別處理前景和背景.....	30
4.4 codec相關參數的介紹.....	32
4.4.1 QP不同的影響.....	33
4.4.2 是否使用Hadamard.....	33

4.4.3 參考畫面的使用.....	34
4.4.4 不同Entropy Coding的影響.....	34
第五章 實驗結果與分析.....	35
5.1 實驗平台.....	35
5.2 評估標準.....	35
5.3 整張畫面使用相同大小區塊.....	36
5.4 使用Rate-Distortion來控制編碼.....	39
5.5 使用前景和背景分析編碼.....	43
5.6 比較編碼速度.....	47
5.7 問題與討論.....	49
第六章 結論與未來展望.....	51
參考文獻.....	52



表目錄

表 4.1	MPEG2/4 和H.264 所使用區塊大小的比較	16
表 4.2	對於I、P、B 畫面所可以使用到的巨區塊(區塊)的種類.....	20
表 4.3	影片序列區塊的PSNR和bitrate比較	20
表 4.4	影片序列區塊的PSNR和bitrate比較	22
表 4.5	前景和背景區塊不同的PSNR和bitrate比較	28
表 4.6	QP量化對應值	33
表 5.1	I畫面的區塊編碼結果	36
表 5.2	P畫面的區塊編碼結果	37
表 5.3	B畫面的區塊比較.....	38
表 5.4	比較 λ 值不同而影響的位元速率和PSNR值	39
表 5.5	Singer影片編碼方式	44
表 5.6	編碼速度的比較.....	48



圖目錄

圖 1.1 影片利用前景背景編碼的示意圖.....	1
圖 1.2 原始影像與前景和背景的切割圖.....	2
圖 2.1 1/4 像素精準度位置圖	6
圖 2.2 多重參考畫面.....	7
圖 2.3 4 x 4 轉換矩陣.....	8
圖 2.4 FMO的使用方式.....	9
圖 2.5 SP畫面的使用	9
圖 2.6 MPEG視訊位元串列架構	11
圖 2.7 GOP示意圖。N = 12，M = 3.....	12
圖 4.1 編碼流程示意圖.....	17
圖 4.2 Macroblock: 16x16, 4x4, 2 種區塊大小	18
圖 4.3 Macroblock : 16x16, 8x16, 16x8, 8x8, 4 種區塊大小	19
圖 4.4 8x8 block : 8x8, 4x8, 8x4, 4x4, 4 種區塊大小.....	19
圖 4.5 使用foreman影像結果的比較	21
圖 4.6 P畫面使用不同區塊大小影像結果的比較	22
圖 4.7 前景和背景應用示意圖.....	24
圖 4.8 區塊選擇流程圖.....	25
圖 4.9 I、P、B應用於不同前景的區塊種類	27
圖 4.10 表 4.5 前景和背景的PSNR值比較.....	29
圖 4.11 整段影片以不同 λ 值的資料量比較	31
圖 5.1 Singer主觀畫面品質比較	38
圖 5.2 四段影片相對不同 λ 值的位元速率	40
圖 5.3 不同 λ 對於影像品質的影響	41
圖 5.4 不同 λ 的區塊種類分布	42

圖 5.5 不同 λ 對於影像區塊大小的影響	43
圖 5.6 Singer編碼後的前景PSNR值比較圖	44
圖 5.7 Singer第 246 張影像	45
圖 5.8 Singer使用表 5.5 編碼方式的背景資料量比較	46
圖 5.9 Singer不使用固定區塊來編碼背景	47
圖 5.10 I、P、B的不同編碼方法在時間上比較	49



第一章 簡介

有鑒於多媒體的應用日漸廣泛，在生活中已經是不可以缺少的一部分，因此本文希望能提出一個方法來應用前景和背景的分析在現有 H.264/AVC 的壓縮標準上面，使得壓縮效率和壓縮品質提升，而有助於壓縮資料的儲存、傳送與利用。

於現今多媒體的發展中，視訊影像已經是被廣泛應用的資料型式，因為它具備了影音聲光效果，所以是非常適合被用來傳遞資訊的工具。再加上現在網路快速發展，因此相關的資訊傳遞應用，如：電影、視訊、數位電視等，已具有越來越重要的角色。由於儲存容量上的限制，如何於給定資料量中增加更多的多媒體資訊，已經是視訊影像壓縮的一個重要的研究課題。

由 ISO Motion Picture Experts Group (MPEG 組織) 和 ITU-T Video Coding Experts Group (VCEG 組織) 所成立的 Joint Video Team (JVT 組織)，提出的 H.264/AVC (Advanced Video Coding) (或 MPEG4 Part 10/AVC) [1][2][3][4] 標準，為本論文所要探討的對象。本論文將針對現行的視訊壓縮標準 H.264，探討前景和背景分析的應用，目的在透過前景的影像品質和主觀視覺接受度提升、背景的资料量之降低，以及壓縮速度之提升等三方面，來改善整體標準的壓縮效率。

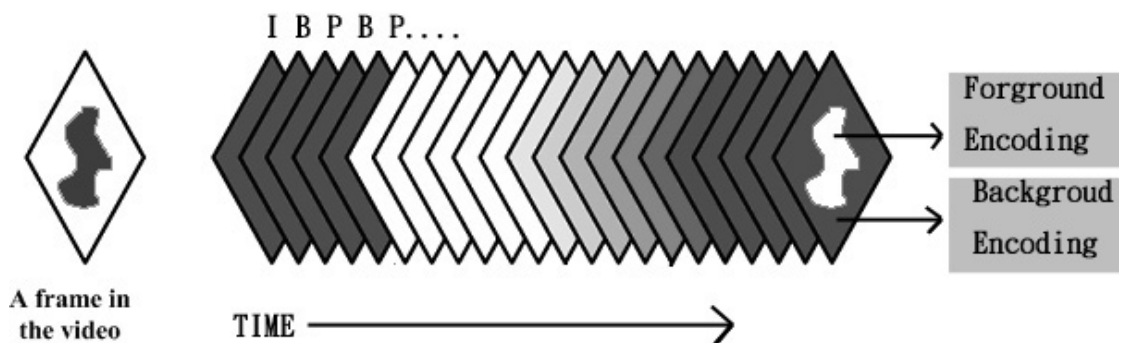


圖 1.1 影片利用前景背景編碼的示意圖

在一段影片中，我們經常可以明顯的分出來所謂的前景 (Foreground)和背景 (Background)，一般而言，前景是影片中重要的部分，可能是主要物體在畫面移動的部分，或是整段影片中，較具有代表性意義或是觀看者想要仔細注意的畫面部分；而背景主要以靜態為主，或是整段影片中，觀看者較不注意的部分。

對於整個視訊壓縮標準來說，其中過程分為 2 個部分：編碼流程 (Encoding Process) 和解碼流程 (Decoding Process)，分別用編碼器 (Encoder)和解碼器 (Decoder) 來實作，本論文主要以編碼內容來討論。圖 1.1 為我們如何利用前景和背景來編碼的示意圖，其中是把影片的內容，區分為 2 個部分後(如圖 1.2)，在編碼器中以不同的編碼流程來處理，達到不同的編碼效能和編碼品質。在本論文中，使用的前景和背景都是假設已經切割好的區塊，以作為輔助編碼壓縮的資訊，並根據標準的編碼程序和方法加以分析改善，期望得到較好的壓縮效能。本論文所考量壓縮效能標準的依據有：訊號雜訊比(SNR, Signal to Noise Ratio)、位元速率(Bitrate)、編碼速度(Coding Speed)...等。



圖 1.2 原始影像與前景和背景的切割圖

而本文對於應用前景和背景在標準內之探討，其前提是可以符合在 H.264 規格架構中的解碼過程 (Decoding process)，也就是說符合原來標準定義的語法 (Syntax)和語意(semantics)；因此經過本方法壓縮後的影片，皆可以使用在一般的

解碼器上，不會有相容性的問題。

與本文主題相關的研究之一為可調式編碼 (Rate Control Coding) [5]，這類研究主要是對整體的編碼效果和應用環境來考慮，也就是發展多種的編碼內容，再依據當時的一些條件，如：頻寬、解析度等等的要求來進行編碼控制，這類方法著重在原來標準內的編碼工具的功能調整。而另一種在 MPEG4 其中新的編碼技術 Sprite Coding，則是對於重複出現或是主要的一些畫面，如：背景圖案等，以一次編碼的方式來進行編碼。如此，在一個變動不大的環境中，就可以利用畫面間的背景可能有部分的重複部分，減少背景重複編碼的情況，以提升編碼效率。以上是相關編碼技術的研究技術，而在前景和背景的研究方面，大部分都是研究如何切割出有效或是有意義的區塊為探討主題。

正如本文前面所提，前景和背景的決定跟人的主觀意識有關，而人的主觀意識並沒有一個明確的指標可以來衡量，也造就了這方面的困難性，因此本文提出的方法，是在假設已經確立前景和背景的條件下進行。其中有關我們所提出前景和背景在編碼工具上的不同應用，是以調整可變區塊大小的方式，找出增強前景 PSNR 的區塊大小，同時以調整失真速率比參數的方式來減少背景的資料量，並且提高了編碼速度。

本論文共分成六章，下一章是 H.264 視訊壓縮標準及其壓縮原理的簡介。在第三章是討論相關前景和背景應用的一些提升壓縮效率的方法研究並和現在一些研究作比較。第四章為本論文之主要部分，其目的在探討所提出前景和背景應用的方法和實驗分析，包含了編碼過程的流程等，並且探討前述若干系統參數的影響。第五章為主要的實驗結果，包括影像和數據之分析與探討。第六章為本文章的結論，並且探討一些將來可能繼續研究的方向和應用。

第二章 H.264/AVC 視訊壓縮標準

2.1 簡介

MPEG 的全名為” Motion Picture Expert Group “，是由 IEC/ISO 在 1988 年成立的組織，主要的工作是制定標準化的數位視訊壓縮演算法。它在 1991 年提出了 MPEG-1 視訊壓縮標準[6]，主要目標是應用在位元速率 (Bitrate)最高為 1.5 Mbps (Megabits per second)的數位儲存媒體上，例如：影音光碟 (VCD)。MPEG-1 只支援非交錯式掃描 (non-interlaced scan) 及 4:2:0 的色彩取樣 (color subsampling)，並不支援一般電視的交錯式顯示。

為了能夠有效支援未來數位電視和高畫質視訊壓縮的應用，MPEG 在 1994 年完成了 MPEG-2 [7] 視訊壓縮標準的制定。MPEG-2 可說是 MPEG-1 的延伸，因為它們針對視訊編碼的基本理論相同，但 MPEG-2 改進一些 MPEG-1 的缺點，並支援交錯式掃描(interlaced scan)，和更多的色彩取樣格式，以及其它進階的編碼技術。

而到了 1998 年 MPEG 又再提出了 MPEG-4 [8] 視訊壓縮標準。基本上 MPEG-4 和先前的 MPEG-1 和 MPEG-2 是不同的系統，因此也沒有辦法相容，而 MPEG-4 除了希望應用在數位電視外，還著重在整合式互動多媒體之上。

對於資料的日漸龐大，相較於其他的視訊壓縮標準，ITU-T Video Coding Experts Group (VCEG) 希望建立一個效率加倍的標準，而提出了 H.26L。而後，在 2001 年，VCEG 和 MPEG 共同合作組織了 Joint Video Team (JVT) 並且在 2003 完成了整個新的視訊壓縮標準 H.264/AVC(或稱 MPEG-4 Part10 或是 MPEG-4 /AVC)。

在 MPEG 壓縮標準中，其實都只定義了視訊資料經壓縮後的位元串列的語法和語意，以及解碼的程序。對於編碼的過程，在標準內並沒有明確的制定，主

要可提供編碼器設計者根據編碼速度、品質和複雜度來做彈性的取捨，只要編碼後的位元串列符合 MPEG 標準定義的語法和語義即可。

接下來，我們將針對 H.264/AVC 視訊的編碼工具做簡單的介紹。

2.2 H.264/AVC 視訊編碼工具

因為新的視訊系統必須適應未來發展的網路環境，因此為了兼顧方便性和可調性，H.264/AVC 設計包含 Video Coding Layer (VCL) 和 Network Abstraction Layer(NAL)，其中 VCL 為影像處理編碼過程，而 NAL 則主要是提供 VCL 來傳輸和儲存的標準規格。本篇論文的重點是以 VCL 為主要的範疇。

相對於之前的影像編碼標準，H.264 有一些不同或是編碼效率較為提升的設計，簡述如下。



1. 可變區塊大小的運動補償

H.264 支援了較為彈性的區塊大小來作為運動補償(Motion Compensation)之用，和之前的視訊編碼標準相比，新的標準已經從只能使用固定的區塊大小 16 x 16 或 8 x 8 大小，到現在可由最大的(16 x 16 pixels)巨區塊(Macroblock)到最小的 4 x 4 區塊之間作選擇，彈性較大。其中相關內容在本篇論文第四章會有詳細介紹。

2. 1/4 像素精準的運動補償

在之前的標準大多數只支援到 1/2 精準度 (Accuracy) 像素距離的運動估測 (Motion Estimation)，在 H.264 中則是支援到 1/4 的精準度。如圖 2.1，其中整數

點為大寫的英文代號，代表的是在畫面片中實際的像素位置以 G、H、M、N 表示，且這 4 個都是影像上的實際像素點，進而可以預測出 1/2 像素精準度的像素預測位置和其灰階值，如圖中的 b、h、m、s、j，再來可以用這些所預測出來的 1/2 像素精準度的預測結果來預測 1/4 像素精準度的位置和值，如圖中的 a、c、d、e、g 等。如此便可以在計算運動補償的時候有更小的誤差值。

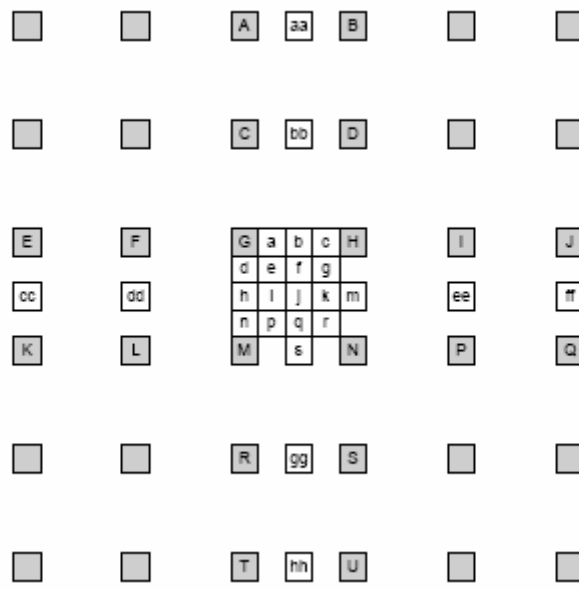


圖 2.1 1/4 像素精準度位置圖

3. 多重參考畫面的使用

在之前的標準中使用的參考畫面都只有一個，而在 H.264 中則對於 P、B 畫面使用的參考畫面都可以不只是一個，這樣可以增加畫面上的資料使用效率和品質。如圖 2.2，在圖中 Current Picture 的區塊，可以由前面 3 張不同的畫面所共同參考來組成。

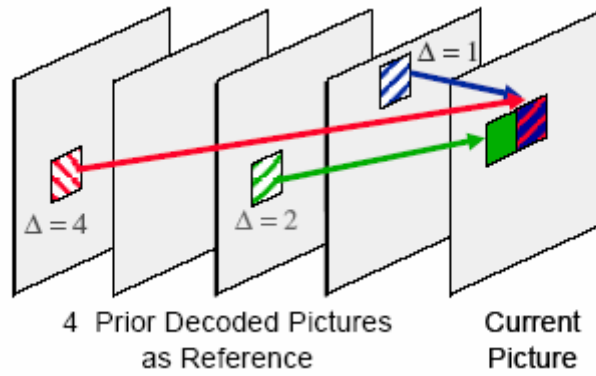


圖 2.2 多重參考畫面

4. 加權的預測

H.264 對於運動補償後所得到的預測結果，可以採用加權的方式來取得，主要可以用來消除一些影片中特殊效果，如：淡出、抹去等的特殊效果。



5. 塊狀消除濾波器 (Deblocking Filter)

基於一般視訊的壓縮標準都是以區塊為主的編碼方式，因此很容易在區塊的邊界上，產生了塊狀缺陷 (Blocking Artifacts)，主因是由於運動預測編碼和誤差編碼是分別的兩個步驟而造成。因此 H.264 在此提出了利用一個濾波器來消除這種情況。

6. 4x4 整數轉換

在之前的標準之中，主要都是以區塊 8x8 為主的轉換，而 H.264 則是採用 4x4 的轉換，除了因為區塊變小而消除的光暈 (Ringing) 現象，同時 4x4 的轉換是可逆的轉換，也就是逆轉換後可以和最初始的資料一樣，而不會有資料的損失。圖 2.3 為此 4x4 矩陣轉換的形式。

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

圖 2.3 4x4 轉換矩陣

7. 熵編碼(Entropy Coding)

H.264 中使用了 2 種 Entropy Coding，CAVLC (Context-Adaptive Variable-Length Coding) 和 CABAC(Context-Adaptive Binary Arithmetic Coding) 來使得資料量變小，主要的方法就是利用資料間的過多資訊(Redundancy)來達成資料減量，其中 CABAC 的方法複雜度較高壓縮效果也比較好。

8. 可變式的 Slice(Macroblock) 順序



為了抵抗傳輸環境的不佳所造成的資料遺失問題，H.264 使用了可變式的 Slice(Macroblock) 順序來提高資料的抗誤性。如圖 2.4，其中每個 Slice Group 都可以包含數個巨區塊，左邊的圖中，可以把重要的資料區塊分別歸類為 Slice Group #0 和 #1 剩下的資料以 Slice Group #2 來編碼，如此即可以對於不同重要性的資料加以儲存和編碼，而右邊的圖中則是為了資料的抗誤性，也就是把整個畫面分成 2 部分，間隔的分成 Slice Group #0 和 #1 如此當資料有錯誤或是遺失的時候，並不會造成資料非常不完整而提升抗誤性。

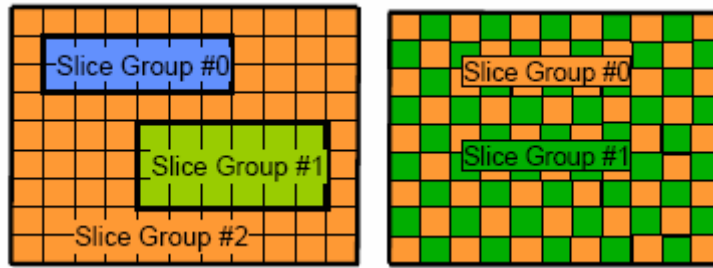


圖 2.4 FMO 的使用方式

9. SP/SI 畫面的轉換(Switching I/Switching P)

同樣是為了傳輸上的需求，主要是使解碼器可以在不同的位元速率下可以更方便的轉換。如圖 2.5， $A_0 \sim A_{11}$ 代表原本影像序列上的順序，當需要 fast-forward 的時候只需要把 A_{10} 和 A_0 產生了 A_{0-10} 後就可以達到快速轉換畫面的效果。

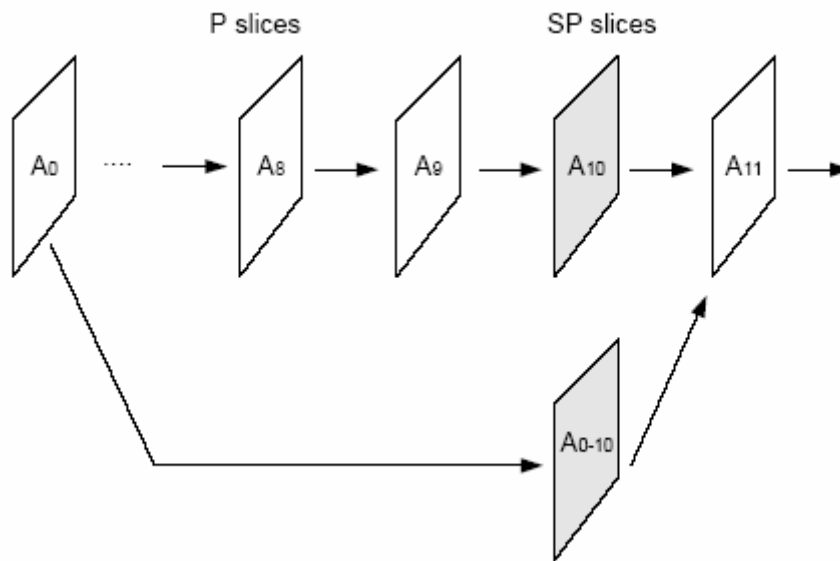


圖 2.5 SP 畫面的使用

2.3 視訊壓縮位元串列架構

標準的 MPEG 視訊可分為六個階層，如圖 2.6，由上而下依序為：Sequence、

Group of Pictures (GOP)、Picture、Slice、Macroblock(Sub-Macroblock)以及 Block。

以下就各層的功能作簡單的介紹：

1. Sequence layer

Sequence layer 是 MPEG 架構中最高階的一層，它包含了解碼時所需的參數，例如：畫面尺寸，畫面播放速率，以及數個連續的 GOP。

2. Group of pictures (GOP) layer

GOP 是由一或多張連續畫面構成，每個 GOP 內有三種不同類別的畫面，分別是 I 畫面(Intra-Coded Picture)、P 畫面(Predictive-Coded Picture)和 B 畫面(Bidirectionally Predictive-Coded Picture)。通常一個 GOP 包含一張 I 畫面，數張 P 畫面及數張 B 畫面，一個典型的 GOP 樣式為 IBBPBBPBBPBBPBB。I 畫面和 P 畫面又稱為 Anchor Frame，因為它們是用來做向前 (Forward) 或向後(Backward) 估測的基準。根據 MPEG 的定義，兩張最接近的 I 畫面之距離為 N ，也就是一個 GOP 的長度；兩張最接近的 Anchor Frame 之距離為 M ，即任兩張 Anchor Frame 中間有 $M-1$ 張 B 畫面。參數 N 和 M 的大小可由編碼器或使用者自由選擇，沒有硬性規定。一般 N 的值被設定為 12 到 15， M 的值為 1 到 3。GOP 主要是用來提供隨機播放(Random Access)的功能。圖 2.7 為一個 GOP 的示意圖， N 為 12， M 為 3。

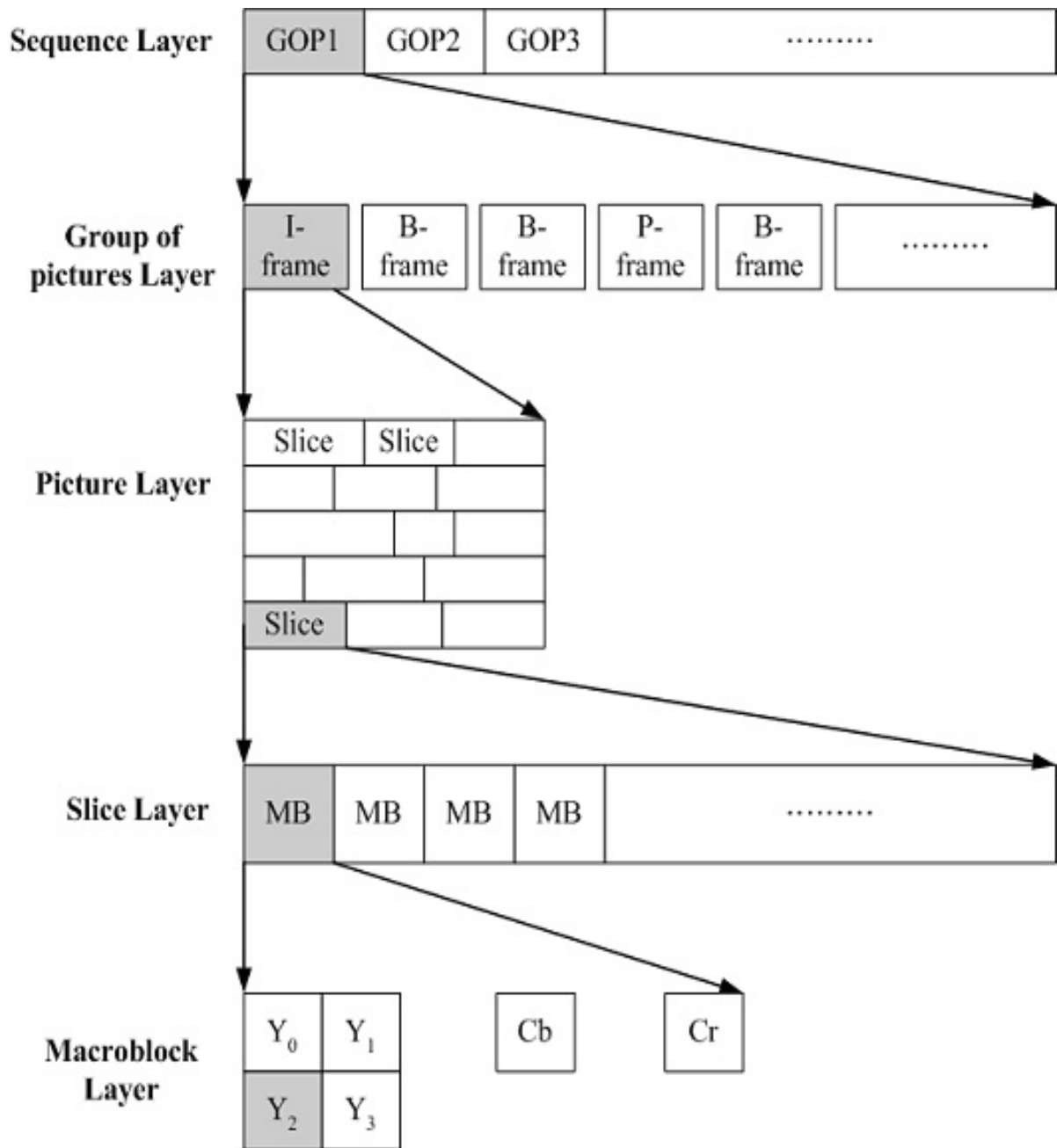


圖 2.6 MPEG 視訊位元串列架構

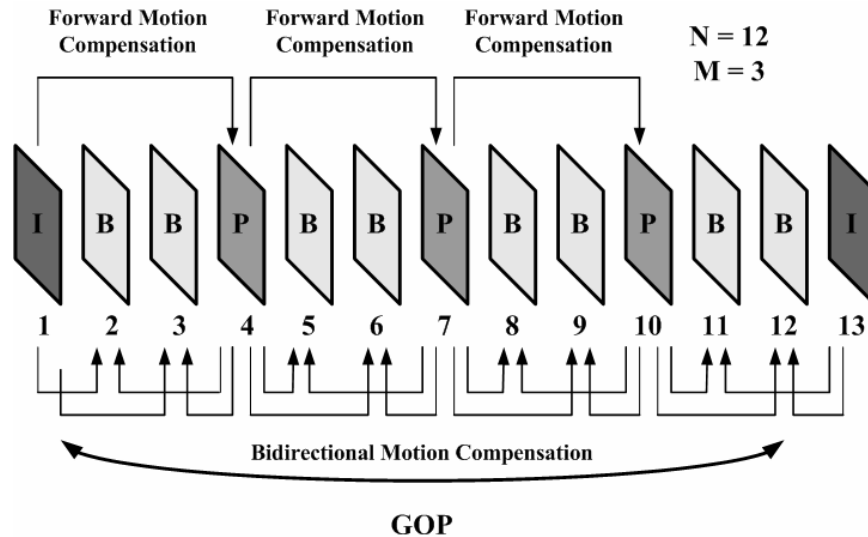


圖 2.7 GOP 示意圖。N = 12，M = 3

3. Picture layer

在 MPEG 視訊中，一張畫面是由一個 Luminance(Y) 畫面以及兩個 Chrominance(Cb、Cr) 畫面組成，其中 Y 代表亮度，Cb 和 Cr 代表顏色。依照編碼方式的不同，MPEG 定義了三種不同類型的畫面：

- I 畫面(Intra-Coded Frame，簡稱 I-Frame)

I 畫面在編碼時只利用單張影像畫面本身的資訊，先進行離散餘弦轉換及量化，再以 Entropy Coding 的方式編碼，過程中不需參考其他畫面的資訊，故稱為 Intra-Coded Frame。I 畫面可以做為視訊資料流中的索引點，提供隨機存取的能力，並在解碼時做為 P 及 B 畫面的參考影像來源。

- P 畫面(Predictive-Coded Frame)

P 畫面的編碼過程，是先以前一個 I 或 P 畫面作為參考畫面，進行運動估測和運動補償的計算，產生一個運動向量(Motion Vector)，再將餘下的誤差項做 Transform、量化及 Entropy Coding。

- B 畫面 (Bidirectionally Predictive-Coded Frame)

B 畫面的編碼過程類似 P 畫面，但是它使用的參考畫面是雙向的。B 畫面使用前面及後面的 I 或 P 或 B 畫面當作參考畫面，進行雙向的運動估測和補償，產生向前及向後的運動向量，再將餘下的誤差編碼。

由上面的敘述可知，I 畫面可以完全獨立解碼；P 畫面必須等前面的參考畫面解出之後，才能進行解碼；B 畫面則必須等前後的參考畫面都解出來後才能解碼。因此，在 MPEG 影片中，資料存放的順序與實際播放的順序有所不同。舉例來說，若 MPEG 影片播放的順序為：

I(1) B(2) B(3) P(4) B(5) B(6) P(7) B(8) B(9) I(10) B(11) B(12) P(13)

則存放在 MPEG 視訊串流內的順序為：

I(1) P(4) B(2) B(3) P(7) B(5) B(6) I(10) B(8) B(9) P(13) B(11) B(12)

4. Slice layer



Slice 是由一連串任意數目的巨區塊組成。每個 Slice 至少有一個巨區塊，且各 Slice 包含的巨區塊不可重疊。在編碼時，每一張畫面被切割成數個 Slice，分開來解碼，可以避免解碼時產生的錯誤影響到整張畫面。Slice 的長度由編碼器決定。

5. Macroblock layer

Macroblock layer (簡稱 MB layer) 是 MPEG 串流中最重要的一層，MB 是進行運動估測和運動補償的基本單位。在畫面上，MB 是一個 16*16 大小的大區塊，根據不同的色彩取樣格式，一個 MB 內所包含的區塊個數亦不同。在 MPEG 中(4:2:0 格式)，一個 MB 有 4 個 Y、1 個 Cb 和 1 個 Cr 總共 6 個區塊。

第三章 利用前景與背景來提升壓縮效能的 探討與相關研究

我們希望在視訊影片的壓縮過程中，能針對影片中前景與背景區域之特性，進行不同方式的壓縮。在符合 H.264 的解碼規格下，針對前景區域，作較高畫質之視訊壓縮，而對於背景部分，則減小背景視訊的大小，使得視訊壓縮能夠更具彈性。因此，前景和背景的區分為第一步。

在視訊影片的前景與背景分割研究中，可分為影片場景固定與場景不固定兩類。其中，場景固定的視訊影片，常見於一般的監控系統中，這類視訊，多以固定式攝影機進行拍攝，針對特定場景，作背景估計與前景物體的切割。在這一類的研究中，由於是針對特定的場景進行監控，因此可以預先估計該場景之背景模型；而後，當有前景物體移動、變化時，便可即時的偵測出變化區域，進一步作前景物體的即時追蹤。這方面的研究通常都是利用影像的差異，如：影像相減等，得到一些靜態的背景畫面，在由所得到個背景模型來對監控系統的操控。

其中由 C. Stauffer and W. Grimson [9] 所提到的，是以像素為單位不斷的更新這些背景模型 (Background Model)，他針對每一個像素以混合高斯分布 (Mixture of Gaussians) 來做分類。他所提出的發法可以處理光線的變化、重複的動作甚至是長時間下畫面的改變。

對於另一類場景不固定的視訊影片，其背景與前景物體的分割研究，則屬於視訊層分析 (Video Layer Analysis) 的範疇。在視訊層分析的研究中，通常以視訊影像內容之顏色、紋理 (Textures)、運動模式等時間與空間上 (Spatial and Temporal) 之特性，作為分析依據，透過視訊群集 (Video Grouping) 的方式，將影片中相同性質的視訊物體 (Video Objects)，切割為同一層。而這方面的研究通

常都是以最佳化 (Optimization) 為方法，對整個視訊影片進行運算，找出相同模式的視訊階層。

一般來說通常在影片切割 (Video Segmentation) 方面，通常都是利用影像中的運動、整體動作或是彩色影像資訊來做切割。其中由 S. Khan and M. Shah [11] 所提出的，以最大化事後機率 (Maximum a Posteriori Probability MAP) 的方法，將影片中相似的位置、顏色、運動等，判斷出適合的部分作群集。

另一些視訊層分析的研究，則與視訊編碼壓縮的過程作配合，如分析編碼計算時，依運動向量 (Motion Vector) 的性質，切割出不同的視訊層。這類分析方法，多同時考慮了視訊層計算效率的問題，因此採用較簡單的計算方法，以便快速的切割出視訊層。在 Y. Lu, W. Gao, and F. Wu [12] 中所提到的 Sprite Generation，定義了在壓縮過程中，建置的背景的方法；在 Sprite Coding 之前，必須先把移動的物體分割出來，因為現在沒有很有效的切割方式，所以在此方法目前是把以經前處理過的影片來編碼，對於已經切割處理過的影像使用混合的方式把已經切割出的背景部分建置出來，如果所得到的畫面是很清晰沒有模糊，那在那些被切割出來畫面部分，也就會有較好的接合效果。

在 H.264 中可變區塊的部分，實驗中主要先參考 Mathias Wien 所提出 ABT [13] 的方法，其中他提出了 Adaptive Block-Size Transforms (ABT)，他最主要的觀念就是把預測所產生的預測誤差所使用轉換的區塊大小，以及運動補償所使用的區塊大小一致化。而當使用 ABT 編碼後，減少了 12% 的資料量、增加了 0.9db 的 PSNR 值，因此我們考慮在區塊大小的使用上，可以細部的再分析過。

藉由上面的一些相關研究，由於切割出有效的前景和背景區塊有其複雜度，因此實驗中也將利用假設已經分割完畢的前景和背景區塊來作為編碼依據，利用現今視訊壓縮標準希望能把標準 H.264 的訊號雜訊比 (PSNR)、位元素率 (Biterate)、編碼速度 (Coding Speed) 等的編碼效能提升。

第四章 可變區塊大小的方法

在此章將說明在現行 H.264/AVC 的標準與早期的 MPEG-2/4 標準，對於 Intra Prediction、運動估計(Motion Estimation)中所使用的區塊大小 (Block Size) [15] 之不同的定義。

表 4.1 MPEG2/4 和 H.264 所使用區塊大小的比較

Algorithm Characteristic	MPEG2/MPEG4	H.264
Intra Prediction (Spatial)	Fixed	Multi-direction, Multi-mode(4x4, 16x16)
Motion Estimation Blocks (Temporal)	16x16,8x8	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4,skip/direct

對於新標準 H.264 和之前標準所定義的區塊大小比較可由表 4.1 得知，以 Intra Prediction 而言，H.264 除了提供 2 種區塊大小以外並且還有較多的預測方向，而對於 Motion Estimation 的區塊大小而言，H.264 更支援到 7 種不同大小和 2 種特殊模式，因此在這方面的選擇，H.264 有著較大的彈性空間。

而標準中所提供這些的編碼工具(Coding Tools)，通常是指所使用到編碼方法的過程(如：Intra Prediction 可以當成是一種 Coding Tool)，將會是後面我們所要使用的部分。另外，在 H.264 中也引用了 Rate-Distortion Optimization (RDO)的觀念，本章節也將會討論 RDO 和前景背景的影響。

圖 4.1 為部分的編碼流程圖，輸入的影片(Input Sequence)會先分成 2 個部分，一個為 Intra Prediction，另一個為運動估測(Motion Estimation)，而經過運動估測的會再做運動補償(Motion Compensation)，而兩者都會經過轉換，再進入下一步編碼。

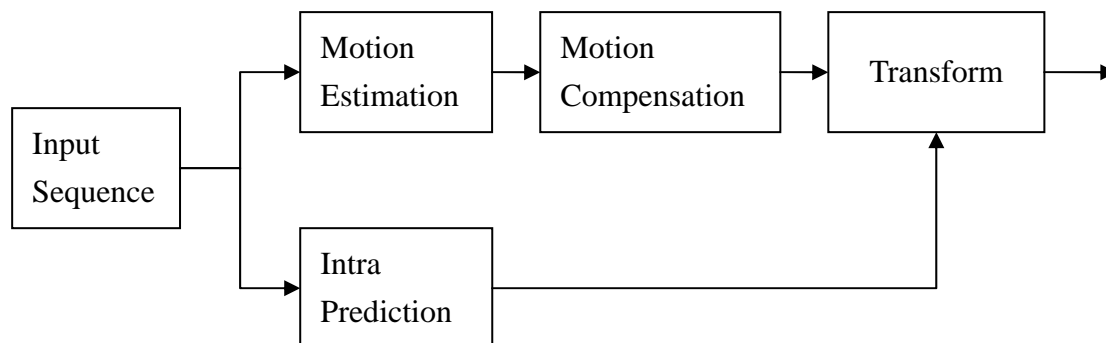


圖 4.1 編碼流程示意圖

4.1 可變區塊大小的種類



在編碼的過程中，一個巨區塊為編碼的重要基本單位，大小為 16×16 (pixels) 的大小，因此每一個畫面都會被區分成為 $N \times M$ 個的巨區塊 (N, M 為正整數)，而在 H.264 中每一個巨區塊都可以再細分成數個區塊 (Block)，分別把每一個區塊都經過編碼器編碼。

4.1.1 畫面內預測 (Intra Prediction)

當一個巨區塊或區塊被以 Intra Mode 來編碼的時候，必須根據此區塊附近所編碼後再重建的區塊為依據，建立出預測區塊，再把此預測區塊和原始的區塊相

減，可以求得誤差值再進行編碼。對於亮度(Luminance)的部分，(圖 4.2)可以把巨區塊分成 16 個 4x4 區塊 或是 1 個 16x16 的巨區塊。4x4 的 Block 總共有 9 種的預測方向，16x16 的巨區塊總共有 4 種的預測方向。

在 H.264 標準中，I (Intra)畫面中，所有的巨區塊都必須以上述的方法編碼而成。通常經過 Intra Prediction 的巨區塊所使用的資料量會比較大，但相對的壓縮的影像也會較為接近原始影像。

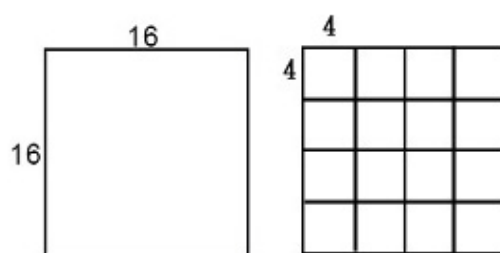


圖 4.2 Macroblock: 16x16, 4x4, 2 種區塊大小



4.1.2 運動估測(Motion Estimation)

當一個畫面在編碼的時候，需要使用到時間序列在前面或後面的畫面來編碼時，稱為畫面間的預測(Inter Prediction)並且必須做運動估測找出運動向量(Motion Vector)。在 H.264 之中主要支援運動估計的區塊大小範圍由 16x16 到 4x4(Luminance)，每一個巨區塊 (16x16 pixels)可以分成 4 種，如圖 4.3，而其中 8x8 的 Mode 又可以再細分成 4 種，如圖 4.4 所示，如此每一個巨區塊，總共有 7 種不同區塊大小，因此將會增加非常多種的組合可能。

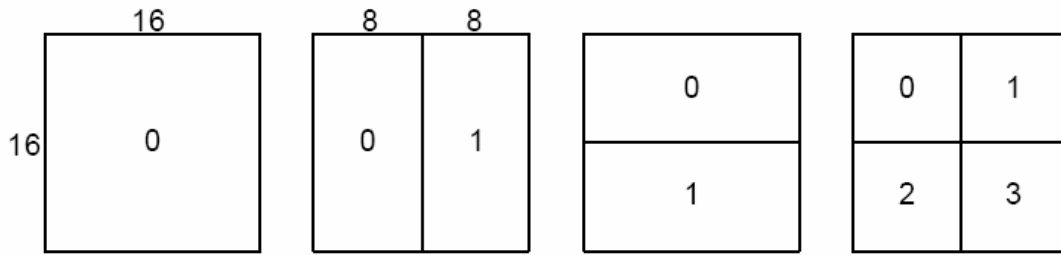


圖 4.3 Macroblock : 16x16, 8x16, 16x8, 8x8, 4 種區塊大小

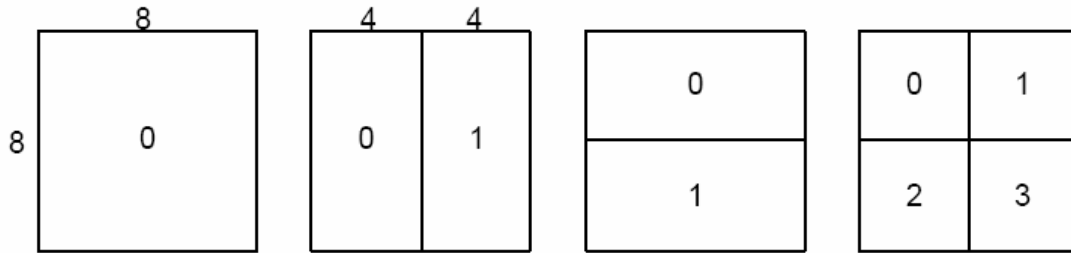


圖 4.4 8x8 block : 8x8, 4x8, 8x4, 4x4, 4 種區塊大小

對於上面所分出來的每一個區塊，都必須要計算出一個運動向量 (Motion Vector)，而每一個運動向量和區塊的種類也都需要被編碼後傳送。如果選擇一個較大的區塊(如: 16x16, 16x8, 8x16)，則所需要的運動向量的和區塊的種類資料量會比較少，但卻也因為較多細節的畫面經過運動補償(Motion Compensation)之後剩下的預測誤差(Prediction Error)資料量會比較多。如選擇一個較小的區塊(如: 8x4, 4x4 等)，雖然可以在運動補償過後，使得預測誤差資料量變小，但是因為區塊變小而會增加的運動向量和區塊種類的資料量，而這些都會影響到資料壓縮的效果。一般來說，較平滑的畫面通常用較大的區塊而較多細節部分的畫面則用較小的區塊整體效能會比較好。

在 H.264 標準中，P(Prediction)和 B(Bi-Prediction)畫面，是屬於 Inter Prediction，除了上述了區塊大小，也可以使用 4.1.1 的 Intra Prediction 的 2 種區塊大小，所使用的方法是一樣的，另外 P 畫面還有一種區塊模式，當一個巨區塊沒有編碼後剩下的預測誤差需要傳送，並且也不需要傳送運動向量的資訊時候，叫做 Skip Mode，此種類在解碼的過程中，並不使用到運動向量來表示預測後的結果。另

外，在 B 畫面之中，當一個巨區塊 (或 8x8 區塊)種類是利用參考畫面 (Reference Frame)裡相對位置的巨區塊(Co-located Macroblock)所產生的時候，此區塊的運動向量則是依據和參考畫面的時間距離的比率 (Temporal Distance Scaling)來產生，叫做 Direct Mode。表 4.2 為對 I、P、B 畫面使用所有區塊大小的整理。

表 4.2 對於 I、P、B 畫面所可以使用到的巨區塊(區塊)的種類

Frame Type	Skip	Direct	16x16	16x8	8x16	8x8	8x4	4x8	4x4	Intra16x16	Intra4x4
I										√	√
P	√		√	√	√	√	√	√	√	√	√
B		√	√	√	√	√	√	√	√	√	√



4.2 固定區塊大小的分析

由前面所知所有區塊的種類後，先以固定區塊大小的做法來實驗，也就是對於每一張的畫面都採用其中某固定一種的區塊大小，而使用的影片序列格式為：

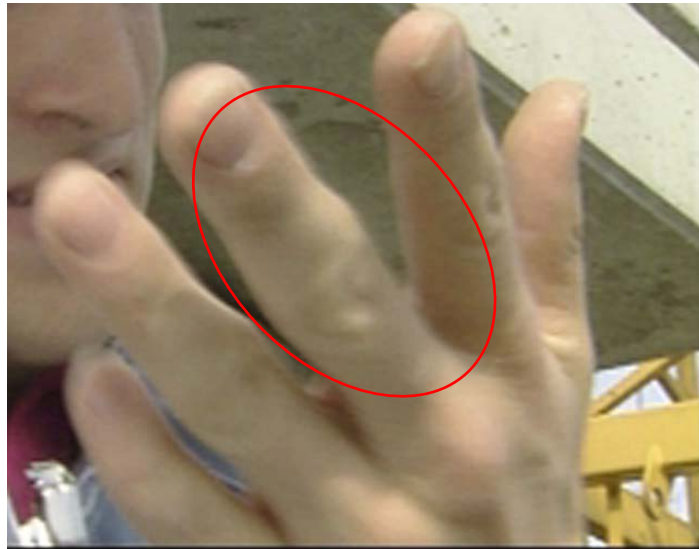
1. 影片序列：IIII

表 4.3 影片序列區塊的 PSNR 和 bitrate 比較

Frame Type	Block-Size	Bitrate(Kbit/s)	PSNR(db)
III	16x16	3047.4	37.05
III	4x4	2543.3	37.17

在表 4.3 中，主要是比較一整段影片，全部以 I 畫面來編碼時，整個畫面均使用區塊大小為 16x16 或是區塊大小為 4x4 的區塊，由表可知，在一般的情況

下，越小的區塊會使得畫面品質提升。如圖 4.5(a) 為原始影像，(b) 和(c)分別為使用 16x16 和 4x4 區塊為編碼方式的結果，在使用 16x16 區塊的紅色範圍中，手指的邊緣明顯的鋸齒狀，而在使用 4x4 的圖中，則和原始影像較為接近。



(a) 原始影像



(b) 區塊大小為 16x16



(c) 區塊大小為 4x4

圖 4.5 使用 foreman 影像結果的比較

2. 影片序列 : I B P B P

這裡沒有考慮 P 畫面的 Skip 模式，主要是因為一般影片來說，除非是固定

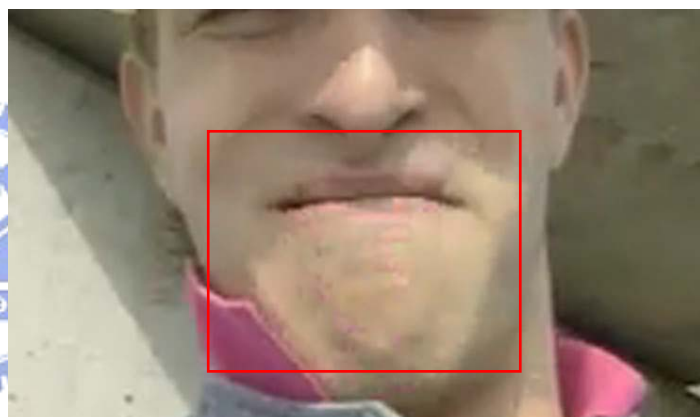
畫面拍攝，畫面上所有東西都靜止不動(或趨近於靜止)，不然此種區塊模式通常很少出現。

表 4.4 影片序列區塊的 PSNR 和 bitrate 比較

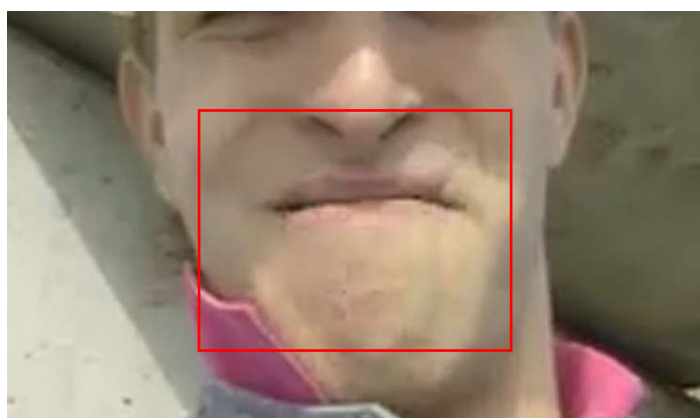
Frame Type	P Block-Mode	Average P Bitrate	PSNR
IBPBPB	16x16	639	36.13
IBPBPB	16x8	688	36.29
IBPBPB	8x16	679	36.32
IBPBPB	8x8	764	36.38
IBPBPB	8x4	977	36.37
IBPBPB	4x8	983	36.37
IBPBPB	4x4	1301	36.28



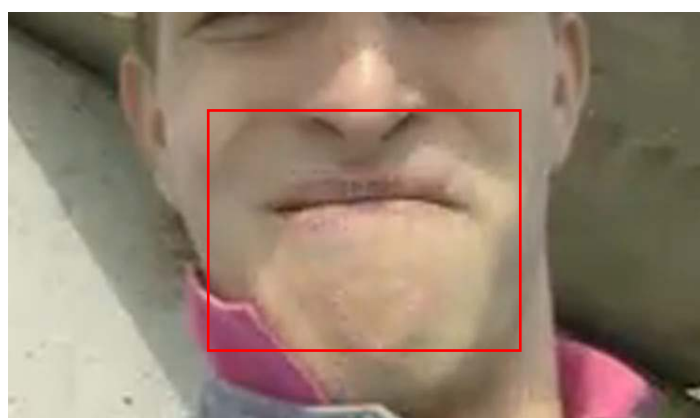
(a) 原始影像



(b) 4x4 區塊



(c) 8x8 區塊



(d) 16x16 區塊

圖 4.6 P 畫面使用不同區塊大小影像結果的比較

表 4.4 為比較在影片中對於 P 畫面使用不同大小的區塊來壓縮產生的效果，可以看到畫面因為區塊越小數量越多也就使得資料量提升，而在 PSNR 則並無太明顯的差別，但是在以圖來看，圖 4.6 (a) 為原始影像，圖 (b)、(c)、(d) 分別為使用 4x4、8x8、16x16 區塊大小為編碼方式的結果。可以發現在嘴唇和下巴的部分當區塊越大的時候，皮膚的皺紋會逐漸消失，由此可看出區塊大小為 4x4 時候和原始影像是最近的。這裡使用的影像片段為 Foreman (MPEG4 測試影片)，CIF 格式，30fps

4.3 前景(Foreground) 和背景(Background)的使用

這個章節，我們將介紹如何應用前景和背景的觀念，並且使用假設已經切割好的前景和背景在標準之中。



4.3.1 前景和背景的使用觀念和流程

一般而言，觀看者在看一段影片的時候，可能只有某些事件的發生或是物件的主要動作會被觀看者特別注意，因此能有效的將這些區域區分出來，並加以用不同方式去壓縮，就可以達到重點區域的品質提升，而整體所耗費的資源下降，如此一來提升了非常多的壓縮品質和壓縮效率。圖 4.7 (a)為一般視訊編碼方式，一整段的影片當畫面一張張的進入編碼器編碼，而在 4.7 (b)中為本篇論提出的想法，藉由上面的原來影像和下面的前景和背景分割的 2 元黑白影像，一起進入編碼器編碼，再由編碼器內的修改來達到效率的提升。

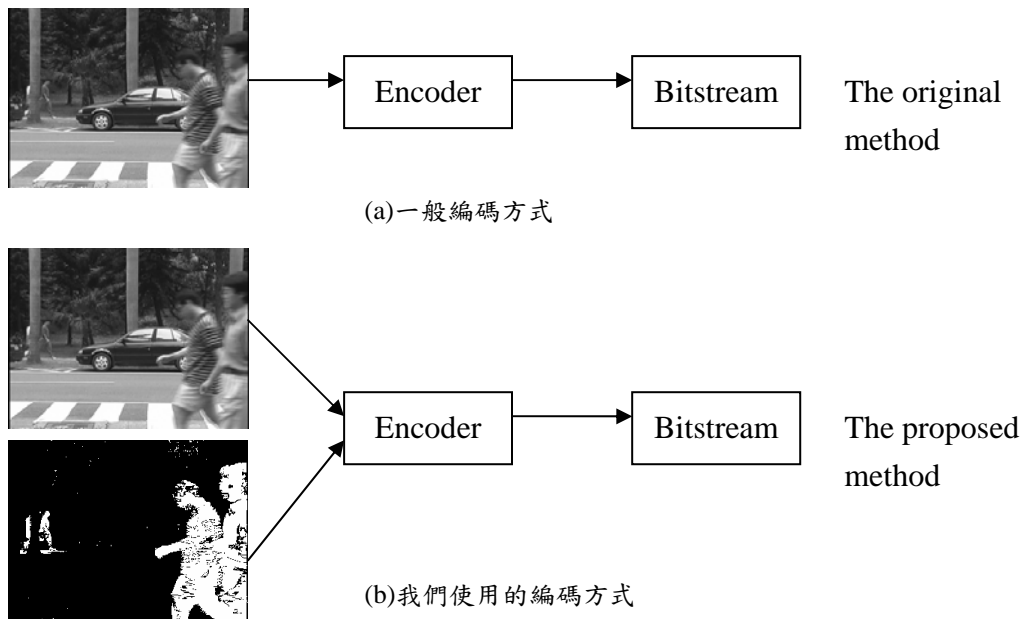


圖 4.7 前景和背景應用示意圖



在這裡主要要討論前景和背景區塊編碼的的使用，圖 4.8 中 Encoding Macroblock 為巨區塊編碼的開始，其單位大小為整個編碼過程的主要單位，一開始對於 I、P、B 畫面分類，當區塊種類決定之後，便會經過 DCT 算出預測誤差，再經過轉換編碼後即可進入下一步。而在 P 和 B 畫面的部分，首先會先切分出 4.1.2 節所說運動估測的區塊大小後，而後利用運動估測算出運動向量，進而決定使用的區塊大小再做運動補償，並且再和 I 區塊種的的預測方式作比較，如果已經是是最好的預測方式則會直接經過轉換編碼，若非，則會以 I 畫面的區塊種類繼續預測。

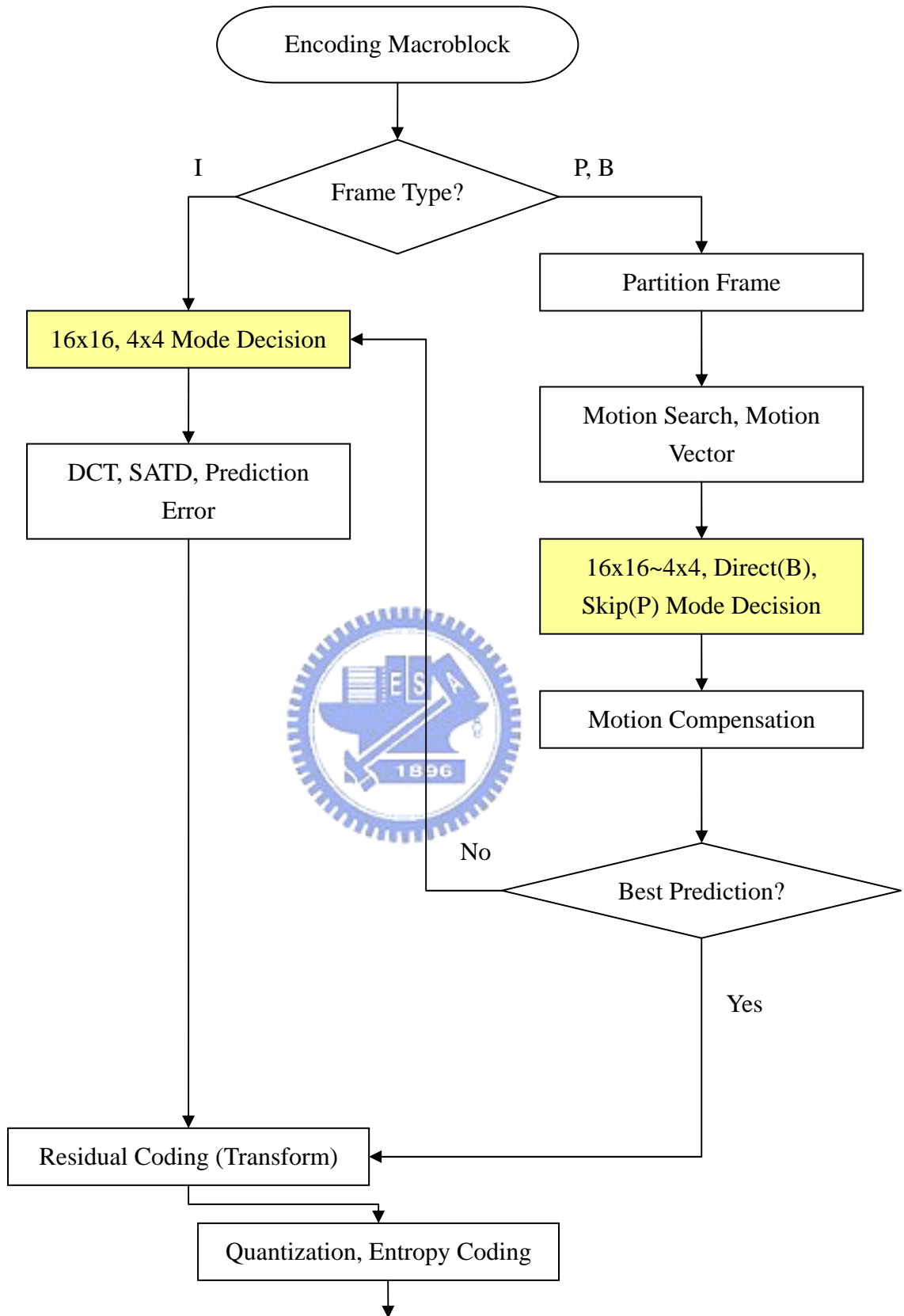


圖 4.8 區塊選擇流程圖

4.3.2 前景和背景區塊種類的比較

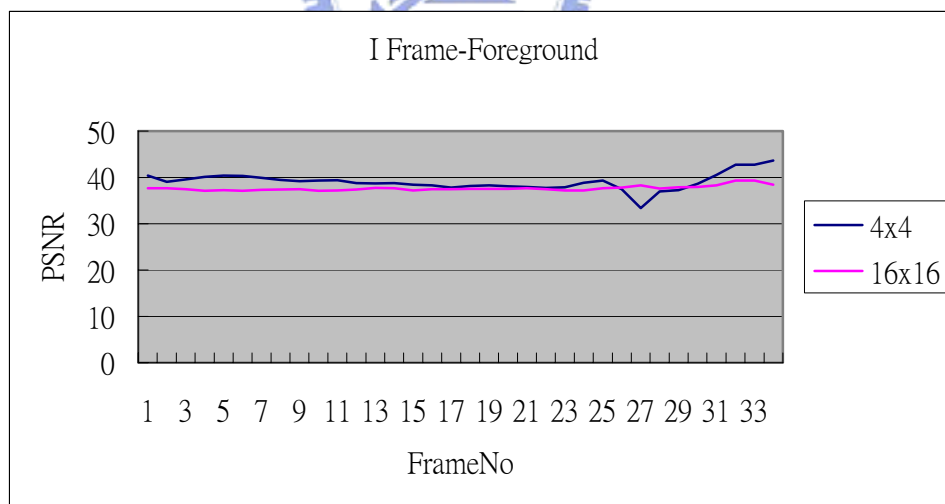
這部份分別對於整段影片中，I、P、B 畫面中前景採用不同區塊大小來做比較，而影像序列為 I B P B P。

I → 4x4, 16x16

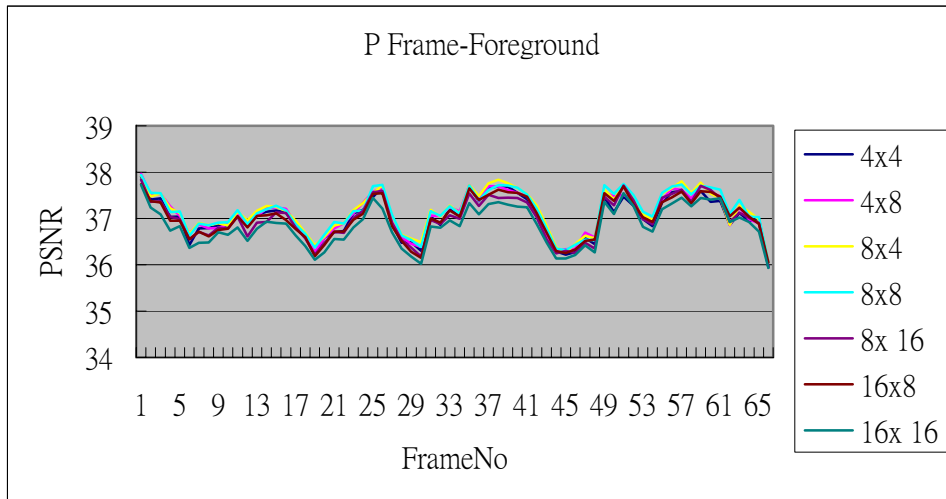
P → 4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16, skip

B → 4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16, direct

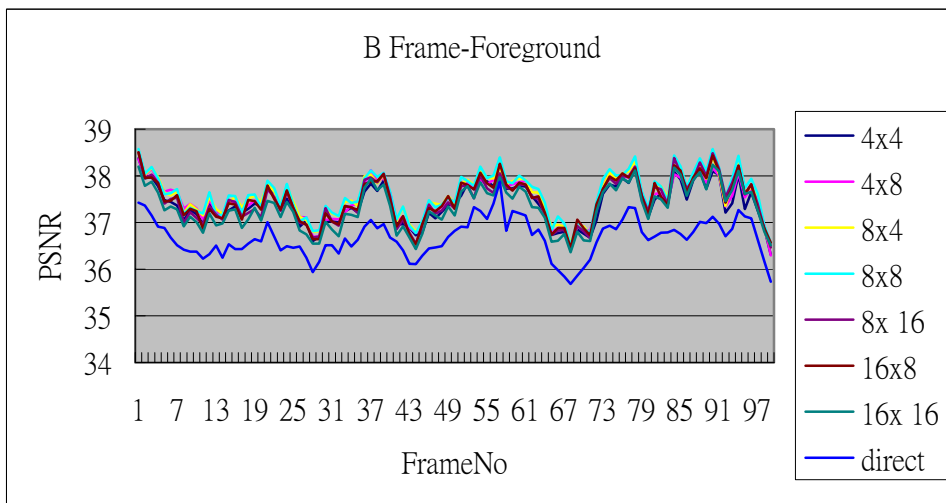
其中所有區塊的種類如上所示，在此我們以其中幾種來作為測試的依據，圖 4.9 為 I、P、B 用上述所說的區塊種類分別對於前景和背景的編碼，發現在 I 畫面(a) 中差距大約 1.5 db 值而已，而在 P、B 畫面(b)、(c)則是沒有明顯的一致性，



(a)



(b)



(c)

圖 4.9 I、P、B 應用於不同前景的區塊種類

其中相差的 PSNR 值大約都在 0.1~0.5 db 之間，但是使用 16x16 區塊大小的效果最為不佳。在這個部分沒有討論到 P 畫面的 Skip 種類，因為這種區塊種類，並沒有空間和時間上的預測編碼，而只單純的對於完全靜態的畫面才會使用到。另外，在 B 畫面上使用 Direct 區塊種類的 PSNR 值較其他 7 種區塊大小的 PSNR 值低。

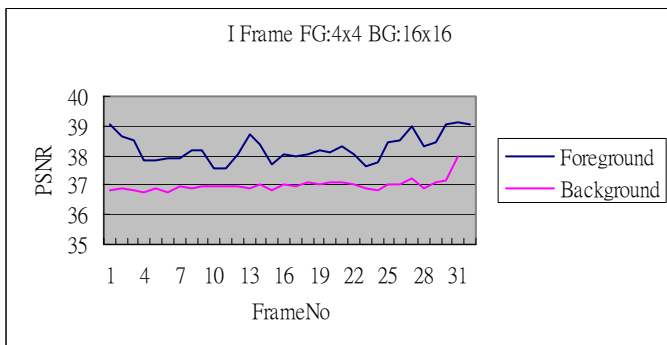
對於前面結果可知，區塊的選擇沒有對於 PSNR 值有非常明顯的差異，在這

部分我們改成把在同一段影像中的前景和背景互相比較，不過如果對於 I、P、B 畫面中前景和背景使用的區塊大小都加以組合會過於複雜，因此在實驗後選擇數種的比較結果來觀察。

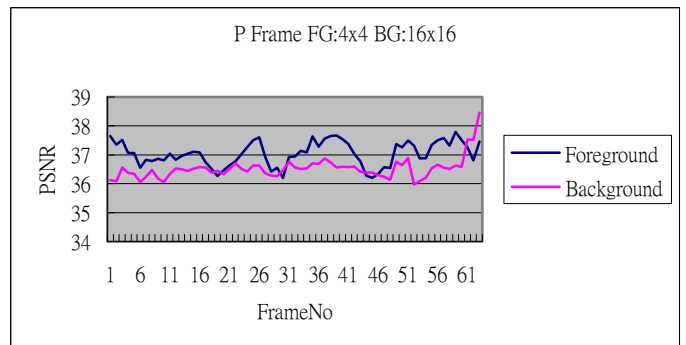
表 4.5 前景和背景區塊不同的 PSNR 和 bitrate 比較

Frame Type	Foreground	Bitrate(Kbit/s)	PSNR(db)	Background	Bitrate(Kbit/s)	PSNR(db)
I	4x4	596	38.25	16x16	2203	36.48
P	4x4	486	37.04	16x16	341	35.64
P	8x8	313	37.16	16x16	342	35.63
P	4x4	486	37.06	8x8	455	35.84
B	16x16	156	37.38	direct	665	35.38
B	4x4	385	37.44	16x16	200	36.09
B	4x4	390	37.39	direct	671	35.44
B	8x8	229	37.59	16x16	200	36.08

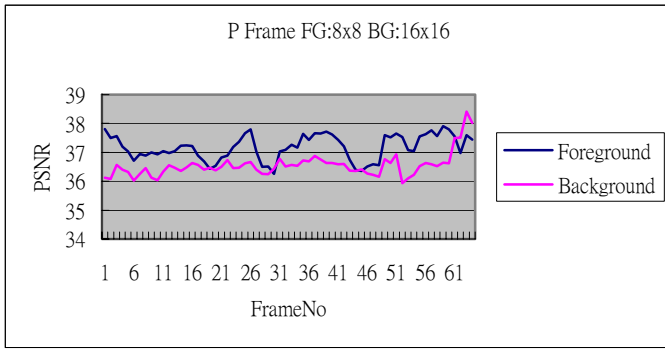
在此部分先利用圖 4.7 (b)所分割出來的前景和背景區塊來做處理，前景 (Foreground) 為灰階值為 255 的點，背景 (Background) 為灰階值為 0 的點，因為這些分割好的畫面區塊，都是像素 (Pixels) 為單位，但是在 H.264 中，編碼的最小參考依據是以一個巨區塊為基本單位，因此必須把這些 2 元影像的畫面先切割成 $N \times M$ 的巨區塊大小 (N 、 M 為影像尺寸長和寬除以巨區塊長度 16)，當每個被切割的巨區塊只要有一個像素點是屬於前景的部分，就把整個巨區塊都歸類為前



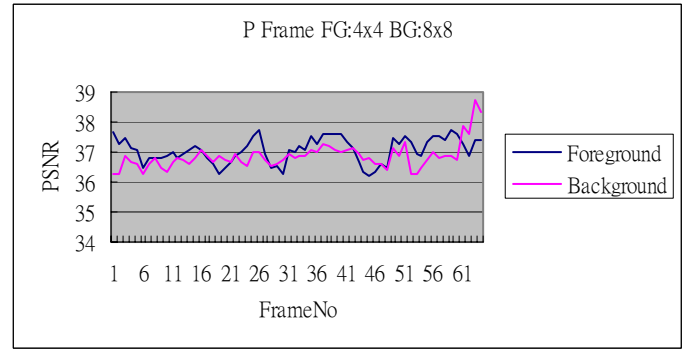
(a)



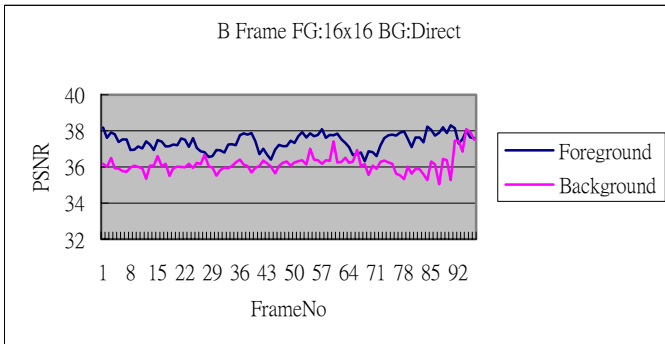
(b)



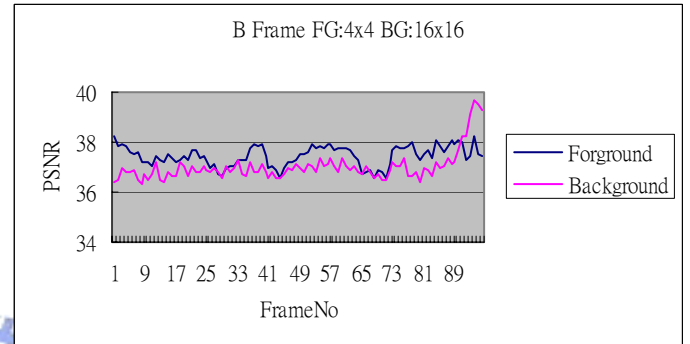
(c)



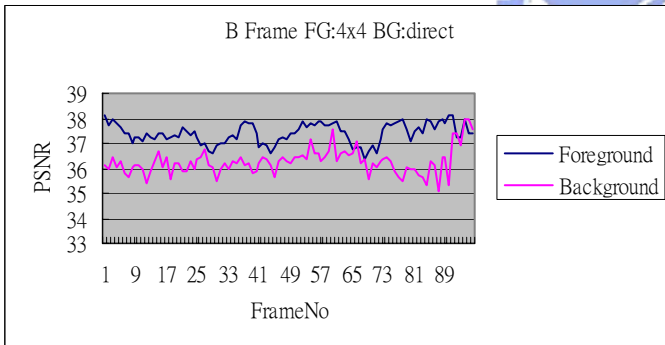
(d)



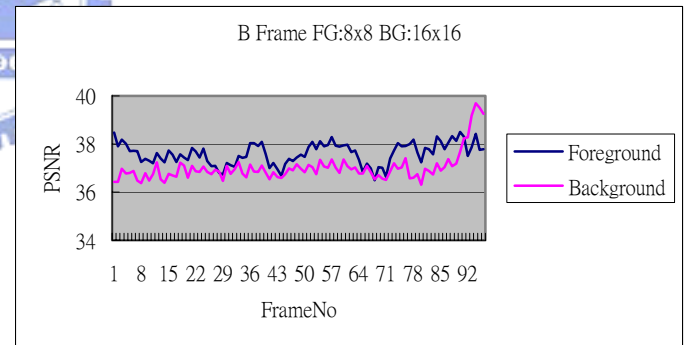
(e)



(f)



(g)



(h)


圖 4.10 表 4.5 前景和背景的 PSNR 值比較

景來編碼，如此便可以把整張 2 元影像來符合巨區塊編碼的規定，表 4.5 就是對於所有切割好的影像，分成 IPB 畫面的前景和背景，並且以不同大小的區塊去加以編碼，表中所表示的為畫面平均的位元速率、PSNR 的標準值。而圖 4.10 是把表 4.5 影片以連續的 I、P、B 來區分，4.10 (a) 為 I 畫面，(b)、(c)、(d) 則為 P 畫面，(e)、(f)、(g)、(h) 為 B 畫面，主要可以看在 I、P、B 中前景部分所使用的

區塊比背景有更好的 PSNR 值；但是前景以不同的區塊編碼，畫面品質卻很接近，因為影像是一個變動的性質，造成編碼時所需的最佳區塊大小與影像的內容有蠻大的相關性。而對於整體的編碼資料量大小，並沒有很好的效能提升，可以看出當區塊變小的時候，資料量都會提升，但是即使使用最大的區塊來編碼，資料量也無法比標準編碼方式小。因此使用選擇的區塊大小來編碼的方法，對於不同的影片所需要的方式有可能會有所不同。

可以觀察到在表 4.5 中，前景和背景的部分大部分都有 1~2 (db)的影像品質差異，對於一般影像品質，屬於肉眼即可分出兩者之間差異的範圍。

4.3.3 利用 Rate-Distortion 分別處理前景和背景



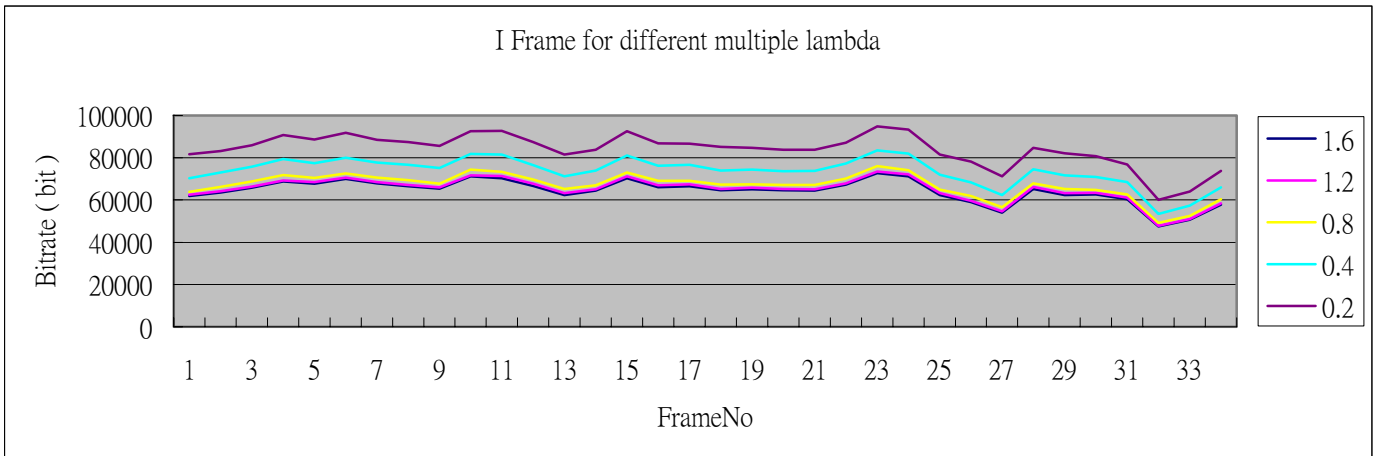
在 H.264 標準中，可以引用 Rate-Distortion Optimization (RDO) [14]來做為 Intra Prediction、Motion Estimation 的預測方法，基本的想法為利用影像的品質還有位元速率來取一個平衡，由編碼後重建的影像模糊(Distortion)和資料在編碼儲存的大小來實做。在前一個章節，由於使用固定的區塊大小沒有辦法解決資料量的問題，因此使用 RDO 來分析是否可以針對資料量的問題來解決。

在 RDO 的過程中，所針對的 Rate 和 Distortion 的部份，以下列方程式為主要依據：

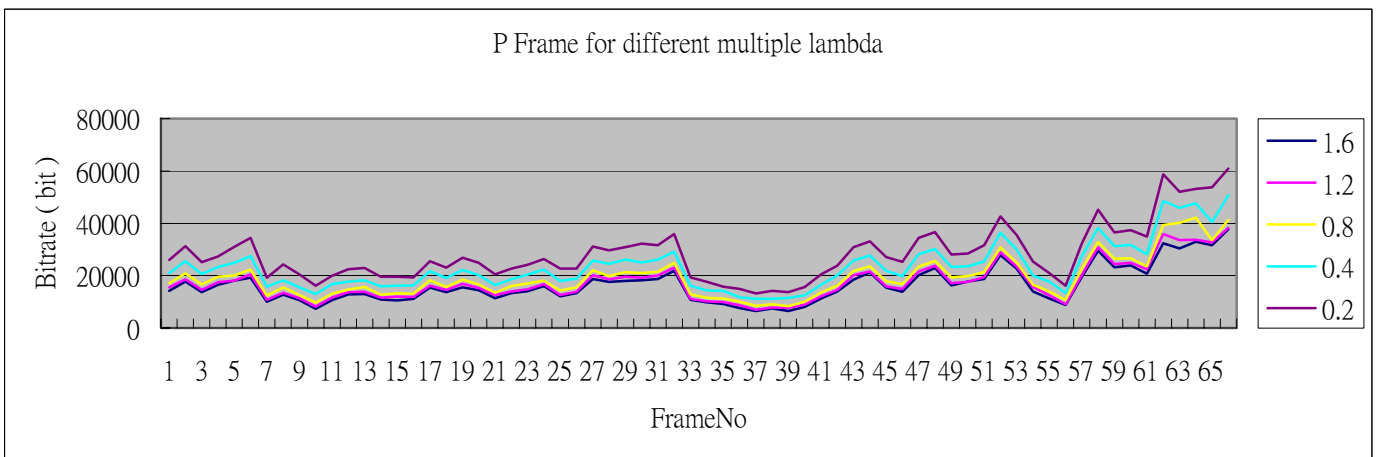
$$Cost = Distortion + \lambda * Rate$$

其中Cost為此巨區塊編碼方式的代價， λ 部分是一個比例常數，藉由這個比例常數調整模糊率，便可以對於畫面的品質和位元速率來做一個控制，在這裡我們使用在I、P畫面中 $\lambda_0 = 27.4$ 、B畫面中 $\lambda_0 = 73.1$ （此時的I、P、B畫面使用的Quantization Parameter(QP)都為 28），在此部分主要是對於壓縮後的位元速率來

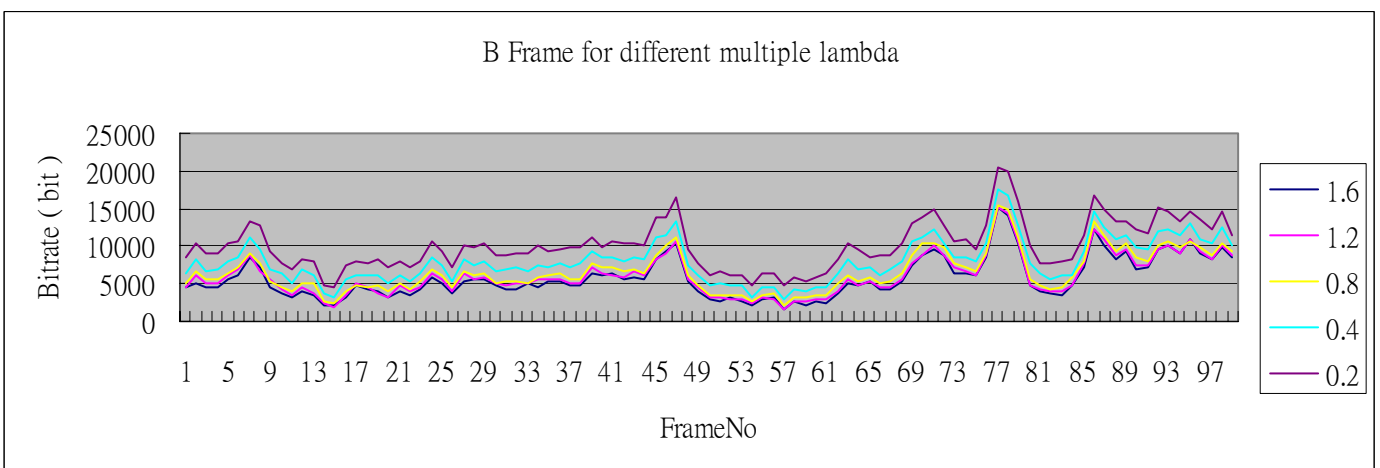
做主要考慮，因此採用 $\lambda=1.6\lambda_0$ 、 $1.2\lambda_0$ 、 $0.8\lambda_0$ 、 $0.4\lambda_0$ 、 $0.2\lambda_0$ 。



(a)



(b)



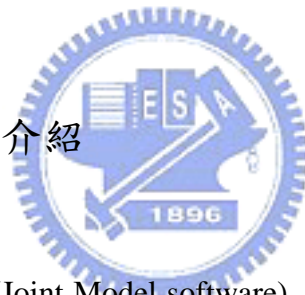
(c)

圖 4.11 整段影片以不同 λ 值的資料量比較

圖 4.11 為不同的 λ 分別對於整體編碼位元速率的影響，圖(a)為使用I畫面後位元速率的影響，由圖可知，使用 $1.6\lambda_0$ 的位元速率大致為使用 $0.2\lambda_0$ 的位元速率的 75%，而在圖(b)和圖(c)則分別為P畫面和B畫面在使用了不同的 λ 所減少的資料量，P和B畫面減少的比率平均在 55%。實驗的結果發現，I畫面使用RDO的參數調整後，減少的資料量最明顯，P和B次之。

上述的方法達到了資料減量，因此可以對於前景和背景的應用相結合，使得背景資料量的減少，再依據方程式的定義可以發現，當 λ 值變小的時候，也會稍微增加影像品質的 PSNR 值，因此也可以把前景編碼和較低的 λ 組合，提升畫面品質。

4.4 codec 相關參數的介紹



對於目前所使用的 JM (Joint Model software) 裡面部分相關的參數設定，以及所影響的壓縮品質和壓縮率的結果來做進一步的分析，相關參數以 JM 裡面所訂定 4 個參數來討論，包括 QP(quantization parameter)、Hadamard、Reference Frame Number、Entropy Coding 參數設定。其中 QP 為量化的單位大小，一般來說，用大的 QP 就是希望能有較高的壓縮率；而 Hadamard 為 H.264 中提出的特殊矩陣，用來減少轉換間的資料誤差；而 Reference Frame Number 則是因為 H.264 對於參考的畫面張數提供的彈性較大，所以我們希望能在這部分也在前景和背景隻間有一個不同的編碼方式；再來就是 Entropy Coding，因為 H.264 提供了 CAVLC 和 CABAC 可以選擇，主要也會影響到影像編碼的品質和壓縮率。

4.4.1 QP 不同的影響

這裡所說的 QP 為在 H.264 中一個參數的設定，它包含了在整個編碼過程中，量化程度的大小，如表 4.6 列。QP 為在編碼器所設定的參數，皆為整數，而 QStep

表 4.6 QP 量化對應值

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	...
QStep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	...
QP	...	18	...	24	...	30	...	36	...	42	...	48	...	51
QStep	...	5	...	10	...	20	...	40	...	80	...	160	...	224

則為相對應的量化值，可以看到每當 QP 增加 6 的時候 QStep 量化值呈 2 倍成長。在本篇論文中所使用的 QP 皆為 28，所以使用的 QStep 為 16，而這個量化值的變化，會直接影響在編碼器做了運動預測後的量化結果，和整體的編碼品質和效能。在編碼器所設定的 QP 種類包含了 I、P、B 畫面的個別 QP 值，可以根據需要的壓縮率和品質來調整。QP 的調整在整個 Codec 佔了相當大的一個控制因素，也是直接壓縮影像最直接的編碼工具。

4.4.2 是否使用 Hadamard

在 H.264 標準中，加入了一個 4x4 的轉換矩陣，叫做 Hadamard 矩陣，在本論文第二章有提到過。因為在以前的標準中，經過轉換後的資料就會有一部分的失真，為了更接近原始影像的結果，H.264 中使用的此可逆矩陣來轉換，經過轉換後在重建影像的過程中就不會有資料的損失，如此就可以保留更接近原始影像的正確資訊。

4.4.3 參考畫面的使用

在 H.264 由於有支援多張畫面的預測，對於前景和背景的應用，可以有更多的選擇，也就是在參考畫面選擇上，前景使用較多的參考畫面數，而背景因為變動少，可以直接以最接近的畫面當做參考畫面即可。

4.4.4 不同 Entropy Coding 的影響

H.264 標準中，提供了 2 種 Entropy Coding 的方法，一種叫做可變長度編碼 (Variable-Length Coding, VLC)，一般影像在經過預測、轉換、量化後，使得大部分的資料已經變成 0，所剩下的資料量只有一小部分的結果，再利用資料間的重複性 (Redundancy) 來減低資料量，而另一種 Content-Based Adaptive Arithmetic Coding (CABAC)，複雜度較高，其方法以二元方式編法為出發，也就是所有要編碼的資訊都先轉換成 0、1 來表示，在利用機率的方式動態的更新每次所要編碼的資料，此種方法的壓縮率較高，不過時間也花費較久。其中 H.264 中使用 CABAC 的編碼方式，壓縮效率普遍會比較好。

第五章 實驗結果與分析

這個章節主要是討論所第四章和第五章所使用的一些方法和參數設定，並且呈現影片壓縮後的結果。

5.1 實驗平台

實驗的環境是在 Windows XP Professional 作業系統，CPU 為 Pentum4 2.4GHz, 256MB RAM 的個人電腦。使用程式開發工具為 Visual C++ 6.0，所採用的 H.264 原始碼是 JVT 所發展的 Joint Model(JM 7.3 版本)[16]，所使用的影片皆為 MPEG 組織標準測試影片。



- A. Foreman, CIF, 300 Frames, YUV 4:2:0, Color Level
- B. Stefan, CIF, 300 Frames, YUV 4:2:0, Color Level
- C. Singer, CIF, 250 Frames, YUV 4:2:0, Color Level
- D. Dancer, CIF, 250 Frames, YUV 4:2:0, Color Level

5.2 評估標準

一般來說影片的畫面品質標準其實很難界定，通常最常用的衡量標準為 PSNR 值(式 5.2.1)，但是人的主觀意識有時候並非和 PSNR 呈現一致的結果，因此有時候直接用肉眼來評估也是一種方法。而評估資料量的大小，則是採用位元速率(Bitrate)的大小做比較，通常是以位元/秒(Bit/Se)表示，另外比較編碼的速度(Coding Speed)，這裡我們以毫秒(Millisecond)來計算編碼時

間。

$$PSNR = 10 \times \log \frac{255^2 \times W \times H}{(I - \hat{I})^2} \quad (5.2.1)$$

$W = width$, $H = height$, $I: origin$, $\hat{I}: restructed$ (use Chinese)

5.3 整張畫面使用相同大小區塊

此節主要實驗一整張的畫面使用相同區塊大小的結果。如表 5.1，我們設定的影片序列以 III... 的片段作編碼方式，其中 Foreman 和 Singer 的影像序列在使用 4x4 區塊的時候，PSNR 值會比較高，而 Stefan 和 Dancer 則使用 16x16 較高，但從表 5.1 可看出，其實差異值都不大，因此 I 畫面在區塊的選擇上似乎沒有太大的效果改善。使用固定區塊大小的結果在 4x4 和 16x16 差異並不大，導因於 I 畫面是以 Intra Prediction 為預測方法，所以在壓縮過後損失的影像將會是最小的。不過也因為是損失最少的影像資訊，所以其資料量也是佔最大的。

表 5.1 I 畫面的區塊編碼結果

	Block	PSNR				Bitrate			
		Foreman	Stefan	Singer	Dancer	Foreman	Stefan	Singer	Dancer
I	16x16	2976	4988	3076	1334	37.05	36.39	37.97	40.67
I	4x4	2483	4565	2687	1309	37.17	36.34	38.01	40.48

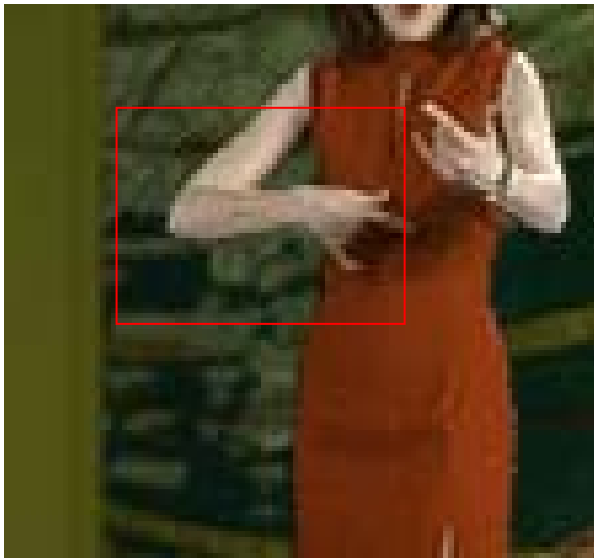
接著我們討論 P 畫面的區塊大小，在 P 畫面可以使用的區塊為 16x16、16x8、8x16、8x8、8x4、4x8、4x4、Skip 共 8 種區塊種類，在此我們不討論 Skip 種類的區塊，因為當前後畫面所產生的運動向量接近零的時候，使用此種的區塊(測試影片中只有 Singer 的影片序列適合使用在背景部分)才會較好。如表 5.2，

所使用的影片序列為 IPPP... 的片段編碼方式，可以看到 4 種使用的影片，當區塊使用的越小的時候，PSNR 大部分有增加的趨勢。

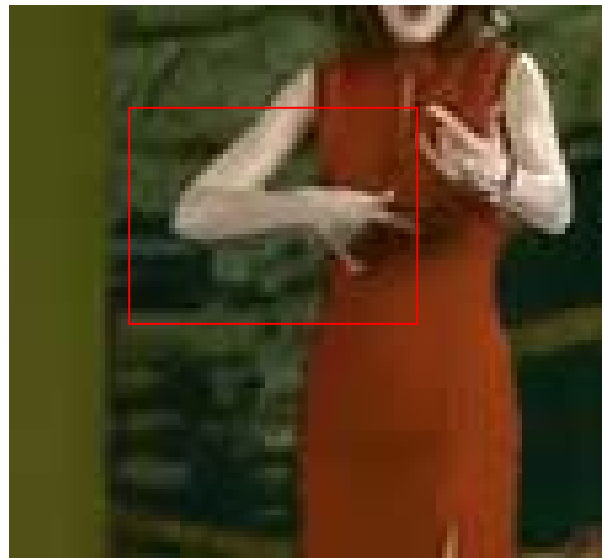
表 5.2 P 畫面的區塊編碼結果

IPPP	Block	PSNR				Bitrate			
		Foreman	Stefan	Singer	Dancer	Foreman	Stefan	Singer	Dancer
P	16x16	446	1409	283	627	36.25	35.14	37.53	39.33
P	16x8	505	1419	373	714	36.45	35.22	37.80	39.53
P	8x16	499	1405	376	692	36.47	35.22	37.81	39.56
P	8x8	594	1458	465	773	36.55	35.26	37.89	39.58
P	8x4	793	1659	644	981	36.56	35.27	37.95	39.50
P	4x8	798	1630	648	964	36.54	35.26	37.92	39.53
P	4x4	1108	1754	920	1276	37.47	35.36	37.94	39.37

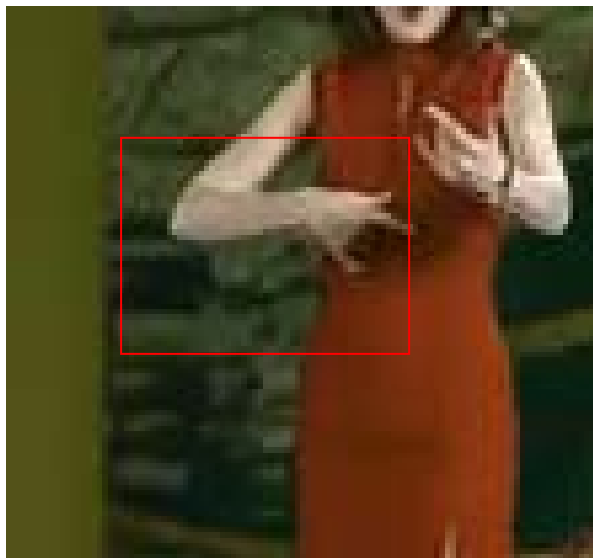
而對於 P 畫面的編碼結果，平均值差異不大，但是比較一張張實際的畫面，可以發現有一些比較明顯的差距。圖 5.1 為 Singer 影片序列使用表 5.2 中 P 為 16x16 和 8x4 區塊的影像編碼結果，可以看到在影像中右手肘的部分，使用 8x4 區塊的線條較為銳利，但是差異並沒有非常明顯，顯示在這影像內容上，以不同區塊大小作壓縮，並沒有太大的作用。接著比較 B 畫面的壓縮，在 B 畫面只列出 2 種，16x16 和 Direct 種類的比較結果，從表 5.3 其中可以看到 Singer 的影片在使用 Direct 種類後資料量有減少的趨勢，是因為影片中的背景區塊是完全靜止的畫面，而另外 3 種則有不斷變動的背景區塊。因為使用 Direct 種類後，都是利用時間的比例來預測，因此所編碼後的影像品質通常都會略差，不過可以看到背景區塊只要越接近靜止的話，品質也就不會下降太多了。雖然 Direct 的編碼效果不好，不過它的速度卻是最快的。



(a) 原始影像



(b) 16x16 區塊的 P 畫面



(c) 8x4 區塊的 P 畫面

圖 5.1 Singer 主觀畫面品質比較

表 5.3 B 畫面的區塊比較

IBPBP	Block	PSNR				Bitrate			
		Foreman	Stefan	Singer	Dancer	Foreman	Stefan	Singer	Dancer
B	16x16	355	1100	284	577	36.67	35.42	37.78	39.70
B	direct	990	2990	207	881	35.94	34.68	37.74	38.99

5.4 使用 Rate-Distortion 來控制編碼

在這個部分，使用第四章所提到的的速率-失真率公式來改善編碼效能

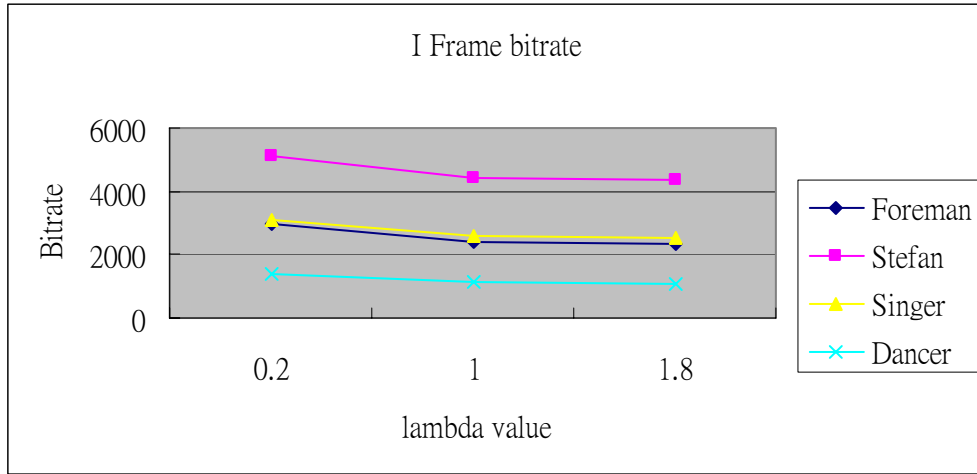
$$Cost = Distortion + \lambda * Rate$$

這節將比較不同的 λ 值，看看資料量和 PSNR 值的增減是多少。

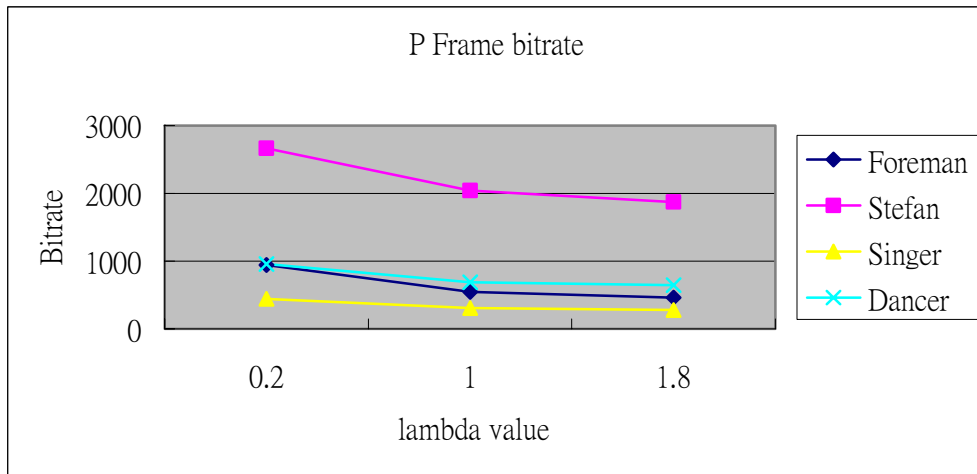
表 5.4 比較 λ 值不同而影響的位元速率和 PSNR 值

IBPBP	λ factor	PSNR				Bitrate			
		Foreman	Stefan	Singer	Dancer	Foreman	Stefan	Singer	Dancer
I	0.2	2985	5128	3120	1384	38.40	37.93	39.36	41.74
I	1	2382	4451	2596	1116	37.51	36.96	38.38	40.95
I	1.8	2315	4359	2540	1094	37.25	36.62	38.10	40.74
P	0.2	942	2666	444	959	37.41	36.18	38.78	40.88
P	1	546	2042	308	689	36.58	35.50	37.86	40.01
P	1.8	461	1872	277	646	36.13	35.02	37.51	39.63
B	0.2	404	1503	231	568	37.30	35.97	38.68	40.47
B	1	229	1075	152	449	36.26	34.80	37.58	39.53
B	1.8	208	1018	129	431	35.78	34.29	37.10	39.05

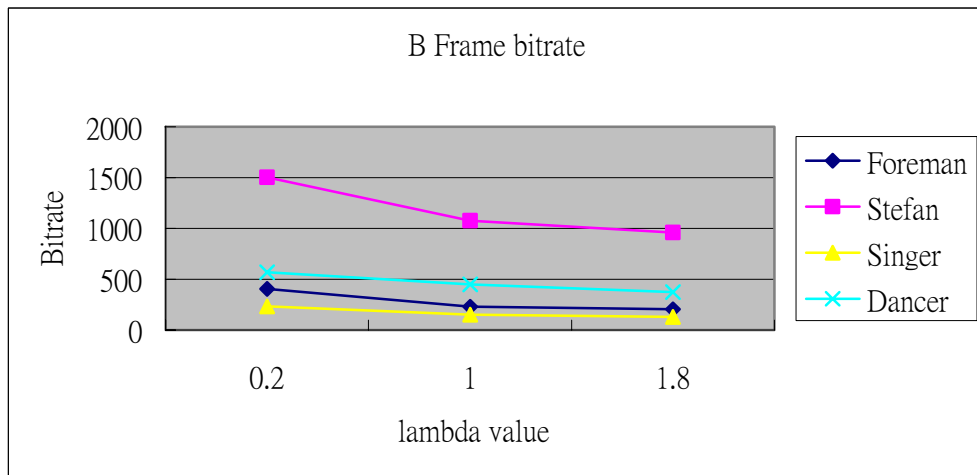
表 5.4 比較了 I、P、B 畫面以不同的 λ 值編碼的結果，表中主要比較的是 3 種畫面在不同 λ 下的差異，而圖 5.2 (a)(b)(c)，分別為 I、P、B 畫面的位元速率圖表，可以看出不同的影片在這三種畫面中，當 λ 變大的時候，其位元速率會變小，因此可以使用此方法來調整我們所需要的位元速率的大小。



(a)



(b)



(c)

圖 5.2 四段影片相對不同 λ 值的位元速率

在調整不同的 λ 值，也會影響到 PSNR 的結果，如圖 5.3 為表 5.4 中 Stefan 影片的畫面比較。可以看到在運動者和裁判之間的牆壁上的差異，並且在運動者的身體周圍，使用越小的 λ 值看起來畫面比較接近原始影像，因此這方法也可以使用在影像主觀品質的提升上。

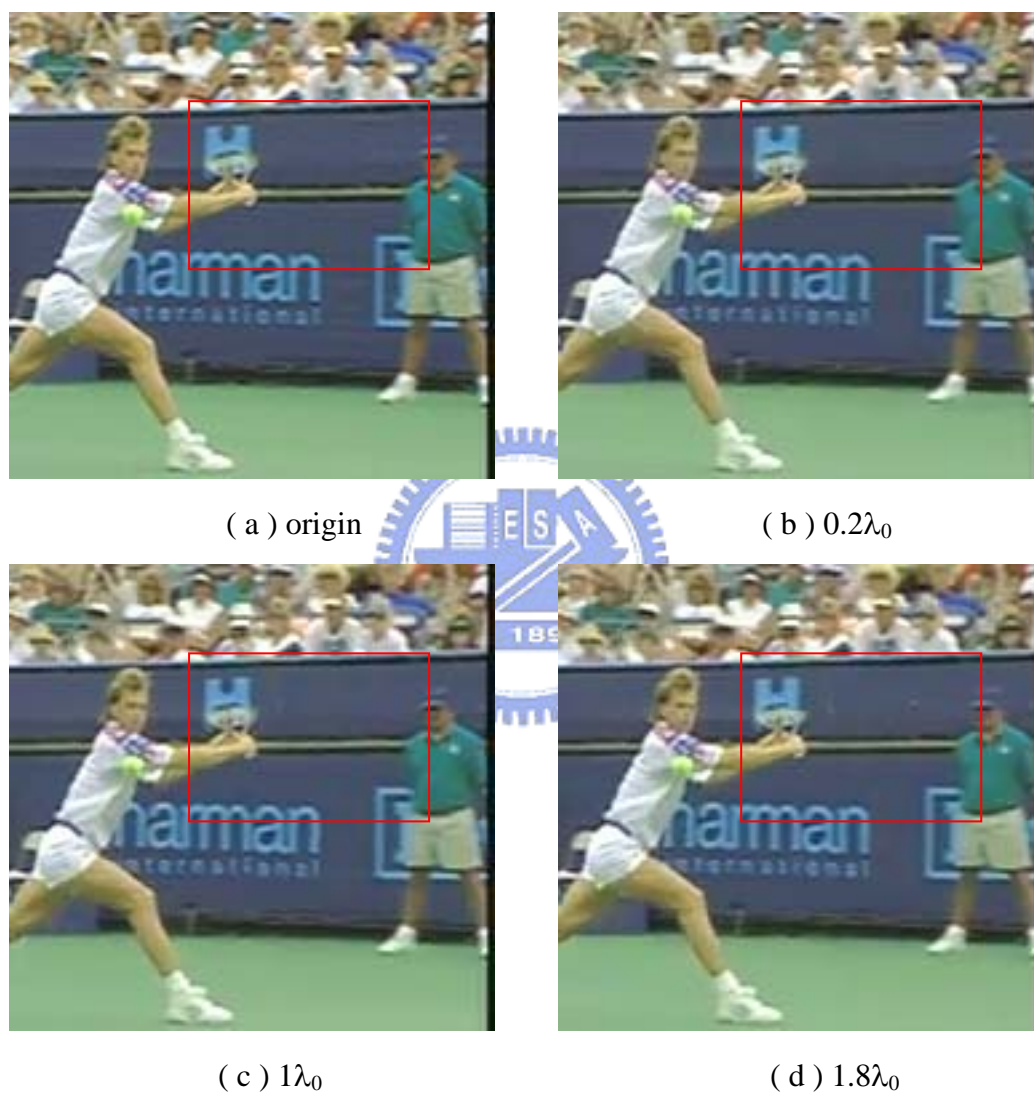
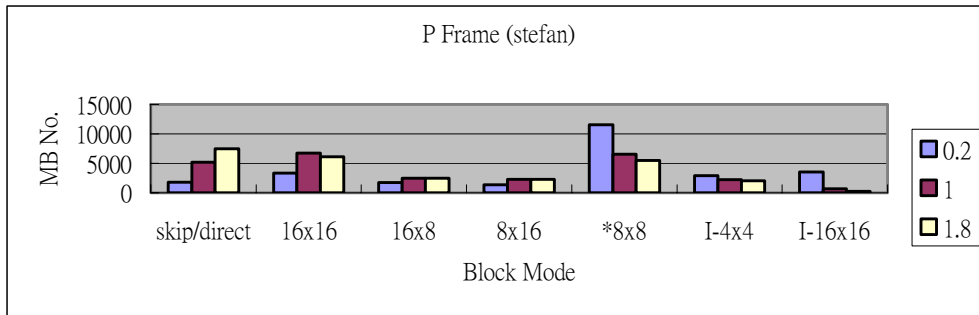
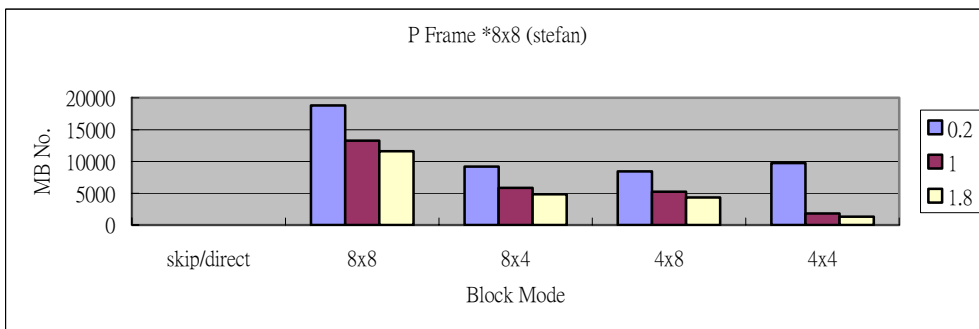


圖 5.3 不同 λ 對於影像品質的影響



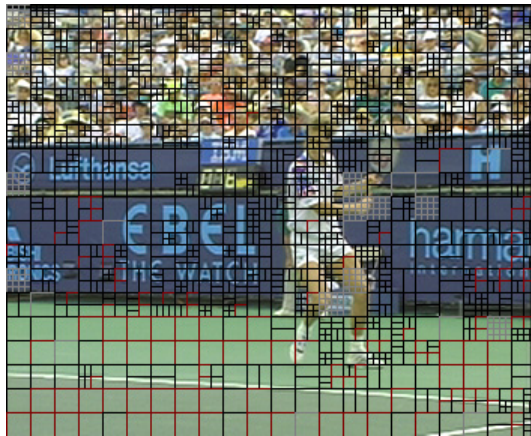
(a)



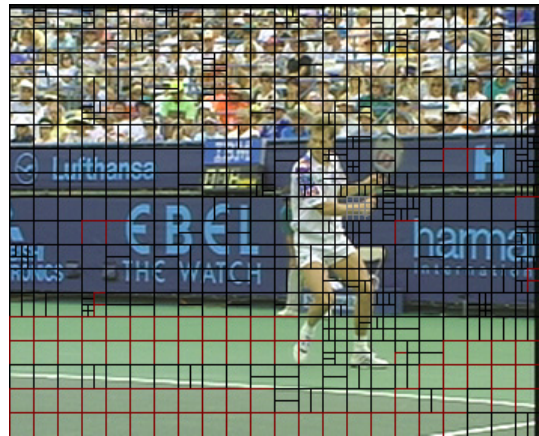
(b)

圖 5.4 不同 λ 的區塊種類分布

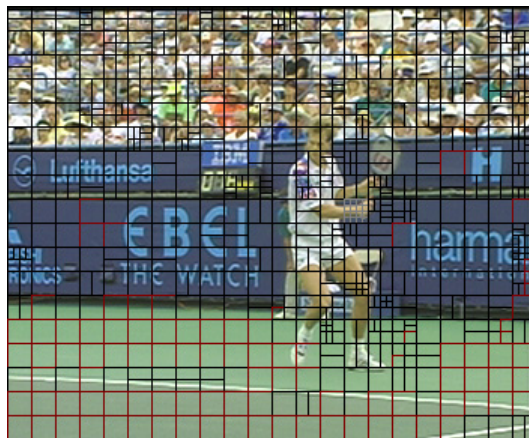
另外，這裡我們也探討區塊大小的相關問題，把這些不同 λ 值的分別以畫面來表示，畫出圖 5.4 的分佈圖，其中為對於整段影片中的 P 畫面統計的總合結果，並且可以由畫面中的內容來了解其中區塊大小分布的情況。如圖 5.5 所示，黑色的編為 16x16~4x4 區塊大小的種類，而灰色的為 Intra Prediction 的區塊，紅色則為 P、B 畫面的 Skip 或是 Direct 區塊種類，可以看到當 λ 提高的時候區塊也慢慢的變大了，這樣的結果可以當作區塊種類使用上的參考。



(a) $0.2\lambda_0$



(b) $1\lambda_0$



(c) $1.8\lambda_0$

圖 5.5 不同 λ 對於影像區塊大小的影響

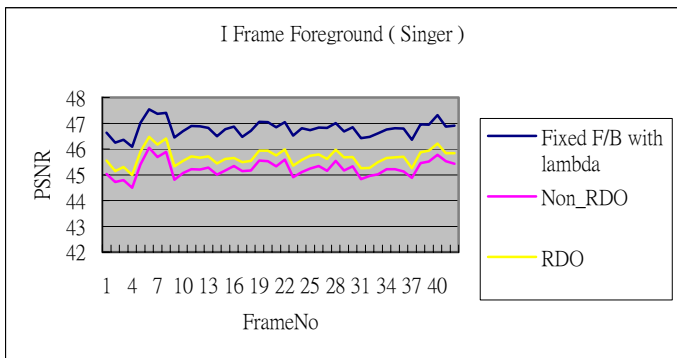
5.5 使用前景和背景分析編碼

在這個部分，主要是把影片利用前面章節所提到，加入已經切割過的前景和背景影像來和原始的編碼流程結合，再加上 5.3 節和 5.4 節所使用固定的區塊大小和 Rate-Distortion 來進行混合式編碼。

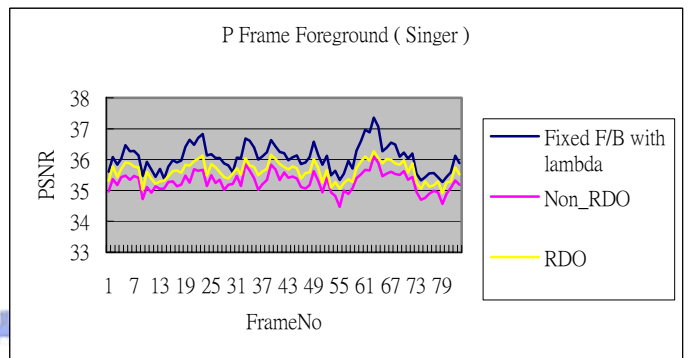
這裡我們以 5.3、5.4、5.5 各節的結果(找出前景和背景區塊較適合的編碼方式)來進行混合式編碼，使用的編碼方式如表 5.5，對於 I、P、B 前景部分我們都使用 4×4 區塊編碼，並且使用 λ 為 $0.1\lambda_0$ ，而背景皆使用 16×16 的區塊並且 λ 為

表 5.5 Singer 影片編碼方式

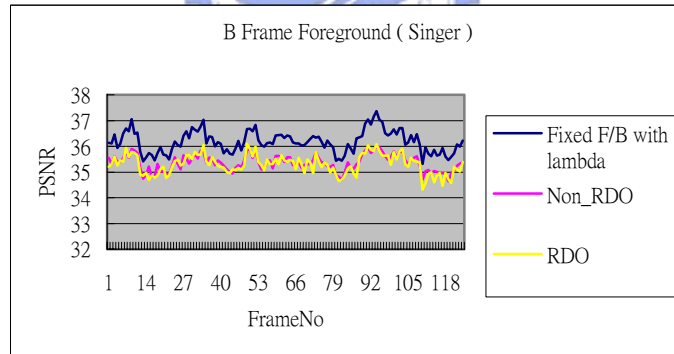
Singer	Foreground		Background	
	block	lambda	block	lambda
I	4x4	$0.1\lambda_0$	16x16	$3\lambda_0$
P	4x4	$0.1\lambda_0$	16x16	$3\lambda_0$
B	4x4	$0.1\lambda_0$	16x16	$3\lambda_0$



(a)



(b)



(C)

圖 5.6 Singer 編碼後的前景 PSNR 值比較圖

$3\lambda_0$ ，結果可以看到圖 5.6 分別畫出 I、P、B 畫面的 PSNR 比較結果，如圖 5.7 可以看到前景部分(字體)在 I 部分大約提升 1.2db 值，而在 P、B 畫面則提升大約 0.7db，我們的方法前景部分周圍就無雜訊。



(a)原始影像

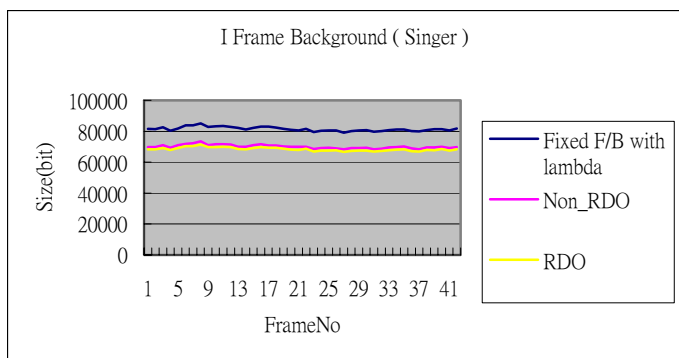


(b)我們的方法

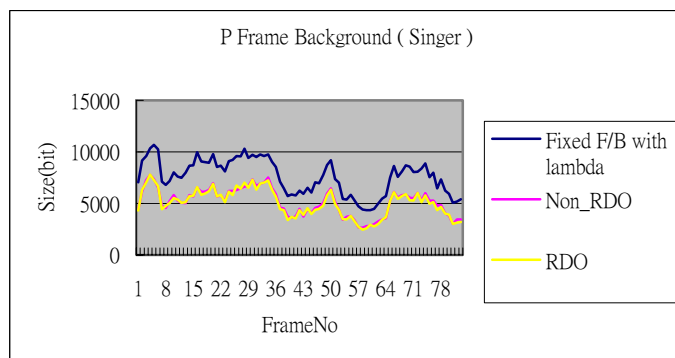


(c) Non_RDO 方法

圖 5.7 Singer 第 246 張影像



(a)



(b)

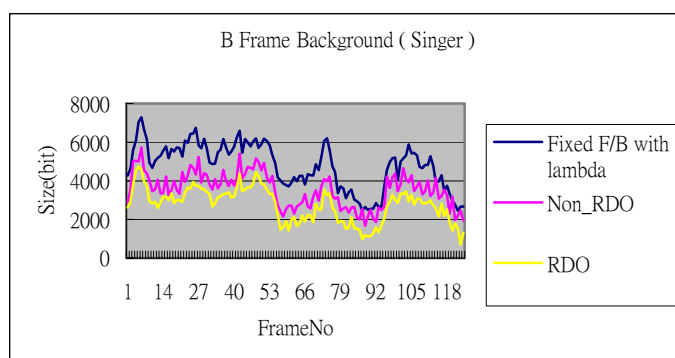


圖 5.8 Singer 使用表 5.5 編碼方式的背景資料量比較

由圖 5.6 可以觀察到，在混合式編碼中，調整的區塊大小和 λ 的參數值，使得前景的部分，PSNR 的值比 H.264 預設的非 RDO 和 RDO 編碼結果都要好，並且當 λ 增加 PSNR 的值也隨之增加(如圖 5.2 的結果)；而在 Bitrate 的部分，卻無法達到預期的效果(如圖 5.8)，最主要的原因就是因為當背景使用了固定的區塊大小時，此時編碼過程中如果使用較大的 λ 值，即使我們使用 Rate-Distortion 調整 λ 值去找到較佳的位元速率壓縮方式，但是卻也間接的造成了更多的預測誤差。而因為在編碼過程中，經過預測過的區塊後，所產生的預測誤差也需要經過編碼，當預測誤差越大時候，則需要更多的空間來儲存這些誤差，也因此造成了資料量的提升，因此使用的區塊大小如何取捨以及如何選擇較佳的 λ 參數對於前景和背景分析的應用將非常重要。

因此本節在最後的實驗中，前景將使用固定區塊大小，加上設定 λ 來微調

區塊種類決定，使得畫面品質的 PSNR 提升，而對於背景，只對 λ 來調整，由 RDO 自動決定資料量小的區塊大小，達到資料減量的目的。如圖 5.9 在背景的部分，改成只使用不同的 λ 加以調整，而不再指定背景使用的固定區塊大小，因此可以看出資料量的減少就非常明顯。

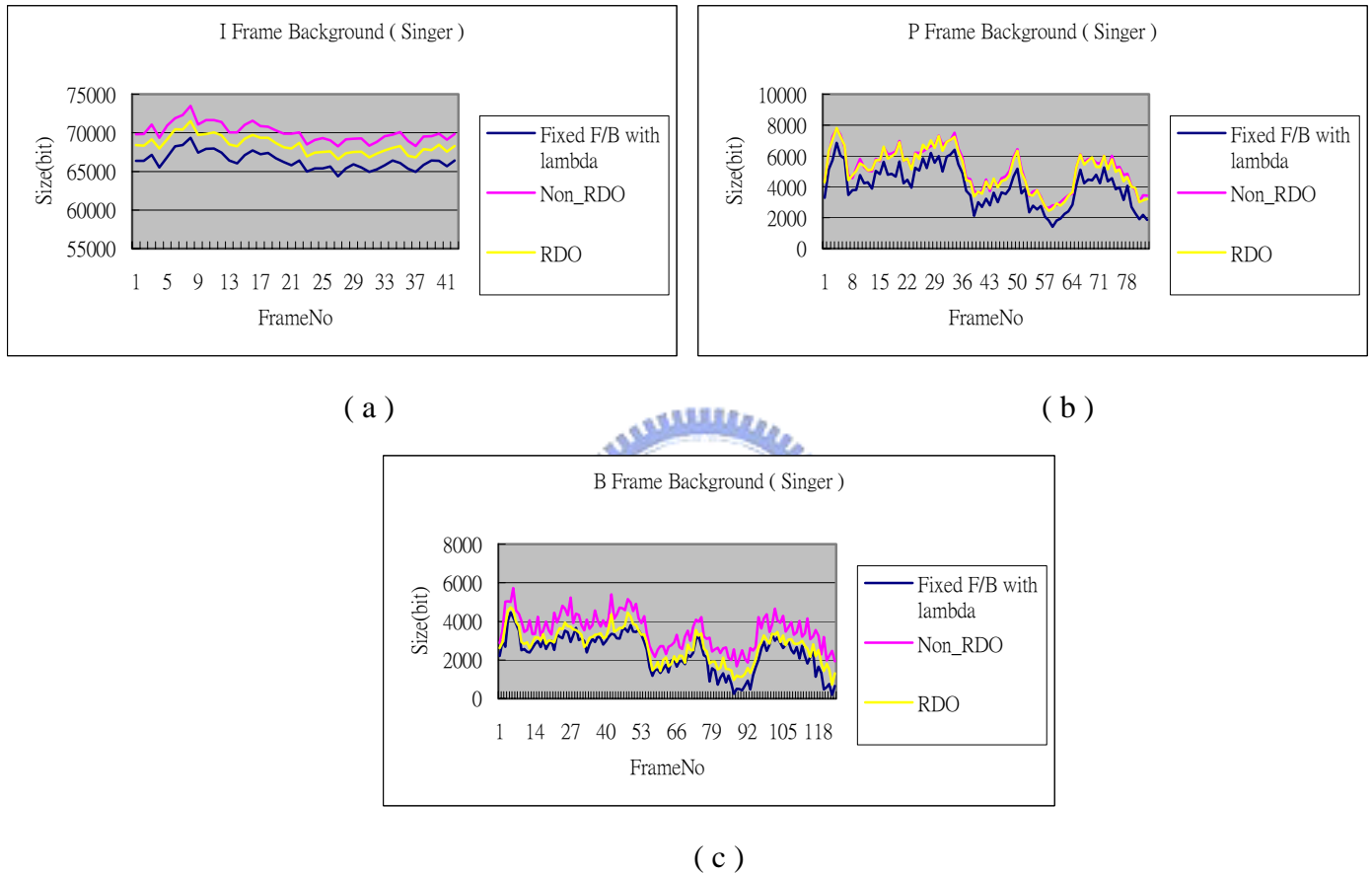


圖 5.9 Singer 不使用固定區塊來編碼背景

由以上的實驗可以看出，目前我們所使用的方法可以在前景和背景的應用上，達到前景 PSNR 值的提升，並且減少背景的資料量。

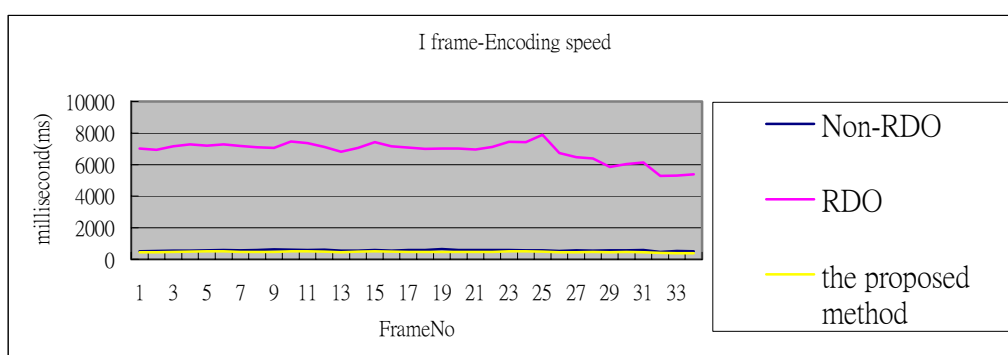
5.6 比較編碼速度

表 5.6 為使用 Foreman, CIF, 30fps, 10 秒的影片所做的測試，使用的 QP、Entropy Coding 等的參數設定都相同下所測試，其中數據為每種畫面的平均編碼時間，圖 5.10 表示對於此段影片的每一個 I、P、B 畫面的比較，可以看到其中

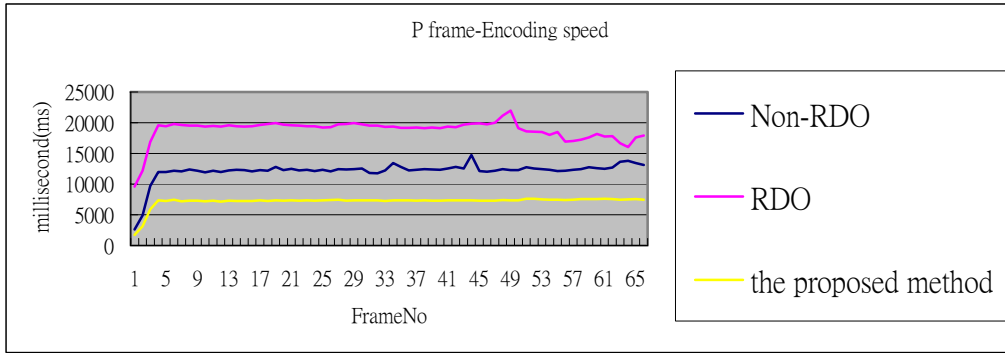
表 5.6 編碼速度的比較

單位：毫秒 (millisecond)	Average I	Average P	Average B
non-RDO	566	12148	15868
RDO	6864	18826	23260
our methods	466	7188	3969

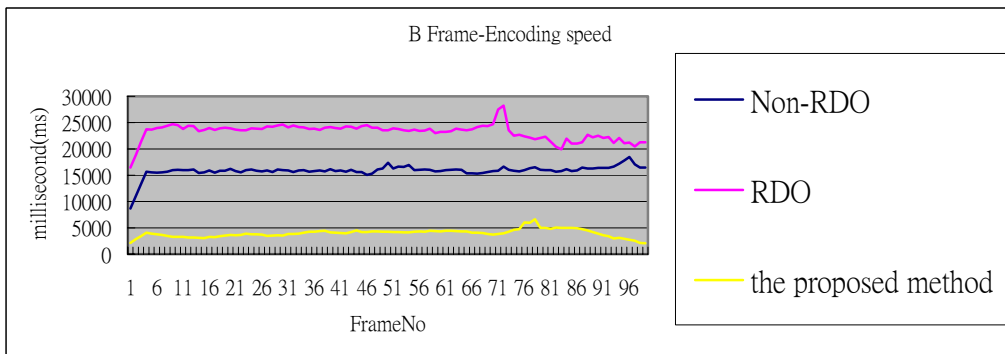
編碼時間都很平均，除了一些誤差上可能由電腦本身造成，可以看到 P、B 畫面中一開始的時間都比較少，因為在標準中採用了 5 張的參考畫面，因此再剛開始的編碼參考數量較少所以時間也較少。(表 5.6)比較所使用的 3 種方法，可以看到使用前景和背景分析的結果所編碼出來的速度在 P、B 畫面比 RDO 和 Non_RDO 都快了不少，尤其 B 畫面所增加的編碼速度更明顯，主要的原因是因為這裡 B 畫面需要做雙向的預測，需要較多的計算預測時間，所以在比較之下，本論文的方法則節省更多的預測時間。



(a)



(b)



(a)

圖 5.10 I、P、B 的不同編碼方法在時間上比較



5.7 問題與討論

在本論文中使用的一些編碼工具，主要都是以前景和背景區塊來做分析使用。因為既定的語意和語法，因此對於整個標準上的編碼工具，對於不同品質和速率的編碼，大部分都只能在標準內取捨。在 H.264 標準中，因為編碼的複雜度非常的高，所以使用的編碼工具時，必須依照原來的語意和語法，因此彈性也就下降了，在本論文提到最主要的 2 個方法為區塊大小的選擇和 λ 值的設定，以結果來看，可以達到前景品質的提升並且背景可以達到資料的減量；以區塊的選擇來說，在編碼過程中並沒有去更動其中的預測方法，如果也更改了預測方式，或許會有更好的效果。而在 λ 值的選擇也是一個問題，實驗中的結果為測試出

較為可接受的值來使用，似乎沒有什麼既定的方法可以來決定理想的 λ 值，因此目前都是以人工的方式來測試到底要怎麼選擇較好。



第六章 結論與未來展望

本篇論文提出一個新的編碼方式，基本概念是把影像分成前景和背景來討論，希望以不同的編碼方式來達到編碼效率的改善。在此所認定的編碼效率為在前景的部分希望提升其主觀和 PSNR 的品質，而背景的部分希望有更大的資料壓縮量，主要是利用在整個 H.264 標準中的編碼工具來改善，針對工具的細節來分析是否可以有不同的方式編碼。而我們的方法所提到的區塊大小的使用和 Rate-Distortion 的 λ 值設定，的確可以有效改善前景的畫面品質和背景的資料量，但由於視訊影片，主觀意識的品質認定很難用數據來評估，因此很多時候還必須要用人眼觀察來比較。



由於整個 H.264 標準的架構非常龐大，所以可以再分析的部分也應該很多，包含第五章所提到的量化參數、參考畫面數、Entropy 編碼等，都可能繼續應用在前景和背景的壓縮。在本篇論文，我們都以符合標準的解碼程序來做研究，當然，如果能在標準規定的語法和語意外來進行編碼，前景和背景分析的結果，應該會有更大的空間發展。

參考文獻

- [1] ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVT- E022, September 2002.
- [2] ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVT-F100, December 2002.
- [3] Iain E G Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons, 2003.
- [4] ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVT-G050, March 2003
- [5] Li, Z.G.; Pan, F.; Lim, K.P.; Feng, G.N.; Lin, X.; Rahardia, S.; Wu, D.J.;
” Adaptive frame layer rate control for H.264” , *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on* , Volume: 1 ,
Pages:581 - 584,6-9 July 2003
- [6] ISO/IEC IS 11172-2, MPEG-1 Video.
- [7] ISO/IEC IS 13818-2, MPEG-2 Video.
- [8] ISO/IEC IS 14996-2, MPEG-4 Video.
- [9] C. Stauffer and W. Grimson , “Adaptive Background Mixture Models for Real-time Tracking,” *Proc. Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, Fort Collins, Colorado, 1999.
- [10] A. Elgammal, D. Harwood, and L.S. Davis, “Nonparametric Background Model for Background Subtraction,” *Proc. Sixth European Conf. Computer Vision*, vol. 2, pp. 751–767, Trinity College Dublin, Ireland, 2000.
- [11] S. Khan and M. Shah, “Object Based Segmentation of Video Using Color, Motion and Spatial information,” *Proc. Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 746–751, Kauai, Hawaii, 2001.

- [12] Y. Lu, W. Gao, and F. Wu, "Efficient Background Video Coding With Static Sprite Generation and Arbitrary-Shape Spatial Prediction Techniques", *IEEE Transactions on Circuits and Systems for Video Technology*, vol 13, no 5, 394-405, 2003.
- [13] Wien, M., "Variable block-size transforms for H.264/AVC", *Circuits and Systems for Video Technology, IEEE Transactions on*, Page(s): 604- 613, July 2003.
- [14] M. Flierl and B. Girod, , "Generalized B pictures and the draft H.264/AVC video-compression standard", *Circuits and Systems for Video Technology, IEEE Transactions on* , Volume: 13 , Issue: 7 , Pages:587 – 597, 2003.
- [15] JVT Reference Software version 7.3,
ftp://ftp.imtc-files.org/jvt-experts/reference_software/ .

